



Home

About

Contact



E X P L O R A T O R Y   D A T A   A N A L Y S I S

# SPOTIFY

Most Streaming Trends

Get Started

01

02

03

Sakshi Saraiya - 24011



[Home](#)[About](#)[Contact](#)

# Introduction

## OVERVIEW

This project explores the Spotify dataset to uncover patterns in music streams, track releases, and artist collaborations, leveraging data visualization and statistical analysis.

[Learn More](#)

01

02

03



[Home](#)[About](#)[Contact](#)

# Objectives of the Project

- Understand Streaming Trends
- Analyze Artist Impact
- Discover Seasonal Trends
- Correlation Analysis
- Track Longevity

[Learn More](#)

01

02

03





# Research Methodology

- **Data Collection:**

Utilized Spotify's dataset containing track details, artist information, release years, and streaming statistics.

- **Data Cleaning:**

Removed missing values and created new metrics (e.g., release lifespan).

- **Statistical Tools:**

Descriptive Analysis: Summary statistics for streams, artists, and release patterns.

Visualizations: Bar charts, pie charts, density plots, and box plots for insights.

Correlation Analysis: Examined relationships between streams , playlist and artist count.

- **Software Used:**

Analysis and visualizations were performed using R programming language.



01



02

03





# Data and Variables

Dataset: Spotify data

- Total Observations: 953
- Key Columns:
  - Streams: Number of streams per track.
  - In\_spotify\_playlists: Count of playlists a track appears in.
  - Artist\_count: Number of artists on the track.
  - Released\_year and released\_month: Year and month of release.



01

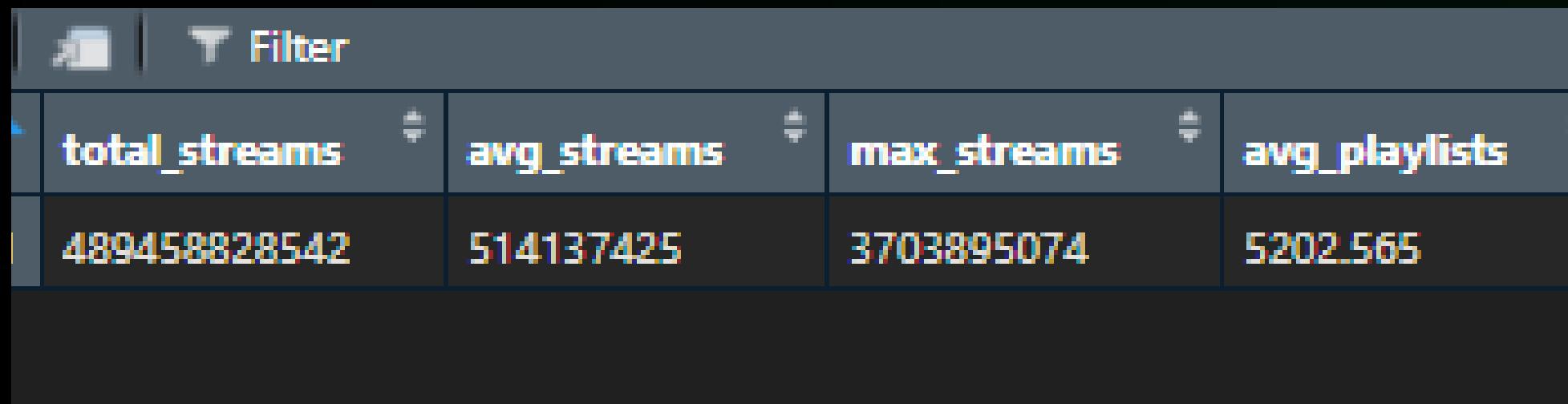
02

03





# Overall insights into streams, playlists



total_streams	avg_streams	max_streams	avg_playlists
489458828542	514137425	3703895074	5202.565

[Learn More](#)

```
01
spotify_data<-spotify_clean %>%
  summarise(
    total_streams = sum(streams, na.rm = TRUE),
    avg_streams = mean(streams, na.rm = TRUE),
    max_streams = max(streams, na.rm = TRUE),
    avg_playlists = mean(in_spotify_playlists,
    na.rm = TRUE) )
  view(spotify_data)
```

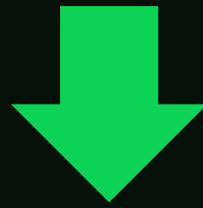
02

01



# Key Analysis

## TOP ARTISTS



	artist.s._name	total_streams
1	The Weeknd	14185552870
2	Taylor Swift	14053658300
3	Ed Sheeran	13908947204
4	Harry Styles	11608645649
5	Bad Bunny	9997799607
6	Olivia Rodrigo	7442148916

```
top_artists <- spotify_clean%>%
  group_by(artist.s._name) %>%
  summarise(total_streams =
    sum(streams, na.rm = TRUE)) %>%
  arrange(desc(total_streams))
  view(head(top_artists))
```

01

02

03



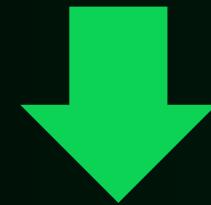


# Key Analysis

```
top_charted_tracks <- spotify_clean %>%
  arrange(desc(in_spotify_charts)) %>%
  select(track_name, artist.s._name,
        in_spotify_charts)
view(head(top_charted_tracks,10))
```

[Learn More](#)

## TOP TRACKS



	track_name	artist.s._name
1	Seven (feat. Latto) (Explicit Ver.)	Latto, Jung Kook
2	As It Was	Harry Styles
3	Flowers	Miley Cyrus
4	vampire	Olivia Rodrigo
5	I Wanna Be Yours	Arctic Monkeys
6	What Was I Made For? [From The Motion Picture "Barbie"]	Billie Eilish

01

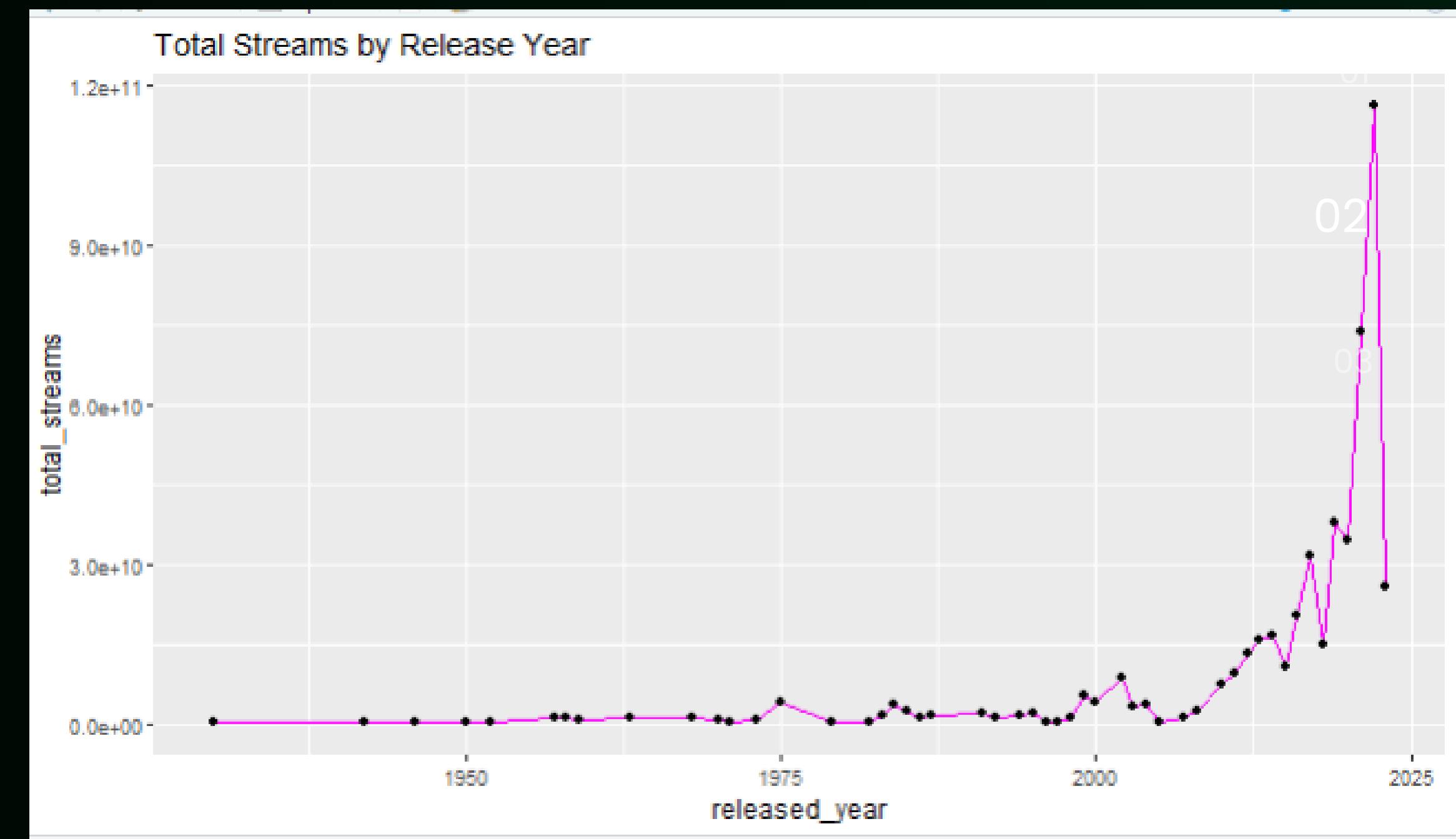
02

03



# Stream Trends by Year

- Tracks released before the 2000s exhibit minimal growth in total streams
- There is a significant spike in total streams for tracks released post-2015
- Tracks released during 2023-2024 show the highest total streams, indicating a preference for recent tracks





# Most Productive Artist

```
most_productive_artists <-
  spotify_clean %>%
  group_by(artist.s._name) %>%
  summarise(track_count = n()) %>%
  arrange(desc(track_count))
view(head(most_productive_artists))
```

[Learn More](#)

	artist.s._name	track_count
1	Taylor Swift	34
2	The Weeknd	22
3	Bad Bunny	19
4	SZA	19
5	Harry Styles	17
6	Kendrick Lamar	12

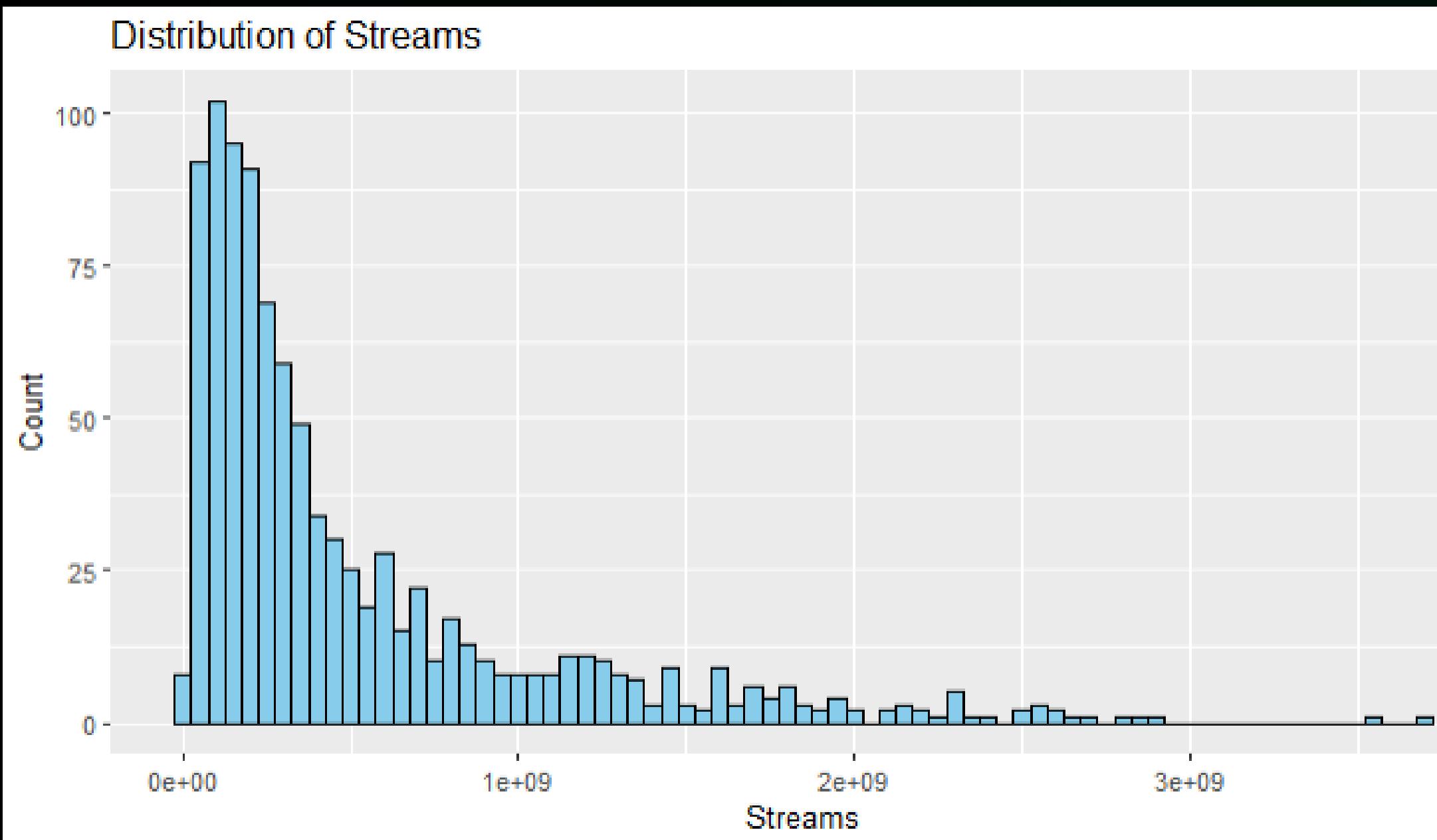
01

02

03



# Histogram: Distribution of streams



- 01
- The histogram is heavily right-skewed, indicating that most tracks have a relatively low number of streams, while only a few tracks achieve very high stream counts.
  - The count of tracks decreases progressively as the number of streams increases, suggesting that only a limited number of tracks achieve viral success or become hits.
- 03





# Artist Collaboration Impact



```
collab_streams <- spotify_clean %>%
  group_by(artist_count) %>%
  summarise(avg_streams = mean(streams, na.rm = TRUE))
  View(head(collab_streams,10))
```

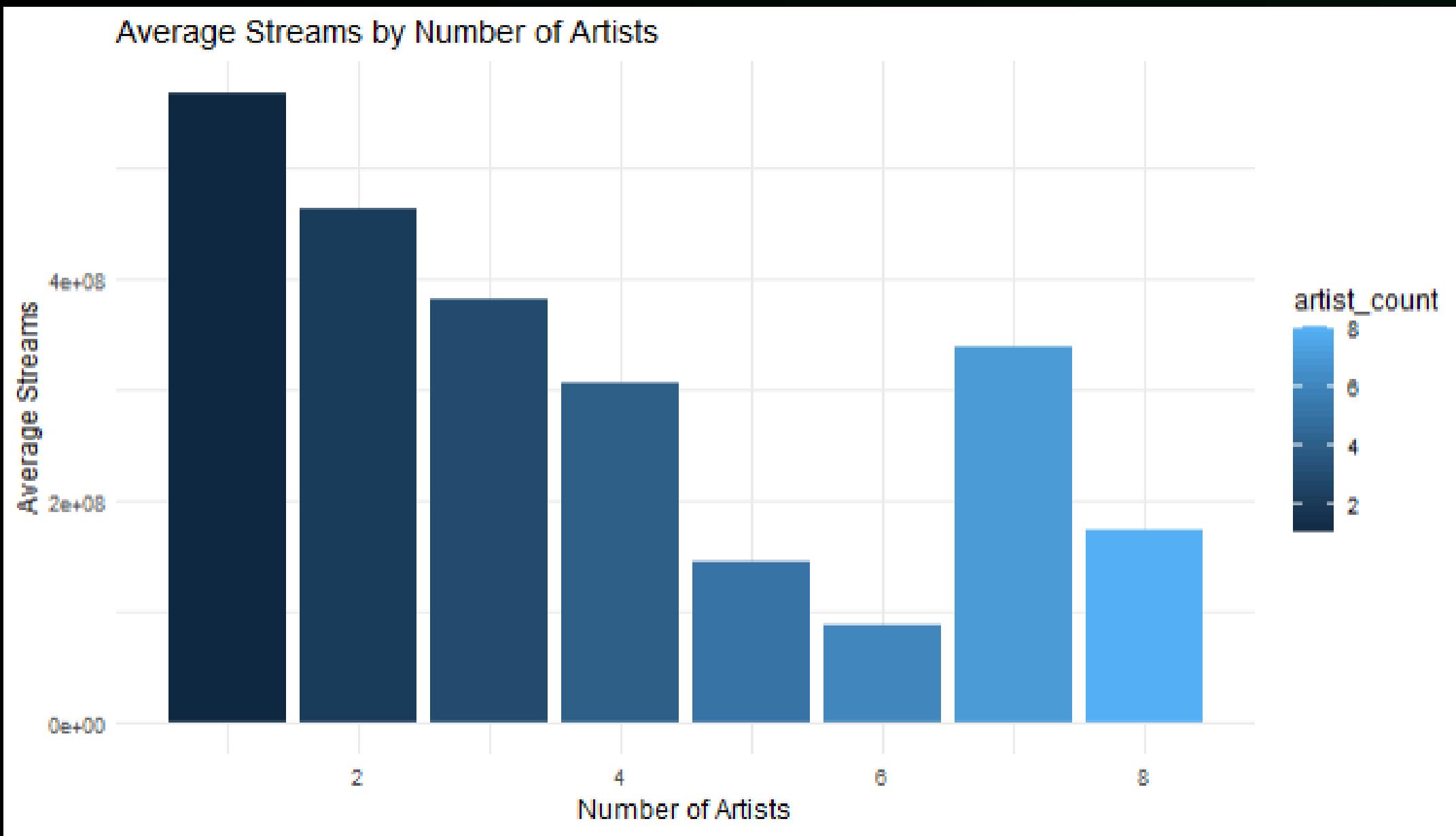
**Tracks with higher artist counts tend to have less streams.**

**Collaborations does not enhance track popularity, contributing to higher stream counts.**

artist_count	avg_streams
1	568211662
2	462480786
3	381342098
4	306106246
5	144808200
6	87466453
7	339060068
8	173221174



# Artist Collaboration Impact



01



- Tracks with a single artist have the highest average streams.
- Tracks with 2 to 4 artists show a gradual decline in average streams
- A significant increase in average streams is observed for tracks featuring 6 artists. This could represent popular collaborations with multiple well-known artists.

[Learn More](#)



# Playlist Performance

	track_name	artists_name	playlists_per_stream
1	Que Vuelvas	Carin Leon, Grupo Frontera	2.762491e-01
2	Jhoome Jo Pathaan	Arijit Singh, Vishal Dadlani, Sukriti Kakar, Vishal-Shekhar, Sh...	1.010853e-04
3	Get Lucky - Radio Edit	Pharrell Williams, Nile Rodgers, Daft Punk	5.664716e-05
4	Mother I Sober (feat. Beth Gibbons of Portishead)	Kendrick Lamar, Beth Gibbons	5.661826e-05
5	The Heart Part 5	Kendrick Lamar	4.114902e-05
6	Auntie Diaries	Kendrick Lamar	4.089661e-05
7	(It Goes Like) Nanana - Edit	Peggy Gou	3.903143e-05
8	Rush	Troye Sivan	3.826198e-05
9	QUEMA	Sog, Ryan Castro, Peso Pluma	3.767440e-05
10	The Next Episode	Dr. Dre, Snoop Dogg	3.766354e-05

```

playlist_performance <- spotify_clean
%>%
  mutate(playlists_per_stream =
    in_spotify_playlists / streams) %>%
  arrange(desc(playlists_per_stream))
%>%
  select(track_name, artist.s._name,
    playlists_per_stream)
View(head(playlist_performance,10))
  
```

[Learn More](#)



	released_month	tracks_released
1	1	133
2	2	61
3	3	86
4	4	66
5	5	128
6	6	12
7	7	62
8	8	46
9	9	56
10	10	73
11	11	80
12	12	75

# Monthly Track Release Trends

01

02

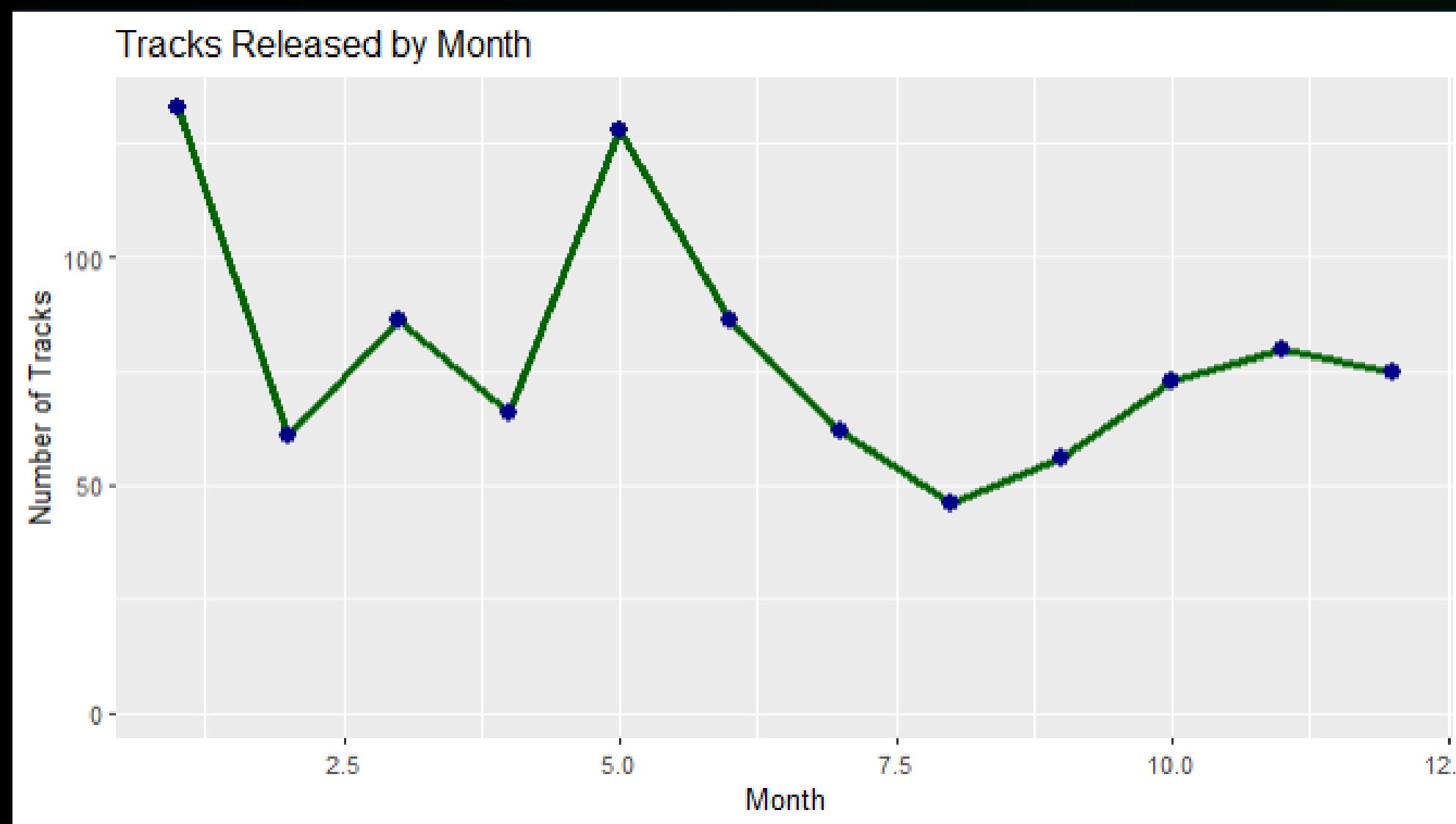
03

```
monthly_releases <- spotify_clean %>%
  group_by(released_month) %>%
  summarise(track_count = n())
View(head(monthly_releases,10))
```



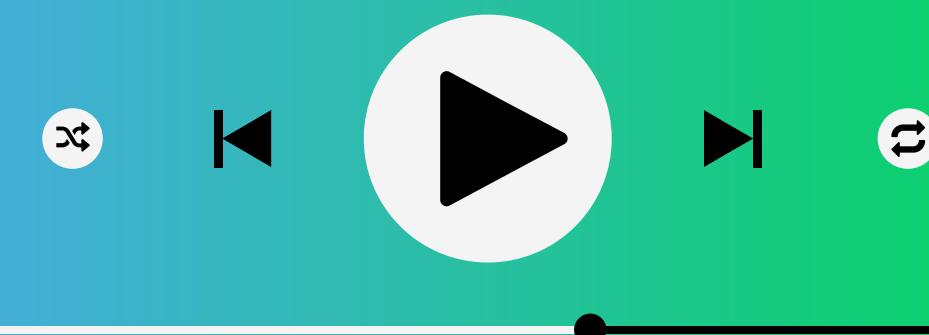


# Monthly Track Release Trends



```
ggplot(monthly_releases, aes(x =  
released_month, y = track_count)) +  
geom_line(color = "darkgreen", size =  
1.2) + geom_point(size = 3, color =  
"darkblue") + labs(title="Tracks  
Released by  
Month",x="Month",y="Number of Tracks")
```

- Maximum number of tracks are released in the month of January, followed by May
- Least number of songs are released in the month of August





# Average Streams by Artist Count

	artist_count	avg_streams	track_count
1	1	568211662	586
2	2	462480786	254
3	3	381342098	85
4	7	339060068	2
5	4	306106246	15
6	8	173221174	2
7	5	144808200	5
8	6	87466453	3



```
streams_by_artist_count <-
  spotify_clean%>%
  group_by(artist_count) %>%
  summarise( avg_streams =
    mean(streams, na.rm = TRUE),
    track_count = n() ) %>%
  arrange(desc(avg_streams))
View(head(streams_by_artist_count,10))
```

01

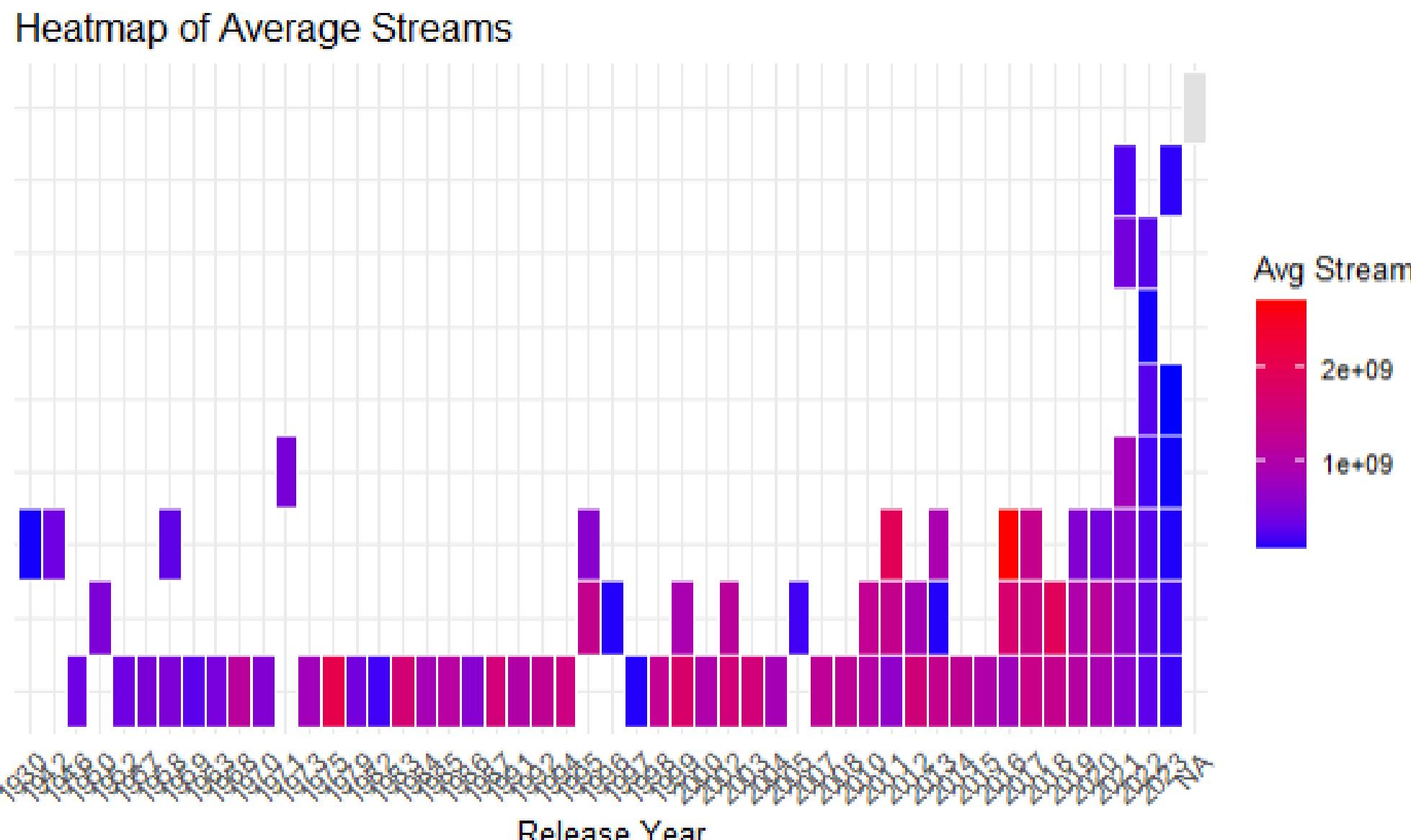
02

03





# Heatmap : Average Streams



- Heatmap showing average streams by release year and artist count.
- Tracks with higher artist counts in certain years achieve higher average streams.
- Collaboration trends and release timing significantly affect streaming performance.





# Most Streamed Tracks

	track_name	artist.s._name	streams
1	Blinding Lights	The Weeknd	3703895074
2	Shape of You	Ed Sheeran	3562543890
3	Someone You Loved	Lewis Capaldi	2887241814
4	Dance Monkey	Tones and I	2864791672
5	Sunflower - Spider-Man: Into the Spider-Verse	Post Malone, Swae Lee	2808096550
6	One Dance	Drake, WizKid, Kyla	2713922350
7	STAY (with Justin Bieber)	Justin Bieber, The Kid Laroi	2665343922
8	Believer	Imagine Dragons	2594040133
9	Closer	The Chainsmokers, Halsey	2591224264
10	Starboy	The Weeknd, Daft Punk	2565529693

01

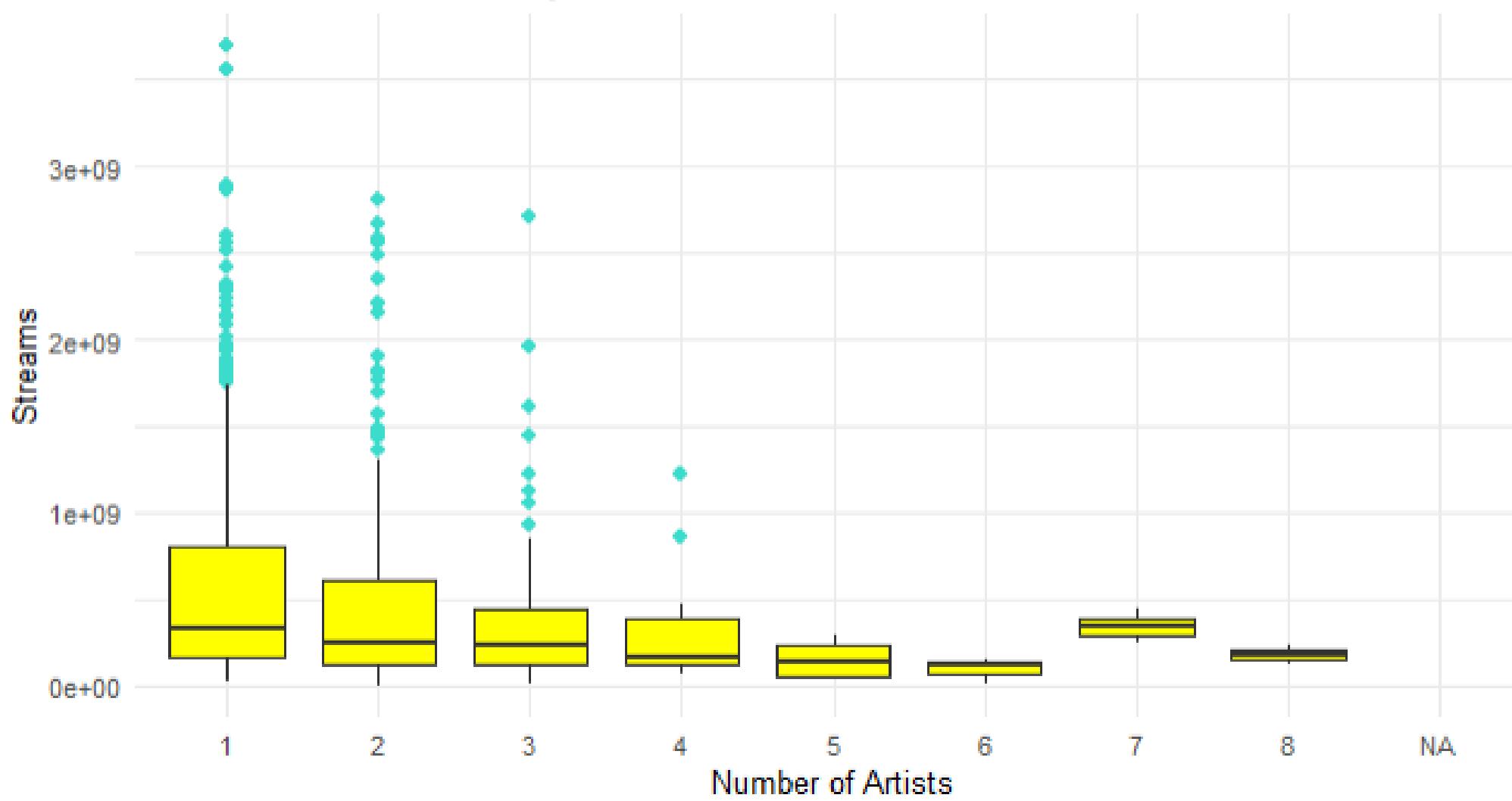
```
most_streamed_tracks <-
  spotify_clean %>%
  arrange(desc(streams)) %>%
  select(track_name, artist.s._name,
        streams)
view(head(most_streamed_tracks,10))
```

[Learn More](#)



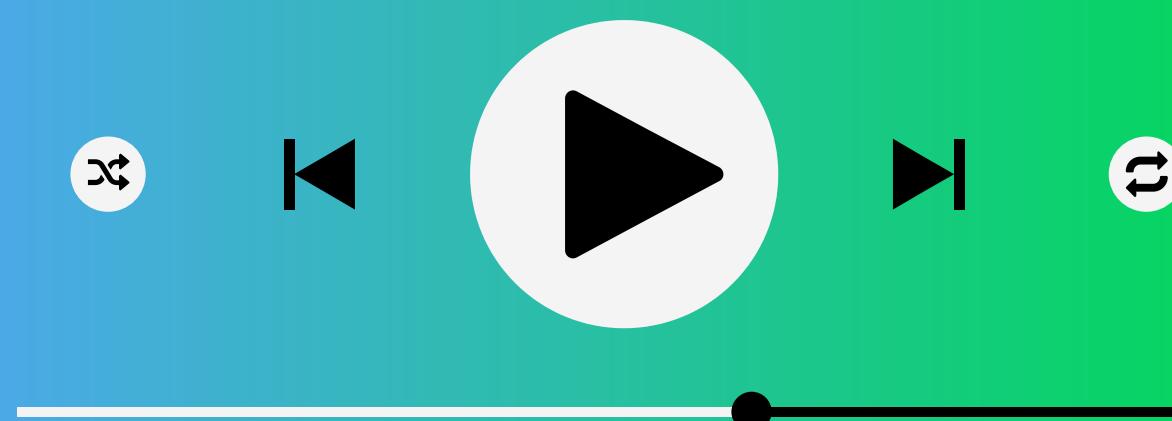
# Box Plot: Streams by Artist Count

Distribution of Streams by Artist Count



- **Popularity of Solo and Duo Tracks:** Tracks with one or two artists dominate the streaming performance, including a significant number of blockbuster tracks with billions of streams.

- **Consistency in Group Tracks:** Tracks with multiple artists exhibit smaller variability in streams, suggesting they cater to more niche or specific audiences.



01

02

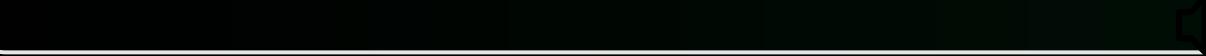
03



# Pie Chart

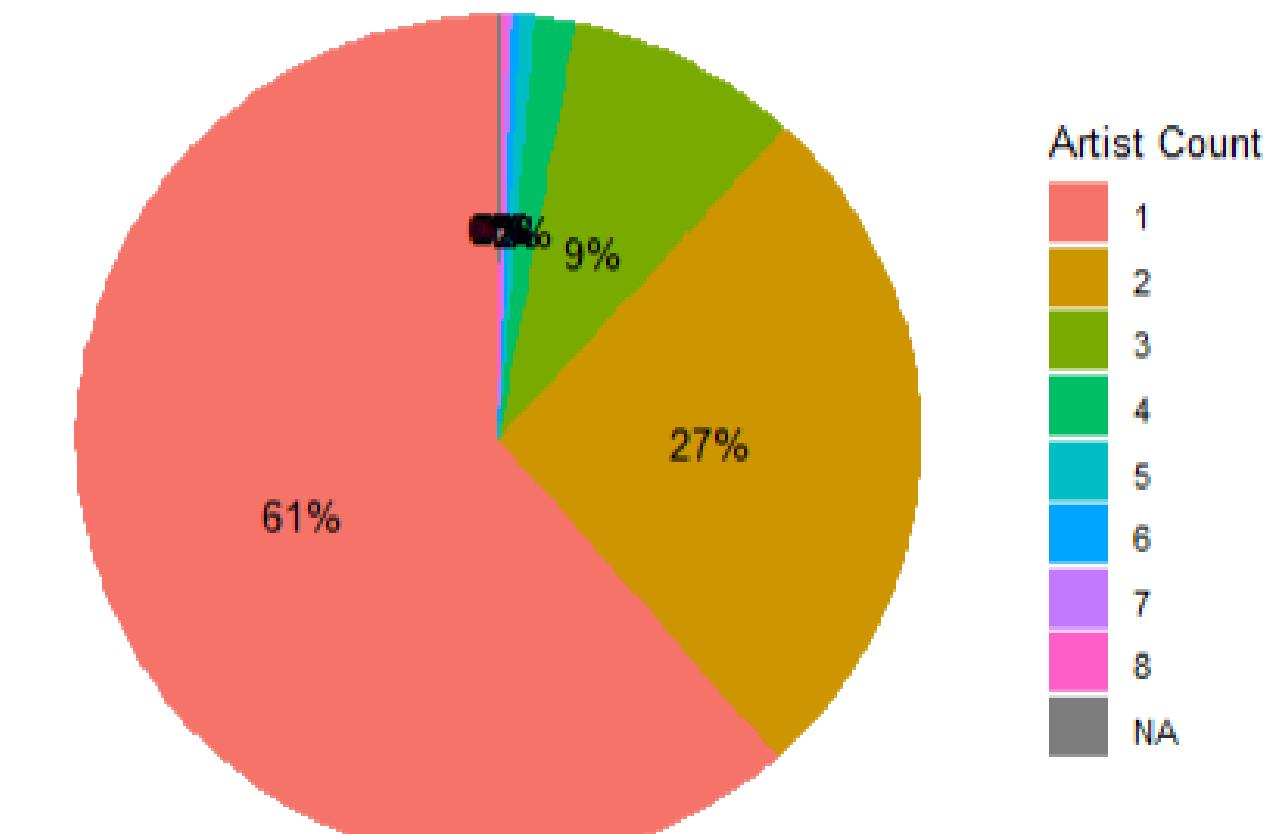


```
ggplot(artist_count_data, aes(x = "", y = percentage, fill =  
                                factor(artist_count))) +  
  geom_bar(stat = "identity", width = 1) +  
  coord_polar(theta = "y") +  
  geom_text(aes(label = paste0(round(percentage, 0), "%")),  
            position = position_stack(vjust = 0.5)) +  
  labs(title = "Proportion of Tracks by Artist Count", fill = "Artist Count") +  
  theme_void()
```



The largest slice of the pie represents a number of tracks with only 1 artist, comprising a substantial 61% of all tracks, indicating that tracks by solo artists are preferred more.

Proportion of Tracks by Artist Count





# Track Lifespan Analysis

```
spotify_clean <- spotify_clean %>%
  mutate(release_lifespan = 2024 - released_year)
%>%
  arrange(desc(release_lifespan))
```

```
long_lifespan_tracks <- spotify_clean %>%
  select(track_name, artist.s._name,
         release_lifespan)
view(head(long_lifespan_tracks))
```

01

02

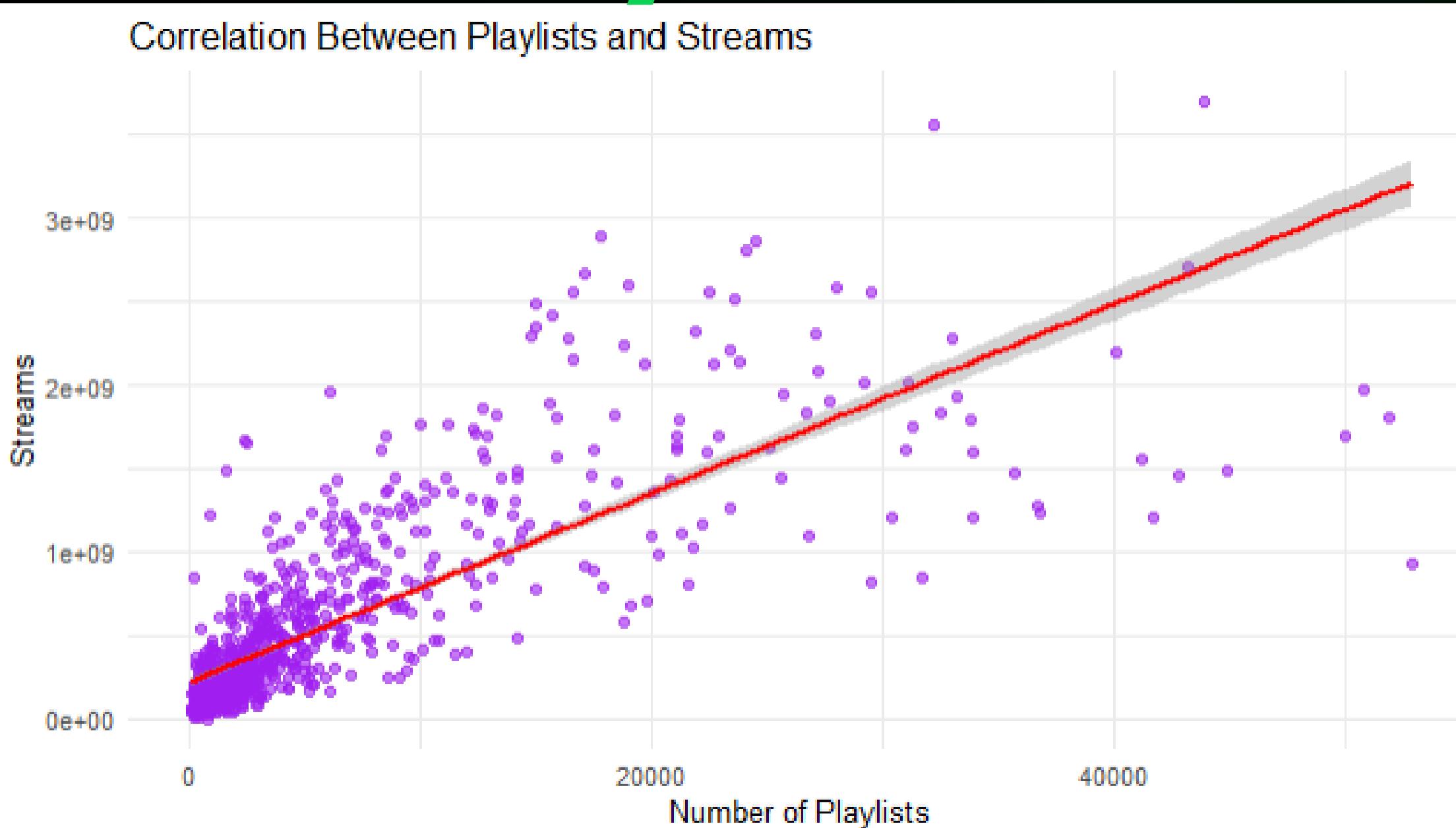
03

	track_name	artist.s._name	release_lifespan
1	Agudo Meogi	Styx, utku INC, Thezth	94
2	White Christmas	Bing Crosby, John Scott Trotter & His Orchestra, Ken Darby ...	82
3	The Christmas Song (Merry Christmas To You) - Remastered ...	Nat King Cole	78
4	Let It Snow! Let It Snow! Let It Snow!	Frank Sinatra, B. Swanson Quartet	74
5	A Holly Jolly Christmas - Single Version	Burl Ives	72
6	Jingle Bell Rock	Bobby Helms	67



# Scatter plot : Playlists vs. Streams

Correlation Between Playlists and Streams

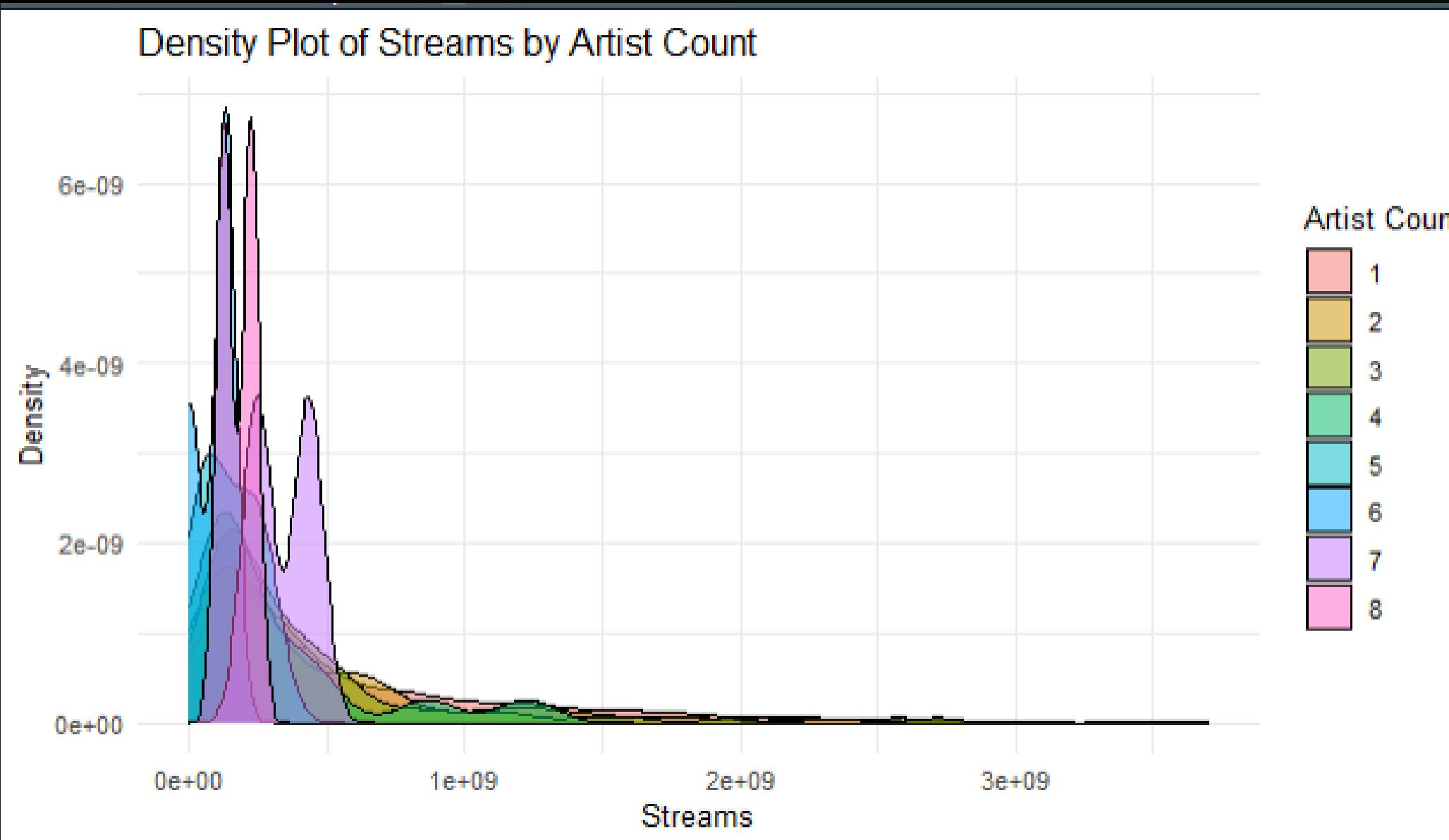


```
ggplot(spotify_clean, aes(x =  
in_spotify_playlists, y = streams)) +  
  geom_point(alpha = 0.6, color = "blue") +  
  geom_smooth(method = "lm", color =  
"pink") +  
  labs(title = "Correlation Between Playlists  
and Streams", x = "Number of Playlists",  
y = "Streams") +  
  theme_minimal()
```

**Positive correlation implies  
playlists boost track visibility  
and streams.**

[Learn More](#)

# Density Plot



```
ggplot(spotify_clean, aes(x = streams, fill = factor(artist_count))) +  
  geom_density(alpha = 0.5) +  
  labs( title = "Density Plot of Streams by Artist Count",  
        x = "Streams",  
        y = "Density",  
        fill = "Artist Count" ) +  
  theme_minimal()
```

The density plot shows a concentration of tracks with low to moderate streams, but collaborative tracks slightly shift the density towards higher streams.



# Key Findings

## 1. Stream Trends by Year

- Newer tracks attract more streams, indicating growing platform engagement.

## 2. Top Artists and Tracks

- A small group of artists dominates streams, showcasing their consistent popularity.

## 3. Track Lifespan

- Older tracks with enduring popularity continue to generate significant streams.

## 4. Collaboration Insights

- Collaborative tracks outperform solo artist tracks in total streams.

## 5. Playlist Influence

- Strong correlation between playlist inclusion and total streams, highlighting their promotional impact.



01





# Conclusion

- New Tracks Dominate: The trend shows newer releases are gaining the most traction.
- Collaborations Matter: Tracks featuring multiple artists leverage wider audiences and drive higher engagement.
- Playlists as a Key Driver: Playlists significantly influence visibility and streaming success.
- Strategy Implications: Artists and producers should focus on collaboration, inclusion in playlists, and strategic timing to maximize track reach.

01

02

03





Home

About

Contact



# THANK YOU

01

02

03

