What Makes A Song Popular?

Deeksha Banala, Liam Hooks, Jai Singh

2023-12-10

Exploring Sonic Success: A Deep Dive into Spotify Songs and Their Popularity

What is Spotify?

In the dynamic landscape of digital music, Spotify emerges as a prominent force, reshaping the paradigms of music discovery, sharing, and consumption. Launched in 2008, Spotify has evolved into an integral component of the contemporary music ecosystem, boasting an extensive catalog comprising over 70 million tracks and serving the varied preferences of its 345 million active users globally.

Diverging from conventional music platforms, Spotify employs a distinctive streaming model that grants users access to an expansive repertoire of songs without the encumbrance of traditional downloads. Its algorithmic sophistication is instrumental in curating playlists and recommendations tailored to individual user preferences, thereby fashioning a bespoke auditory experience. This democratization of music has fundamentally altered the dynamics of how we engage with diverse songs and genres, fostering a global community of impassioned music enthusiasts.

Within this rich musical tapestry, our investigation centers on elucidating the intricacies of Spotify's songs and discerning the underlying factors contributing to their popularity. As we embark upon this exploration of the sonic domain, our approach eschews comparative analyses and contested claims, concentrating exclusively on the digital subtleties that delineate the success of songs and genres.

Focus of this Study

The overarching research inquiry of this study engenders a comprehensive analysis of Spotify's expansive dataset. The objective is to decipher discernible patterns and constituent elements that propel specific musical compositions into the spotlight. In navigating this sonic realm, our investigative endeavor adheres rigorously to academic standards, driven by an earnest quest for insights into the transformative influence of the contemporary digital age on the dynamics of music popularity.

The central theme guiding our inquiry revolves around the exploration of "What attributes contribute to the popularity of a song or genre?" This investigation is underpinned by a meticulously curated dataset obtained from Kaggle. Leveraging this dataset, our aim is to delve into specific subquestions that will collectively culminate in a comprehensive answer to our primary inquiry.

Through a systematic analysis of the provided data, we intend to address key aspects that influence the popularity of songs and genres. The examination of these subquestions will afford us a nuanced understanding, allowing for a well-informed and substantiated conclusion regarding the overarching theme of "What makes a song or genre popular?" The utilization of the Kaggle dataset serves as the bedrock for our empirical exploration, providing valuable insights into the multifaceted dynamics that contribute to the musical landscape's popularity metrics.

Exploring the Dataset

Here we dive into what the dataset encompasses and how the valuable information it contains was utilized in our study.

What Does it Contain?

The dataset that has been selected from Kaggle is a rich collection of information about 30,000 songs available on Spotify. Each song serves as an individual case within the dataset, contributing to a comprehensive overview of musical characteristics. The dataset encompasses a total of 24 variables, providing a multifaceted description of each song. Notably, 10 of these variables are characterized by characters, 5 are represented as integers, and the remaining 9 are of the number type.

Among the 24 variables, a careful selection process has been undertaken to focus on the most relevant and useful features. The chosen variables include essential aspects such as popularity, genre, sub-genre, and release dates. These variables are integral for gaining insights into the songs' overall appeal, categorization, and temporal patterns. By narrowing down the dataset to these key variables, you can streamline your analysis and concentrate on the most meaningful aspects of the music data.

To better understand the nature of the selected variables, it's crucial to delve into their measurement scales. For instance, popularity is gauged on a scale of 0 to 100, reflecting the enduring appeal of each song over time. Meanwhile, other features like danceability, energy, speechiness, liveness, and acousticness are measured on a scale from 0 to 1, providing nuanced insights into the musical composition. The loudness variable, measured in decibels, spans from -60 to 0, offering a quantitative assessment of the songs' audio intensity. Additionally, the binary variable, mode, serves as a categorical indicator, distinguishing between major keys (represented as 1) and minor keys (represented as 0).

In summary, your dataset holds a diverse array of information about songs on Spotify, with a thoughtful emphasis on key variables that offer a comprehensive understanding of each musical piece. The varied measurement scales of these variables add depth to your analysis, enabling you to explore the intricacies of musical characteristics and trends within the dataset.

Data Pre-processing

Managing a dataset of 30,000 Spotify songs came with its challenges. Firstly, some songs appeared multiple times due to being in different playlists and genres. To be fair, we kept all instances of repeated songs, ensuring they were considered separately for each genre they associated with.

Another challenge was the inconsistent date formats for song releases. To make things clearer, we created a new column called "Cleaned Up Release Date," focusing only on the year of release. This helped maintain consistency and simplicity in analyzing when the songs were released.

When it came to visualizing the data, dealing with individual release years proved messy. So, we decided to group the years into decades, making it easier to see trends. To keep things balanced, we excluded songs from 1957 to 1959 and those from 2020, as the data for these years was comparatively limited. This approach allowed us to present a more straightforward and meaningful picture of how music has evolved over broader timeframes.

This made it much easier for us when it came to creating a summary table to understand the trends of each decade when it came to popularity of songs released in that decade.

The Decade Frequency table shown here shows the statistics of the popularity of songs in each decade. This information is important for tracking shifts in musical preferences over the decades and understanding what appealed to audiences in different eras. This can show evolving genres which could help to understand how a song could be made more popular. Another useful takeaway from this table is that comparing the statistics across decades allows for a comparative analysis. It can help pinpoint changes in the distribution of popular

Table 1: Decades Frequency Table

Decade	n	min	Q1	Median	Q3	Max
1960s	225	0	22	59	68	80
1970s	964	0	32	55	68	84
1980s	1377	0	25	49	63	83
1990s	2239	0	20	44	59	90
2000s	3932	0	4	37	57	83
2010s	22413	0	27	46	63	100

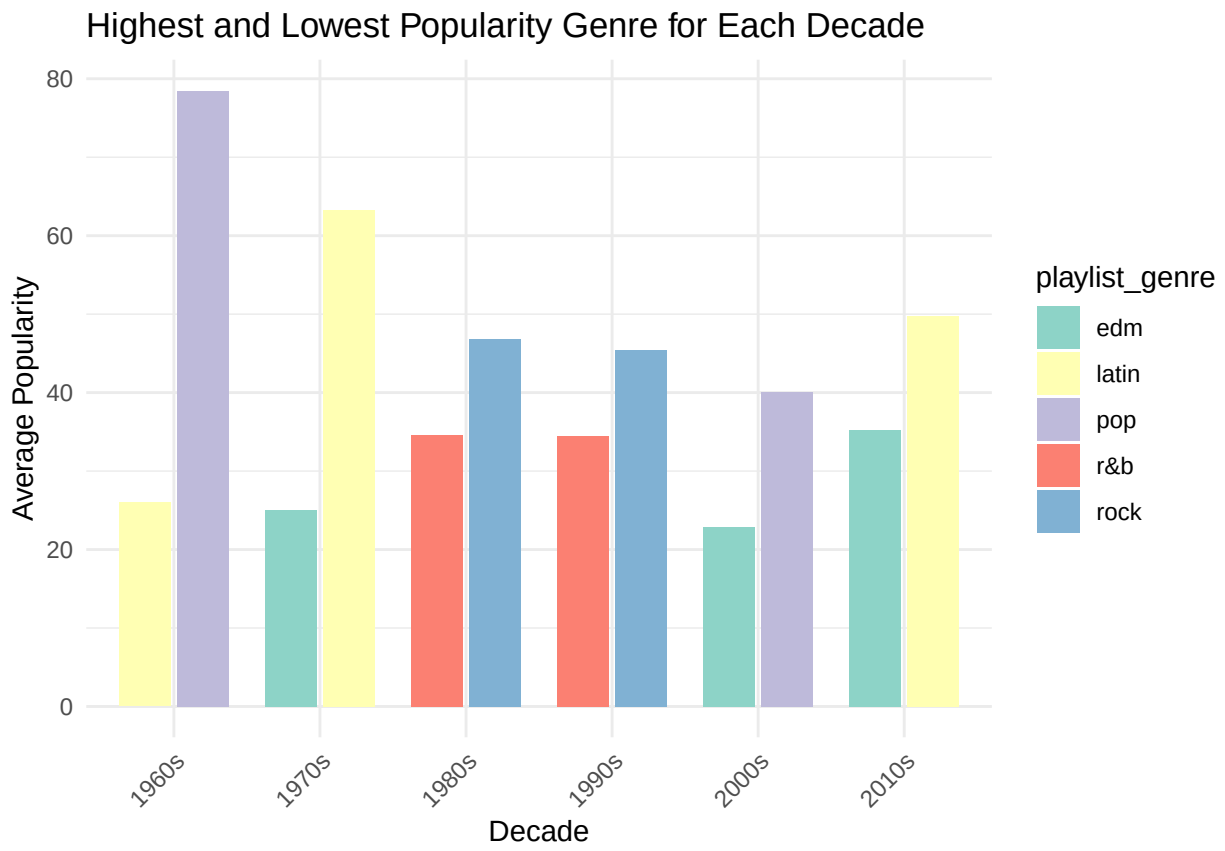
songs-whether there's a widening or narrowing gap in popularity among songs within each decade. One thing we noticed was that there was a noticeable increase in the number of songs per decade indicating a growth in music production.

Research Questions

This leads us to our main discussion of the research questions we have answered in this analysis. Our overarching theme is to find out what makes a song on Spotify popular. To analyse this, we formed a few subquestions to get down to the root cause.

What are the most and least popular genres in each decade?

With this question, we created a graph that easily shows the most and least popular genres in each decade. One major observation we noticed was that the 1960s had the most and one of the least average popularity of songs overall. The most popular genre was Pop with Latin second. Latin and Pop have similar audiences so it is safe to say that the most popular songs are going to be in one of these two categories. If a song wanted to be a popular song, it would be a good idea to make it for these type of audience. Even with Latin and Pop being extremely popular the most popular genres can switch like in the 1980s and 1990s. Their most popular genres was rock. This means that even though Latin and Pop are very popular, the popularity can change with the decades. That would mean that if a song wanted to be popular a good idea would be to follow the trend of the decade.



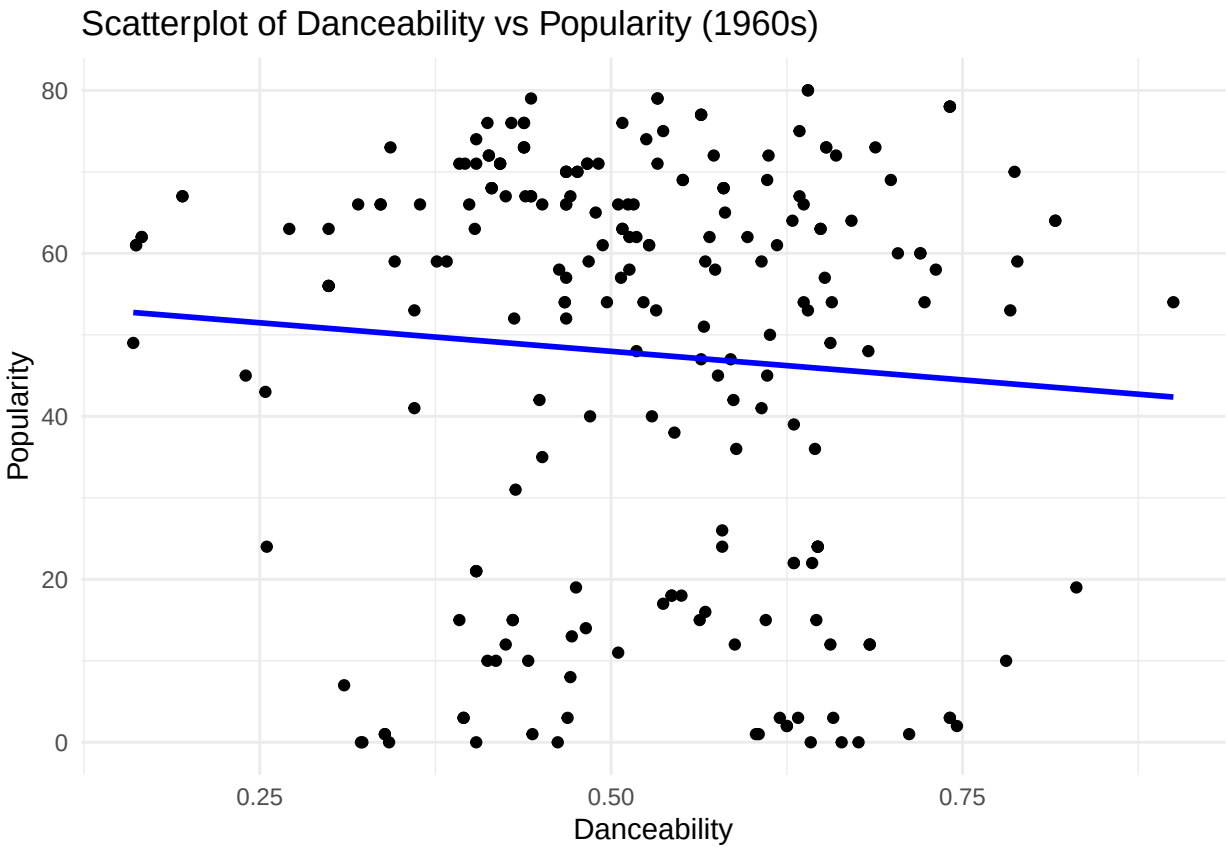
How have factors like danceability, loudness, duration, and tempo affected popularity over the years?

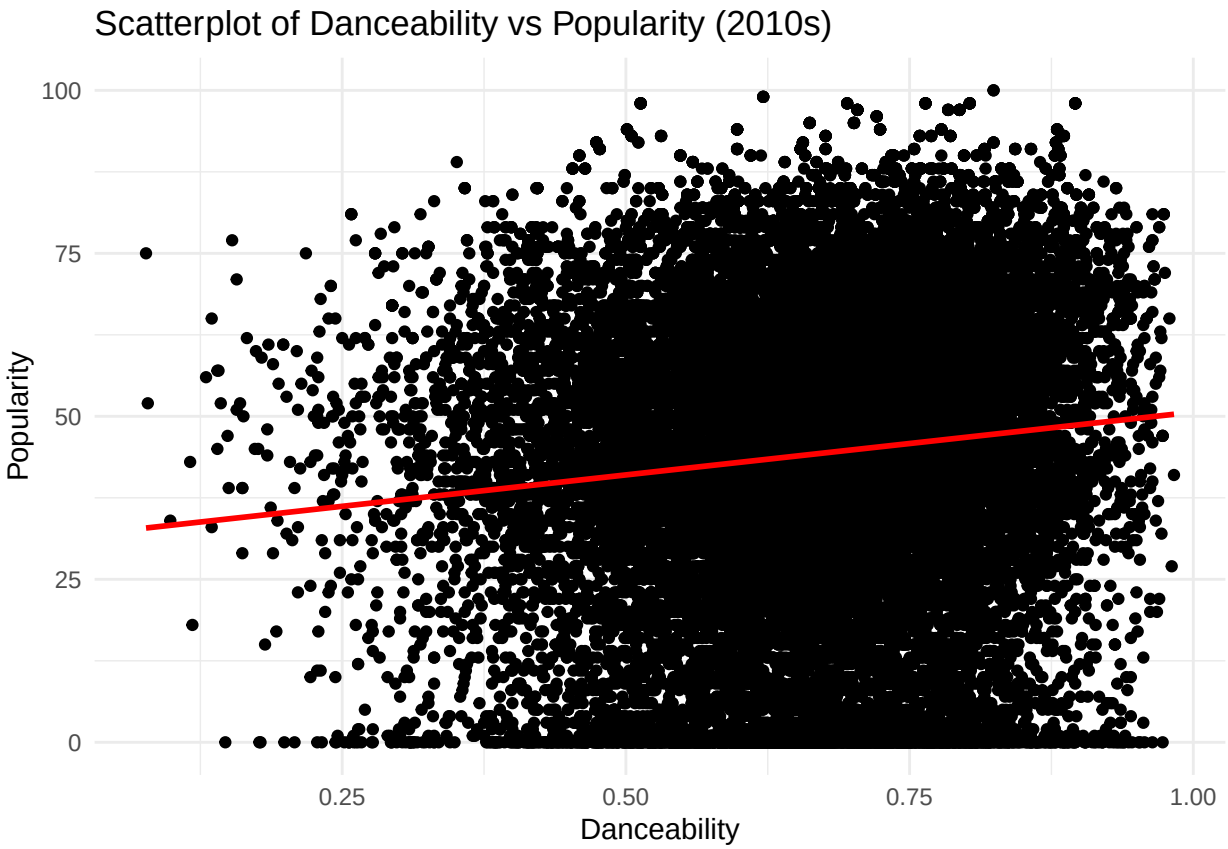
To take a look at these factors we have made eight graphs that show the 1960s vs the 2010s with danceability, loudness, duration, and tempo vs popularity. With these graphs we can see the trend of the decade and compare it with the other decade's trend. This will provide us with the knowledge to make assumptions with what factors in a song will increase it's popularity.

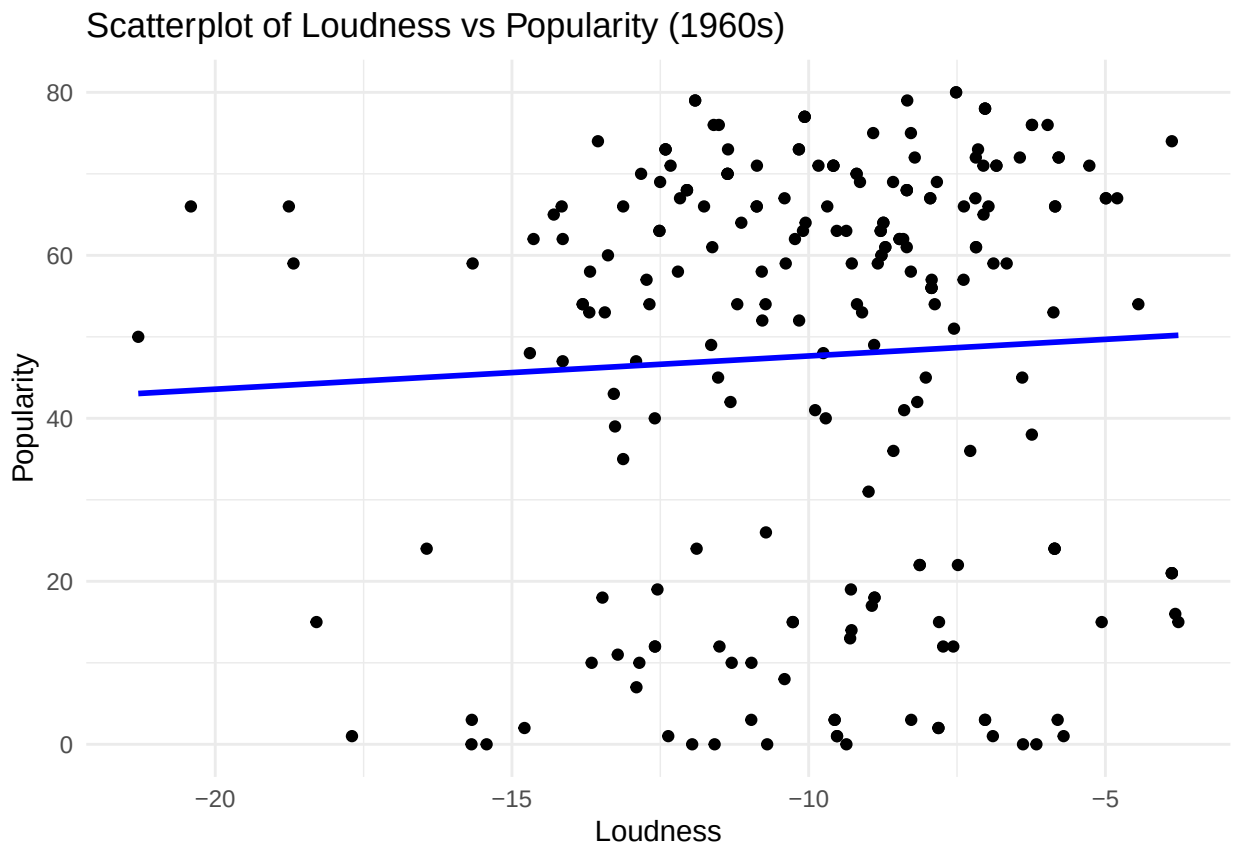
Little to no Change The graphs of loudness and tempo had similar trend lines. One thing to note was that in the loudness graphs, the 2010s graph had most of the dots skewed to the right compared to the 1960s graph. This means that the average song in the 2010 was more loud but the trend line was still the same as the 1960s one. This largely means that these factors had little to no affect on a songs popularity.

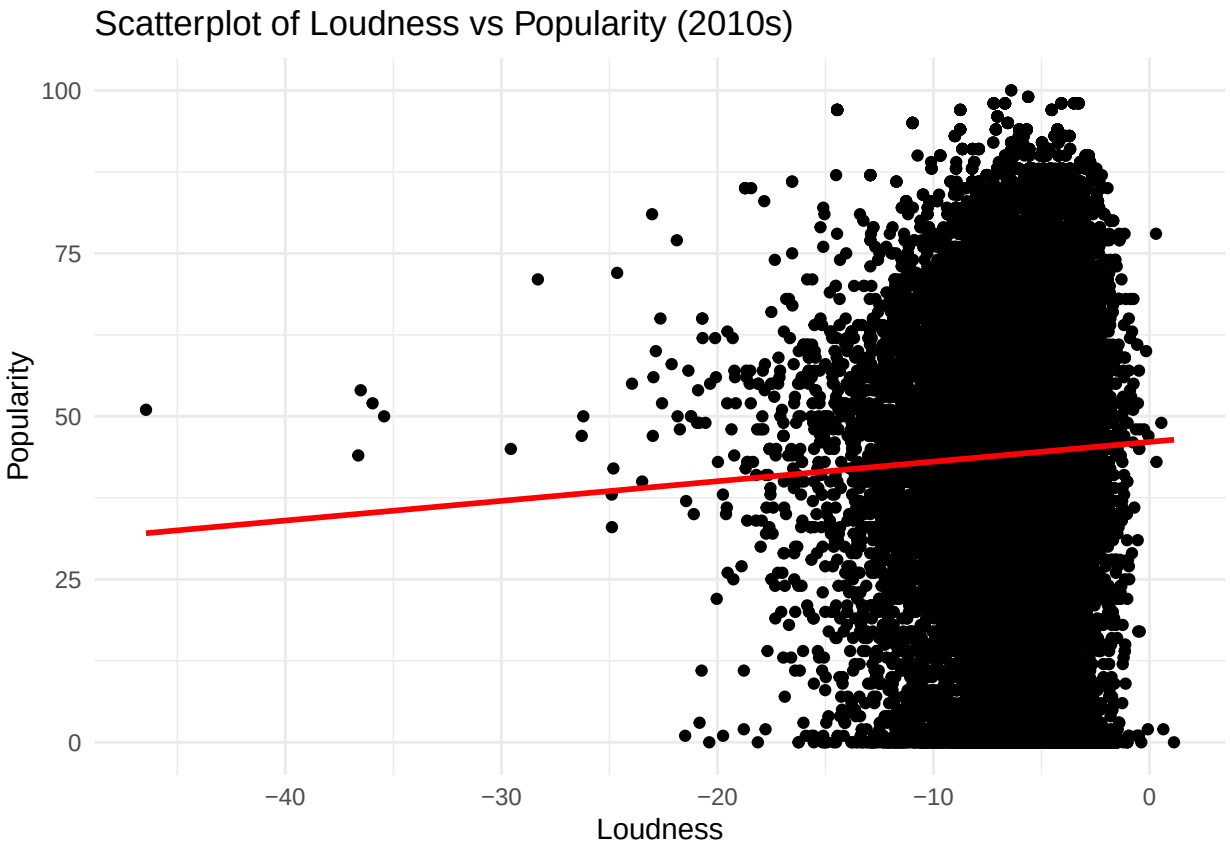
Opposite Change The only factor that had a opposite change was the danceability vs popularity graph. These graphs show that danceability was a bad thing in the 1960s where it would decrease the popularity but in 2010 it had a positive impact where songs with little danceability would be worse in popularity. These graphs show that song popularity factors can change over decades much like the decades more popular genres.

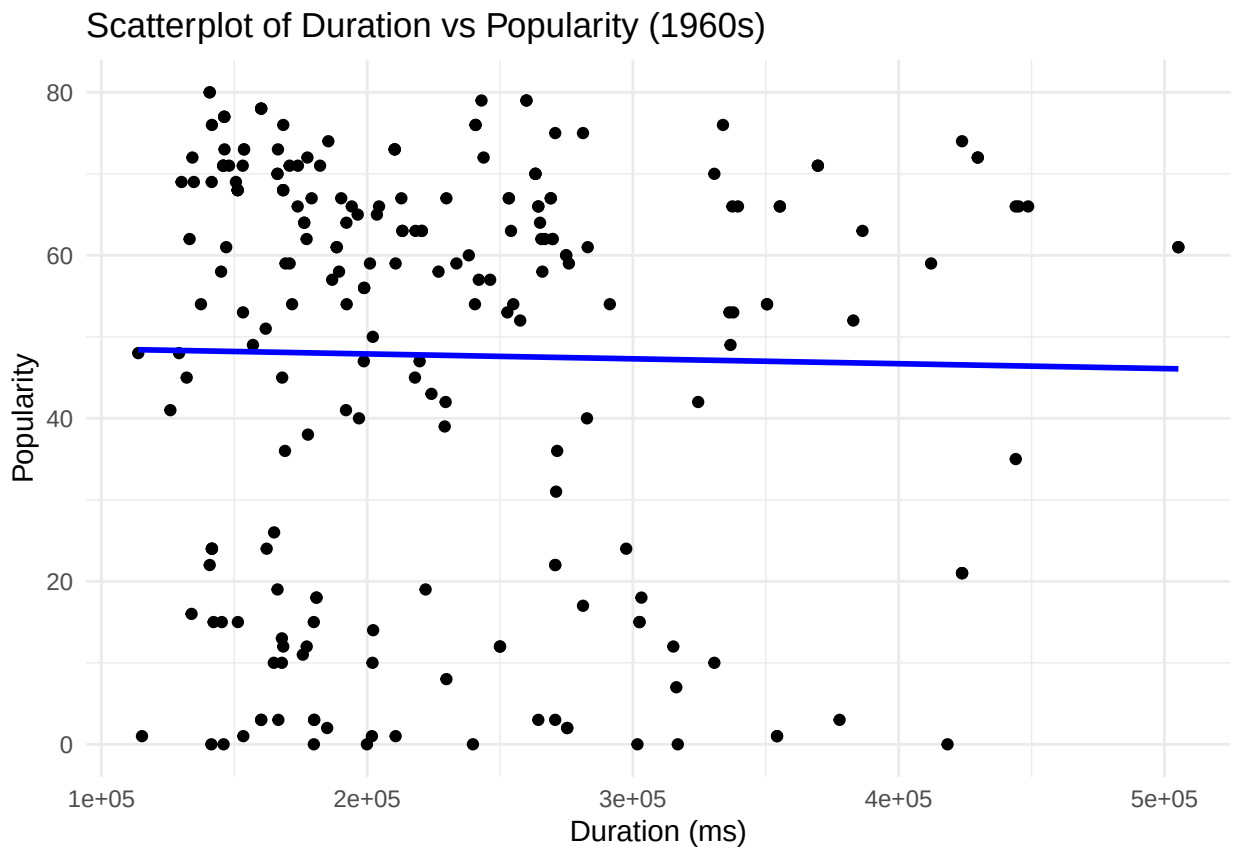
Same Change The graphs that had a big change but not in direction was the duration vs popularity. In the 1960s graph, the trend line showed that duration had little to no impact on popularity but int he 2010 graph the trend line is now negative where the longer a song goes on for the worse it is for the popularity. This also shows that as decades past the audience will change on what they have more of a preference to.

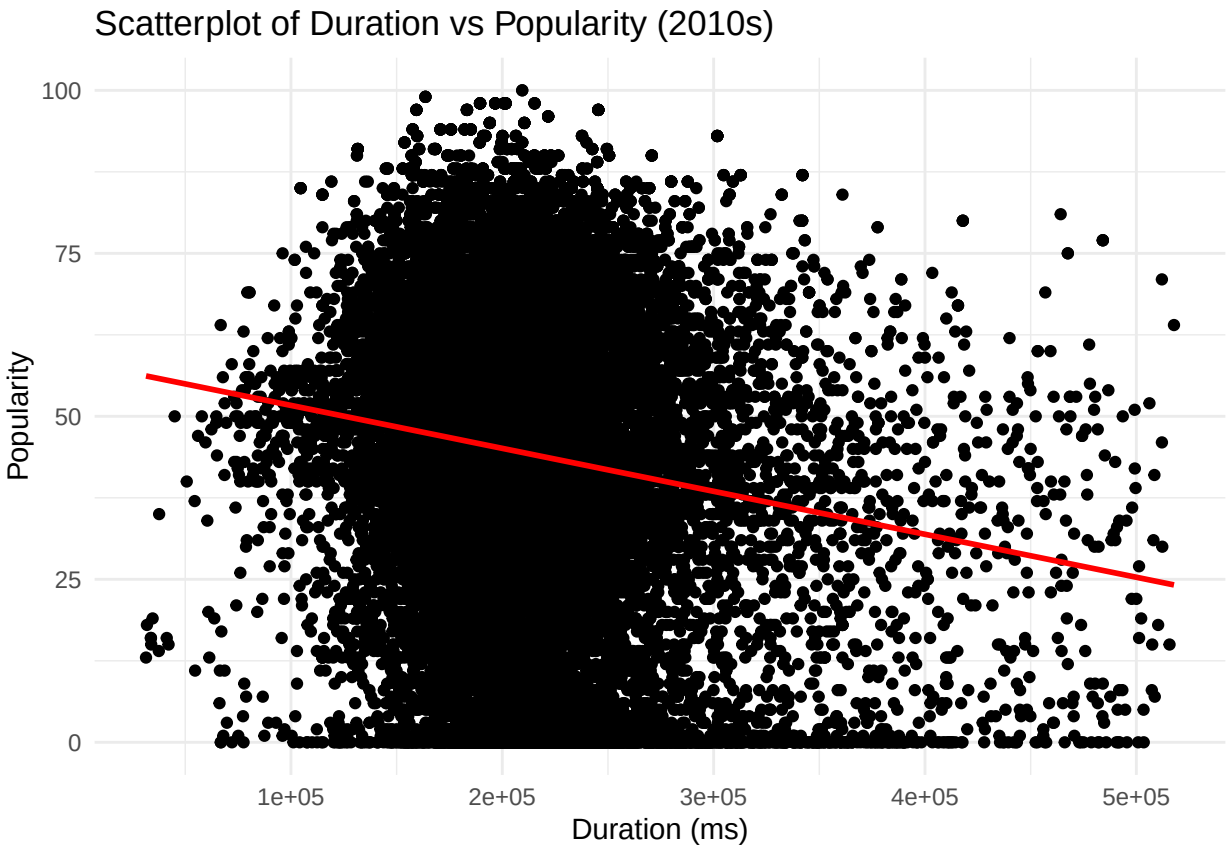

Scatterplot of Danceability vs Popularity (1960s)
Popularity
80
60
40
20
0
0.25
0.50
0.75
Danceability

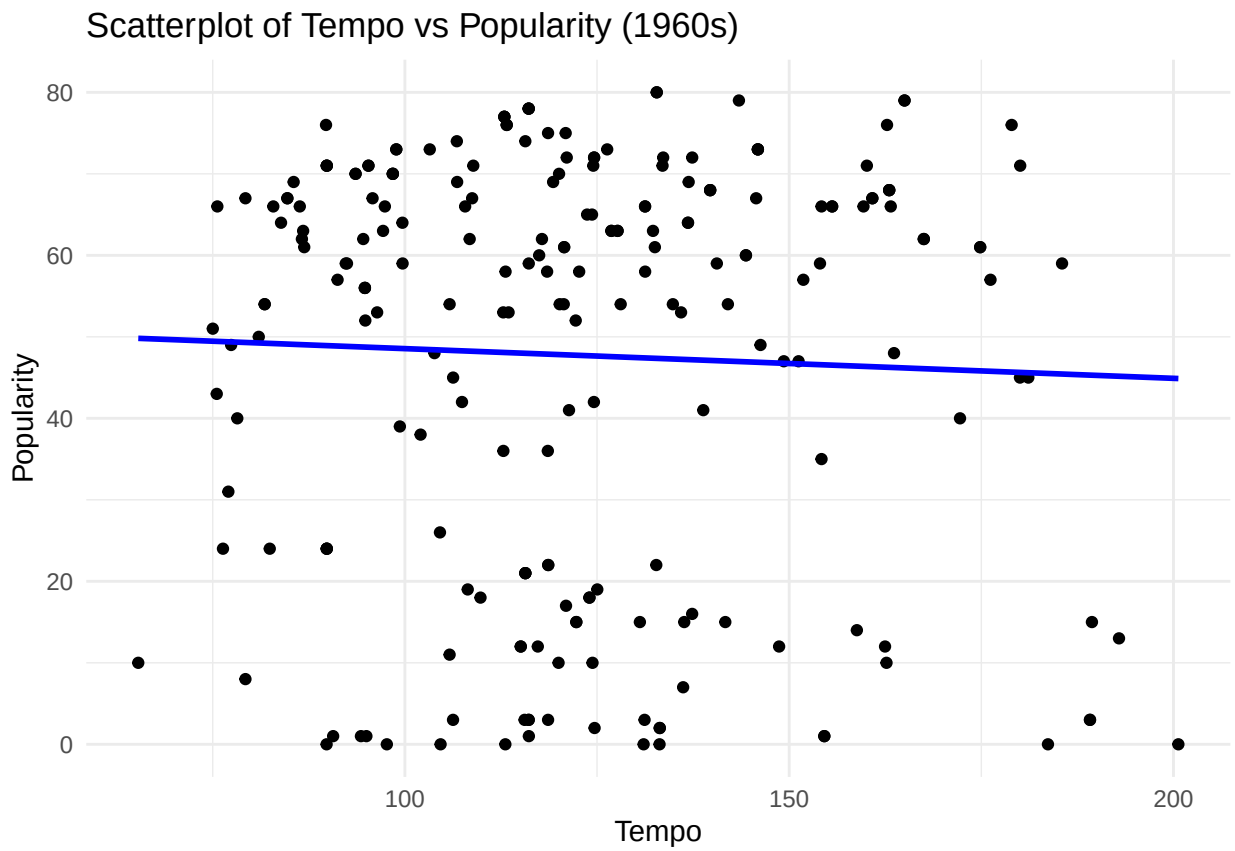

Scatterplot of Danceability vs Popularity (2010s)
Popularity
100
75
50
25
0
0.25
0.50
0.75
1.00
Danceability

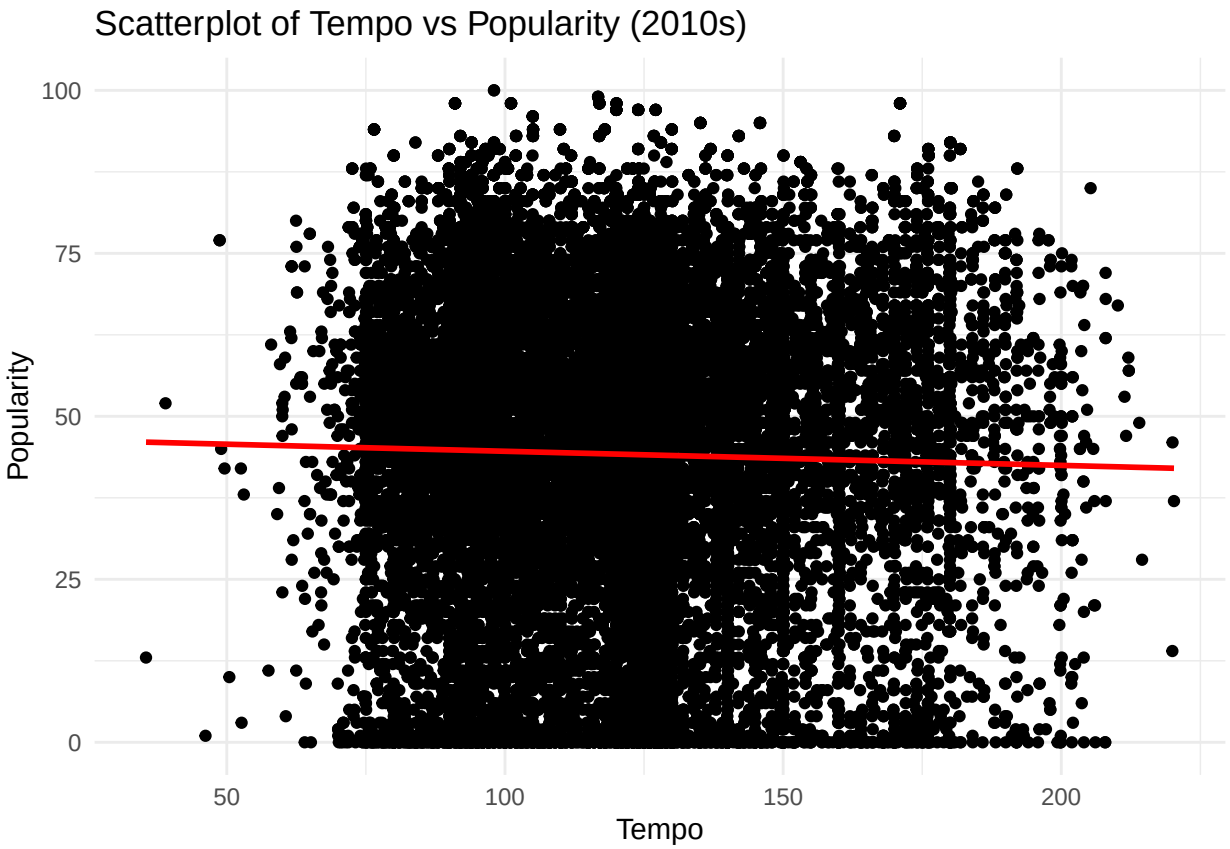

Scatterplot of Loudness vs Popularity (1960s)
Popularity
80
60
40
20
0
−20
−15
−10
−5
Loudness


Scatterplot of Loudness vs Popularity (2010s)
Popularity
100
75
50
25
0
−40
−30
−20
−10
0
Loudness


Scatterplot of Duration vs Popularity (1960s)
Popularity
80
60
40
20
0
1e+05
2e+05
3e+05
4e+05
5e+05
Duration (ms)

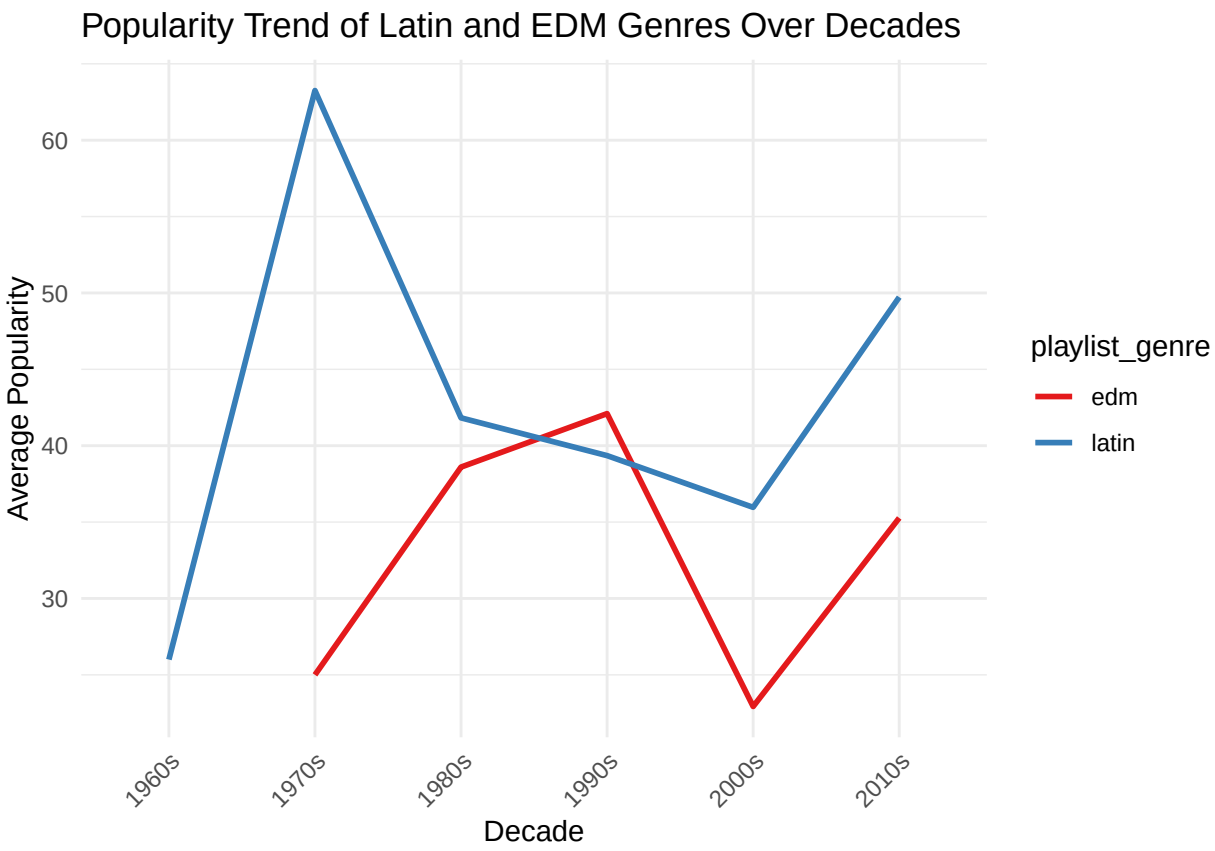

Scatterplot of Duration vs Popularity (2010s)
Popularity
100
75
50
25
0
1e+05
2e+05
3e+05
4e+05
5e+05
Duration (ms)


Scatterplot of Tempo vs Popularity (1960s)
Popularity
Tempo
80
60
40
20
0
100
150
200



How does the average popularity of Latin and EDM vary over time?

We chose to look at the average popularity of Latin and EDM because they showed up the most in the most/least popular genres over the decades graph. We wanted to look at these two to see if there was any growth or decline in their popularity over the years. In the graph below, We can see that EDM had no songs in the 1960s while Latin did. In 1970 Latin had its high in popularity and then went down after that. It declined to 2000s until it went up again. For EDM, it had its high in the 1990s until it declined down to its new low in the 2000s and then started climbing up again. These two go up and down seemingly randomly but it could be due to many different factors. These factors could be the audience changing their preferences over the decades or new artist coming out with songs.



Conclusion

These Spotify songs have many different factors that contribute to their popularity. Although this database covers a lot of factors for these songs, it does not cover nearly all of the attributes like audience or song artist. With this database, one major contributor is the audience's preference. We saw with the most/least popular genre graph that the most popular genres would change over the decades. This is because of many factors but a leading one would be the audience's preferences. The audience can change what they like over the decades and what they consider to be hot in the decade. Of the factors contained in this database, the biggest one that is positive for popularity is danceability. Songs with more danceability are more likely to be more popular in 2010 and that is a fashion that has changed with the decades. If a song wants to be popular it must keep up with what is a new trend and most follow the trend of the decade.

Code Appendix

```
options(repos = c(CRAN = "https://cran.rstudio.com"))
# Install and load required packages
install.packages(c("tidyverse", "dplyr", "ggplot2", "lubridate"))
library(tidyverse)
library(dplyr)
library(ggplot2)
library(lubridate)

# Load data from GitHub
```

```r
github_url <- "https://raw.githubusercontent.com/Stat184-Fall2023/FP-LiamHooks/main/Spotify%20Songs%20N
spotify_data <- read.csv(github_url)

head(spotify_data)

################################################
# Decade Frequency Table
# Install and load required packages
packages_to_install <- c("tidyverse", "dplyr", "ggplot2", "lubridate", "knitr", "kableExtra")

for (package in packages_to_install) {
  if (!requireNamespace(package, quietly = TRUE)) {
    install.packages(package)
  }
}

# Load libraries
library(tidyverse)
library(dplyr)
library(ggplot2)
library(lubridate)
library(knitr)
library(kableExtra)  # Load kableExtra package

# Load data from GitHub
github_url <- "https://raw.githubusercontent.com/Stat184-Fall2023/FP-LiamHooks/main/Spotify%20Songs%20N
spotify_data <- read.csv(github_url)

# Statistical Chart for Decades
filtered_data_decades <- spotify_data %>%
  filter(Cleaned.Up.Release.Date > 1960 & Cleaned.Up.Release.Date < 2020)

summary_stats_decades <- filtered_data_decades %>%
  mutate(Decade = cut(Cleaned.Up.Release.Date, breaks = seq(1960, 2020, by = 10),
                      labels = c("1960s", "1970s", "1980s", "1990s", "2000s", "2010s"))) %>%
  group_by(Decade) %>%
  summarise(
    n = n(),
    min = min(track_popularity),
    Q1 = quantile(track_popularity, probs = 0.25),
    Median = median(track_popularity),
    Q3 = quantile(track_popularity, probs = 0.75),
    Max = max(track_popularity)
  ) %>%
  ungroup()

# Print nicely formatted table with center alignment and caption
kable(summary_stats_decades, "latex", booktabs = TRUE, escape = FALSE, align = "c", caption = "Decades I



################################################
# Genre Frequency Table
```

```r
filtered_data_genres <- spotify_data %>%
  filter(Cleaned.Up.Release.Date >= 1960 & Cleaned.Up.Release.Date <= 2019)


summary_stats_genres <- filtered_data_genres %>%
  group_by(playlist_genre) %>%
  summarise(
    n = n(),
    min = min(track_popularity),
    Q1 = quantile(track_popularity, probs = 0.25),
    Median = median(track_popularity),
    Q3 = quantile(track_popularity, probs = 0.75),
    Max = max(track_popularity)
  ) %>%
  ungroup()


print(summary_stats_genres)

##############################################
# Bar Chart for Genres

# Create a decade variable with custom labels
spotify_data$Decade <- cut(spotify_data$Cleaned.Up.Release.Date, breaks = seq(1950, 2020, by = 10),
                           labels = c("1950s", "1960s", "1970s", "1980s", "1990s", "2000s", "2010s"))

# Create a summary table with average popularity for each genre in each decade
popularity_summary <- spotify_data %>%
  group_by(Decade, playlist_genre) %>%
  summarise(Avg_Popularity = mean(track_popularity))

# Identify the highest and lowest popularity genre for each decade
top_genre <- popularity_summary %>%
  group_by(Decade) %>%
  filter(Avg_Popularity == max(Avg_Popularity))

bottom_genre <- popularity_summary %>%
  group_by(Decade) %>%
  filter(Avg_Popularity == min(Avg_Popularity))

# Combine the highest and lowest popularity genres
top_bottom_genres <- bind_rows(top_genre, bottom_genre)

# Exclude "1950s" and "N/A" from the data
top_bottom_genres <- top_bottom_genres %>%
  filter(Decade %in% c("1960s", "1970s", "1980s", "1990s", "2000s", "2010s"))

# Plot the bar chart
ggplot(top_bottom_genres, aes(x = Decade, y = Avg_Popularity, fill = playlist_genre)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), width = 0.7) +
  labs(title = "Highest and Lowest Popularity Genre for Each Decade",
       x = "Decade",
       y = "Average Popularity") +
```

```r
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")

########################################
# Scatter Plots for danceability, tempo, loudness, duration

# Filter data for the 1960s decade
songs_1960s <- spotify_data %>%
  filter(Decade == "1960s")

# Filter data for the 2010s decade
songs_2010s <- spotify_data %>%
  filter(Decade == "2010s")

# Danceability 1960
ggplot(songs_1960s, aes(x = danceability, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Scatterplot of Danceability vs Popularity (1960s)",
       x = "Danceability",
       y = "Popularity") +
  theme_minimal()


# Danceability 2010
ggplot(songs_2010s, aes(x = danceability, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatterplot of Danceability vs Popularity (2010s)",
       x = "Danceability",
       y = "Popularity") +
  theme_minimal()


# Loudness 1960
ggplot(songs_1960s, aes(x = loudness, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Scatterplot of Loudness vs Popularity (1960s)",
       x = "Loudness",
       y = "Popularity") +
  theme_minimal()


# Loudness 2010
ggplot(songs_2010s, aes(x = loudness, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatterplot of Loudness vs Popularity (2010s)",
       x = "Loudness",
       y = "Popularity") +
  theme_minimal()
```

```r
# Duration 1960
ggplot(songs_1960s, aes(x = duration_ms, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Scatterplot of Duration vs Popularity (1960s)",
       x = "Duration (ms)",
       y = "Popularity") +
  theme_minimal()


# Duration 2010
ggplot(songs_2010s, aes(x = duration_ms, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatterplot of Duration vs Popularity (2010s)",
       x = "Duration (ms)",
       y = "Popularity") +
  theme_minimal()


# Tempo 1960
ggplot(songs_1960s, aes(x = tempo, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Scatterplot of Tempo vs Popularity (1960s)",
       x = "Tempo",
       y = "Popularity") +
  theme_minimal()


# Tempo 2010
ggplot(songs_2010s, aes(x = tempo, y = track_popularity)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatterplot of Tempo vs Popularity (2010s)",
       x = "Tempo",
       y = "Popularity") +
  theme_minimal()

#####################################################
# Trend Chart for Latin and EDM

# Statistical Chart for Decades
filtered_data_decades <- spotify_data %>%
  filter(Cleaned.Up.Release.Date > 1960 & Cleaned.Up.Release.Date < 2020)

summary_stats_decades <- filtered_data_decades %>%
  mutate(Decade = cut(Cleaned.Up.Release.Date, breaks = seq(1960, 2020, by = 10),
                      labels = c("1960s", "1970s", "1980s", "1990s", "2000s", "2010s"))) %>%
  group_by(Decade) %>%
  summarise(
    n = n(),
```

```
    min = min(track_popularity),
    Q1 = quantile(track_popularity, probs = 0.25),
    Median = median(track_popularity),
    Q3 = quantile(track_popularity, probs = 0.75),
    Max = max(track_popularity)
  ) %>%
  ungroup()
```