Spotify Unwrapped: Trends and Patterns in Streaming

About the Data

Spotify is a audio streaming platform that started in 2006 by Daniel Ek and Martin Lorentzon. It is one of the largest music streaming services in the world,offering a vast library of over 100 million songs and more than 5 million podcasts. The platform is known for its personalized recommendations,including curated playlists like Discover Weekly and Release Radar, as well as real-time music charts.Spotify has revolutionized how people consume music and has become a dominant player in the global music industry, holding 31.7% of market share in music streaming space.The dataset used in this project has information about the most streamed songs on Spotify. The analysis focuses only on data from past ten years (2013-23). This report explores:

- Trends in popular music over time,
- Popular Artist across different years,
- Most streamed and saved songs and
- Correlation between musical attributes and song popularity.

Installations

Setting up environment by Installing & loading packages for libraries that are used in processing, analysis and visualization of the data.

```
# load packages
library(formatR)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggplot2)
library(knitr)
```

Read and Store Data

Store data in an variable to perform further actions of data cleaning. **Note:** Spotify Dataset used in this analysis is available on Kaggle.com.

```
data = read_csv("spotify_data.csv")
```

```
## Rows: 953 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): track_name, artist(s)_name, streams, key, mode
## dbl (17): artist_count, released_year, released_month, released_day, in_spot...
## num  (2): in_deezer_playlists, in_shazam_charts
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(data)
```

```
##   track_name        artist(s)_name      artist_count   released_year
##  Length:953         Length:953         Min.   :1.000   Min.   :1930
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:2020
##  Mode  :character   Mode  :character   Median :1.000   Median :2022
##                                        Mean   :1.556   Mean   :2018
##                                        3rd Qu.:2.000   3rd Qu.:2022
##                                        Max.   :8.000   Max.   :2023
##
##  released_month    released_day   in_spotify_playlists in_spotify_charts
##  Min.   : 1.000   Min.   : 1.00   Min.   :   31        Min.   :  0.00
##  1st Qu.: 3.000   1st Qu.: 6.00   1st Qu.:  875        1st Qu.:  0.00
##  Median : 6.000   Median :13.00   Median : 2224        Median :  3.00
##  Mean   : 6.034   Mean   :13.93   Mean   : 5200        Mean   : 12.01
##  3rd Qu.: 9.000   3rd Qu.:22.00   3rd Qu.: 5542        3rd Qu.: 16.00
##  Max.   :12.000   Max.   :31.00   Max.   :52898        Max.   :147.00
##
##    streams          in_apple_playlists in_apple_charts  in_deezer_playlists
##  Length:953         Min.   :  0.00     Min.   :  0.00   Min.   :    0.0
##  Class :character   1st Qu.: 13.00     1st Qu.:  7.00   1st Qu.:   13.0
##  Mode  :character   Median : 34.00     Median : 38.00   Median :   44.0
##                     Mean   : 67.81     Mean   : 51.91   Mean   :  385.2
##                     3rd Qu.: 88.00     3rd Qu.: 87.00   3rd Qu.:  164.0
##                     Max.   :672.00     Max.   :275.00   Max.   :12367.0
##
##  in_deezer_charts in_shazam_charts      bpm            key
##  Min.   : 0.000   Min.   :   0     Min.   : 65.0   Length:953
##  1st Qu.: 0.000   1st Qu.:   0     1st Qu.:100.0   Class :character
##  Median : 0.000   Median :   2     Median :121.0   Mode  :character
##  Mean   : 2.666   Mean   :  60     Mean   :122.5
##  3rd Qu.: 2.000   3rd Qu.:  37     3rd Qu.:140.0
##  Max.   :58.000   Max.   :1451     Max.   :206.0
##                   NA's   :50
##      mode           danceability_%    valence_%        energy_%
##  Length:953         Min.   :23.00   Min.   : 4.00   Min.   : 9.00
##  Class :character   1st Qu.:57.00   1st Qu.:32.00   1st Qu.:53.00
##  Mode  :character   Median :69.00   Median :51.00   Median :66.00
##                     Mean   :66.97   Mean   :51.43   Mean   :64.28
```

```
##                     3rd Qu.:78.00   3rd Qu.:70.00   3rd Qu.:77.00
##                     Max.   :96.00   Max.   :97.00   Max.   :97.00
##
##  acousticness_%  instrumentalness_%   liveness_%    speechiness_%
##  Min.   : 0.00   Min.   : 0.000     Min.   : 3.00   Min.   : 2.00
##  1st Qu.: 6.00   1st Qu.: 0.000     1st Qu.:10.00   1st Qu.: 4.00
##  Median :18.00   Median : 0.000     Median :12.00   Median : 6.00
##  Mean   :27.06   Mean   : 1.581     Mean   :18.21   Mean   :10.13
##  3rd Qu.:43.00   3rd Qu.: 0.000     3rd Qu.:24.00   3rd Qu.:11.00
##  Max.   :97.00   Max.   :91.000     Max.   :97.00   Max.   :64.00
##
```

Understanding data type of variables in the dataset to perform operations.

```
glimpse(data)
```

```
## Rows: 953
## Columns: 24
## $ track_name           <chr> "Seven (feat. Latto) (Explicit Ver.)", "LALA", "v~
## $ `artist(s)_name`     <chr> "Latto, Jung Kook", "Myke Towers", "Olivia Rodrig~
## $ artist_count         <dbl> 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1~
## $ released_year        <dbl> 2023, 2023, 2023, 2019, 2023, 2023, 2023, 2023, 2~
## $ released_month       <dbl> 7, 3, 6, 8, 5, 6, 3, 7, 5, 3, 4, 7, 1, 4, 3, 12, ~
## $ released_day         <dbl> 14, 23, 30, 23, 18, 1, 16, 7, 15, 17, 17, 7, 12, ~
## $ in_spotify_playlists <dbl> 553, 1474, 1397, 7858, 3133, 2186, 3090, 714, 109~
## $ in_spotify_charts    <dbl> 147, 48, 113, 100, 50, 91, 50, 43, 83, 44, 40, 55~
## $ streams              <chr> "141381703", "133716286", "140003974", "800840817~
## $ in_apple_playlists   <dbl> 43, 48, 94, 116, 84, 67, 34, 25, 60, 49, 41, 37, ~
## $ in_apple_charts      <dbl> 263, 126, 207, 207, 133, 213, 222, 89, 210, 110, ~
## $ in_deezer_playlists  <dbl> 45, 58, 91, 125, 87, 88, 43, 30, 48, 66, 54, 21, ~
## $ in_deezer_charts     <dbl> 10, 14, 14, 12, 15, 17, 13, 13, 11, 13, 12, 5, 58~
## $ in_shazam_charts     <dbl> 826, 382, 949, 548, 425, 946, 418, 194, 953, 339,~
## $ bpm                  <dbl> 125, 92, 138, 170, 144, 141, 148, 100, 130, 170, ~
## $ key                  <chr> "B", "C#", "F", "A", "A", "C#", "F", "F", "C#", "~
## $ mode                 <chr> "Major", "Major", "Major", "Major", "Minor", "Maj~
## $ `danceability_%`     <dbl> 80, 71, 51, 55, 65, 92, 67, 67, 85, 81, 57, 78, 7~
## $ `valence_%`          <dbl> 89, 61, 32, 58, 23, 66, 83, 26, 22, 56, 56, 52, 6~
## $ `energy_%`           <dbl> 83, 74, 53, 72, 80, 58, 76, 71, 62, 48, 72, 82, 6~
## $ `acousticness_%`     <dbl> 31, 7, 17, 11, 14, 19, 48, 37, 12, 21, 23, 18, 6,~
## $ `instrumentalness_%` <dbl> 0, 0, 0, 0, 63, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 17,~
## $ `liveness_%`         <dbl> 8, 10, 31, 11, 11, 8, 8, 11, 28, 8, 27, 15, 3, 9,~
## $ `speechiness_%`      <dbl> 4, 4, 6, 15, 6, 24, 3, 4, 9, 33, 5, 7, 7, 3, 6, 4~
```

Data Cleaning

```
skim_without_charts(data)
```

```
## Warning: There was 1 warning in `dplyr::summarize()`.
## i In argument: `dplyr::across(tidyselect::any_of(variable_names),
##   mangled_skimmers$funs)`.
## i In group 0: .
## Caused by warning:
## ! There were 43 warnings in `dplyr::summarize()`.
## The first warning was:
## i In argument: `dplyr::across(tidyselect::any_of(variable_names),
```

```
##   mangled_skimmers$funs)`.
## Caused by warning in `grepl()`:
## ! unable to translate 'T <ef><bf>' to a wide string
## i Run `dplyr::last_dplyr_warnings()` to see the 42 remaining warnings.
```

Table 1: Data summary

Name	data
Number of rows	953
Number of columns	24
Column type frequency:	
character	5
numeric	19
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
track_name	0	1.0	2	123	0	943	0
artist(s)_name	0	1.0	1	117	0	645	0
streams	0	1.0	4	102	0	949	0
key	95	0.9	1	2	0	11	0
mode	0	1.0	5	5	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
artist_count	0	1.00	1.56	0.89	1	1	1	2	8
released_year	0	1.00	2018.24	11.12	1930	2020	2022	2022	2023
released_month	0	1.00	6.03	3.57	1	3	6	9	12
released_day	0	1.00	13.93	9.20	1	6	13	22	31
in_spotify_playlists	0	1.00	5200.12	7897.61	31	875	2224	5542	52898
in_spotify_charts	0	1.00	12.01	19.58	0	0	3	16	147
in_apple_playlists	0	1.00	67.81	86.44	0	13	34	88	672
in_apple_charts	0	1.00	51.91	50.63	0	7	38	87	275
in_deezer_playlists	0	1.00	385.19	1130.54	0	13	44	164	12367
in_deezer_charts	0	1.00	2.67	6.04	0	0	0	2	58
in_shazam_charts	50	0.95	60.00	161.16	0	0	2	37	1451
bpm	0	1.00	122.54	28.06	65	100	121	140	206
danceability_%	0	1.00	66.97	14.63	23	57	69	78	96
valence_%	0	1.00	51.43	23.48	4	32	51	70	97
energy_%	0	1.00	64.28	16.55	9	53	66	77	97
acousticness_%	0	1.00	27.06	26.00	0	6	18	43	97
instrumentalness_%	0	1.00	1.58	8.41	0	0	0	0	91
liveness_%	0	1.00	18.21	13.71	3	10	12	24	97
speechiness_%	0	1.00	10.13	9.91	2	4	6	11	64

```
df <- data %>%
    filter(released_year >= 2013)
```

Considering Period of 10 years

```
df <- na.omit(df)
skim_without_charts(df)
```

Remove Null Values

```
## Warning: There was 1 warning in `dplyr::summarize()`.
## i In argument: `dplyr::across(tidyselect::any_of(variable_names),
##   mangled_skimmers$funs)`.
## i In group 0: .
## Caused by warning:
## ! There were 38 warnings in `dplyr::summarize()`.
## The first warning was:
## i In argument: `dplyr::across(tidyselect::any_of(variable_names),
##   mangled_skimmers$funs)`.
## Caused by warning in `grepl()`:
## ! unable to translate 'T <ef><bf>' to a wide string
## i Run `dplyr::last_dplyr_warnings()` to see the 37 remaining warnings.
```

Table 4: Data summary

Name	df
Number of rows	739
Number of columns	24
Column type frequency:	
character	5
numeric	19
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
track_name	0	1	2	123	0	735	0
artist(s)_name	0	1	1	117	0	517	0
streams	0	1	4	10	0	736	0
key	0	1	1	2	0	11	0
mode	0	1	5	5	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
artist_count	0	1	1.59	0.89	1	1.0	1	2.0	8
released_year	0	1	2021.26	2.16	2013	2021.0	2022	2022.0	2023
released_month	0	1	6.17	3.46	1	3.0	6	9.0	12

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
released_day	0	1	14.22	9.09	1	6.0	13	22.0	31
in_spotify_playlists	0	1	3664.52	5748.11	31	797.0	1801	3969.5	52898
in_spotify_charts	0	1	11.67	18.74	0	0.0	3	16.0	147
in_apple_playlists	0	1	55.84	73.23	0	11.0	28	69.0	532
in_apple_charts	0	1	49.36	49.84	0	6.5	33	84.0	275
in_deezer_playlists	0	1	193.57	655.39	0	12.0	32	102.0	8215
in_deezer_charts	0	1	2.59	5.62	0	0.0	0	2.0	45
in_shazam_charts	0	1	54.02	144.20	0	0.0	3	37.0	1451
bpm	0	1	122.34	27.91	65	99.0	120	140.0	204
danceability_%	0	1	68.13	14.16	25	59.0	70	79.0	96
valence_%	0	1	50.94	23.36	4	32.0	51	69.0	97
energy_%	0	1	64.31	15.73	14	53.0	65	76.0	97
acousticness_%	0	1	26.10	24.89	0	6.0	17	41.0	97
instrumentalness_%	0	1	1.70	8.94	0	0.0	0	0.0	91
liveness_%	0	1	18.27	13.60	3	10.0	12	24.0	97
speechiness_%	0	1	10.87	10.44	2	4.0	6	13.0	64

```
# Remove invalid rows
invalid_rows <- which(!stringi::stri_enc_isutf8(df$track_name))
df <- df[!is.na(iconv(df$track_name, from = "UTF-8", to = "ASCII",
    sub = NA)), ]
df <- df[df$streams != "BPM110KeyAModeMajorDanceability53Valence75Energy69Acousticness7Instrumentalness
    ]

# Update Data Types of the Column
df$streams <- as.numeric(df$streams)
df$released_month <- as.character(df$released_month)
df$released_year <- as.character(df$released_year)
```

Remove Invalid Value and Convert Data Types

Data Exploration & Analysis

Performed various analysis to understand the dataset, and gather insights.

```
max_data <- max(df$streams, na.rm = TRUE)
top_song_ov_yr = df[df$streams == max_data, ]
top_song_ov_yr <- top_song_ov_yr %>%
    summarise(track_name, `artist(s)_name`, released_year, total_stream = streams/1e+09)
colnames(top_song_ov_yr) <- c("Track Name", "Artist Name", "Release Year",
    "Total Streams (In Billions)")
kable(top_song_ov_yr, align = "lccrr")
```

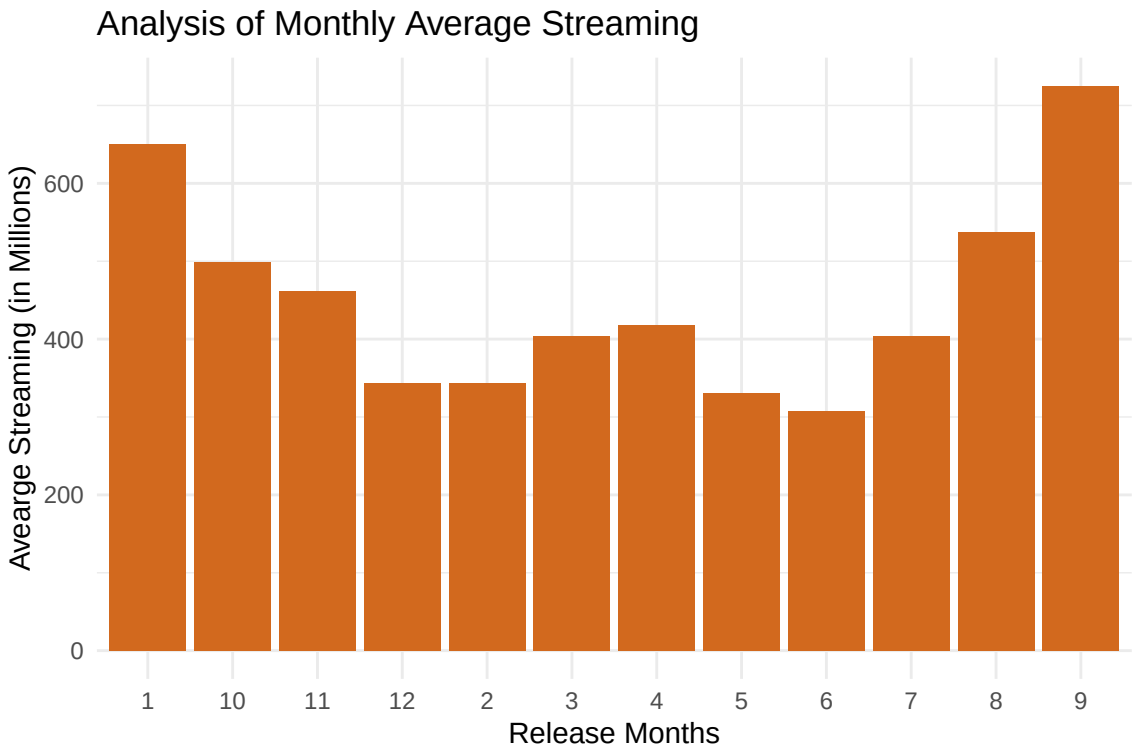
Which song has the highest number of streams over the years?

Track Name	Artist Name	Release Year	Total Streams (In Billions)
Shape of You	Ed Sheeran	2017	3.562544

```r
# Monthly analysis
df_grp_month <- df %>%
    group_by(released_month) %>%
    summarise(mean_stream = mean(streams)) %>%
    arrange(as.numeric(released_month))

ggplot(df_grp_month, aes(x = released_month, y = mean_stream/1e+06,
    group = 1)) + geom_col(fill = "chocolate") + labs(x = "Release Months",
    y = "Avearge Streaming (in Millions)", title = "Analysis of Monthly Average Streaming") +
    theme_minimal()
```

What is the impact of music Release Month or Year on Streaming?



Interpretations & Insights The plot shows in the month of January and September the average streaming on music platform are comparatively higher than other months. It could be due to following reasons:

- Songs released in *January* might be benefited from the *New Year season* when users are actively streaming music.
- Similarly, *September* could be associated with *back-to-school energy or pre-holiday* season engagement.
- *February to July* reflects lower engagement might be due to *fewer significant events* during this time. This could also be due to *summer holidays* so people would be at home streaming OTT or travelling.

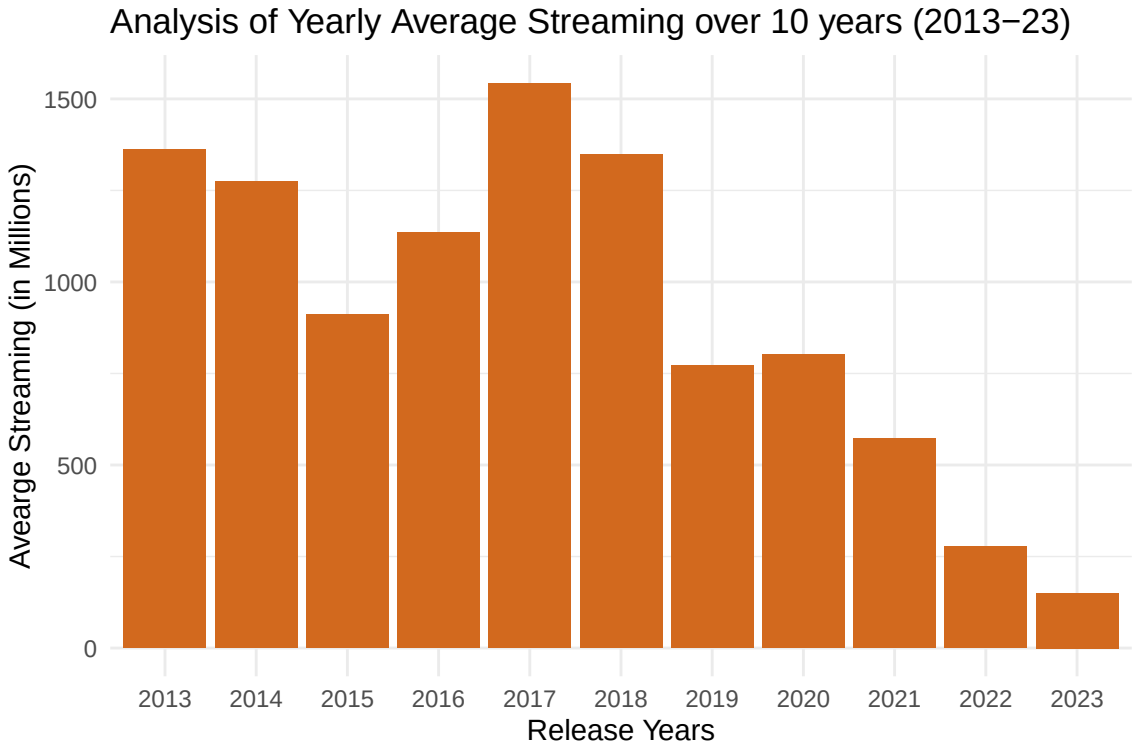
The numbers 1 to 12 depicts the months of the year. The plot shows months in the following order:

```r
# Numeric to Months
months = tibble(Numbers = c(1, 10, 11, 12, 2, 3, 4, 5, 6, 7,
    8, 9), Released_Month = c("Jan", "Oct", "Nov", "Dec", "Feb",
    "March", "April", "May", "June", "July", "Aug", "Sept"))
kable(months, align = "lccrr")
```

Numbers	Released_Month
1	Jan
10	Oct
11	Nov
12	Dec
2	Feb
3	March
4	April
5	May
6	June
7	July
8	Aug
9	Sept

```
# Yearly analysis
df_grp_yr <- df %>%
    group_by(released_year) %>%
    summarise(mean_streams = mean(streams)) %>%
    arrange(as.numeric(released_year)) %>%
    top_n(11, released_year)

ggplot(df_grp_yr, aes(x = released_year, y = mean_streams/1e+06,
    group = 1)) + geom_col(fill = "chocolate") + labs(x = "Release Years",
    y = "Avearge Streaming (in Millions)", title = "Analysis of Yearly Average Streaming over 10 years
    theme_minimal()
```

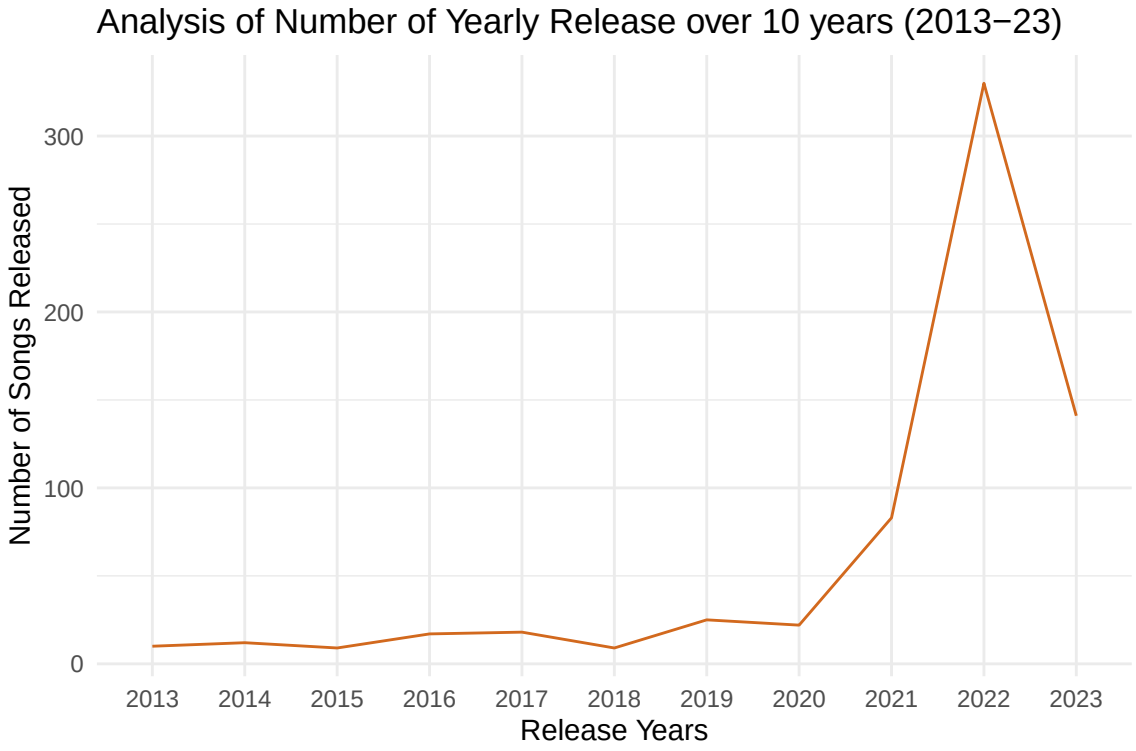


Interpretations & Insights The graph shows highest average streaming in 2017. With decline from 2020-23, indicating lower engagement with streaming platform. There has been slight drop between 2013 and 2014 also, remaining other years the streaming was steady.

- According to report, Spotify experienced massive growth in their user base in 2017 reaching *157 million monthly active users and 71 million premium subscribers.*
- The company launched new features like *Discover Weekly* in July 2015 and *Release Radar* in August 2016 that further contributed to increase in streaming.
- With rise in *short-form video platforms* like TikTok, Instagram Reels, YouTube Shorts etc., lead to change in customer behavior, instead of streaming full songs, *users engaged with music in short clips.*
- *Pandemic and lockdowns* accelerated this shift in user behavior leading to decline in streaming since 2020.

```
# Yearly release count
df_grp_yr <- df %>%
    group_by(released_year) %>%
    summarise(count_tracks = n()) %>%
    arrange(as.numeric(released_year)) %>%
    top_n(11, released_year)

ggplot(df_grp_yr, aes(x = released_year, y = count_tracks, group = 1)) +
    geom_line(color = "chocolate") + labs(x = "Release Years",
    y = "Number of Songs Released", title = "Analysis of Number of Yearly Release over 10 years (2013-23
    theme_minimal()
```



Interpretations & Insights

- Song release *frequency is steady and low between 2013-19*, which means fewer artists dominated streaming platforms.
- Since 2019, there has been rise in count of songs released or available on streaming platform. This rise could be explained by *growing influence of independent artists using Spotify, SoundCloud, and YouTube Music.*
- The rise since 2020, is also supported by pandemic effect. During pandemic many artists *focused on digital releases over live concerts.*
- In 2022, there is sudden spike in release count, that happened because many *artists has delayed releases*

in 2020-21 and launched multiple projects in 2022.

- Boom in independent artist could also explain spike release count in FY 2022. Pandemic paved way for many unsigned artists, releasing music via streaming services.
- Release frequency on streaming platform again *declined in 2023*, that could be due to *artists prioritizing quality over quantity.*
- However, more release does not lead more streaming. According to yearly streaming analysis there has been decline in streaming since 2020, whereas above analysis shows spike in 2022. *This suggest that while releases surged in 2022, streaming growth dropped after 2020.*

```
df_grp_artist <- df %>%
    group_by(`artist(s)_name`) %>%
    summarise(total_streams = as.integer(sum(streams)/10^9)) %>%
    arrange(desc(total_streams)) %>%
    top_n(10, total_streams)
colnames(df_grp_artist) <- c("Artist Name", "Total Streams (In Billions)")
kable(df_grp_artist, align = "lccrr")
```

Who are the top 10 Artist with large customer base over the years?

Artist Name	Total Streams (In Billions)
Ed Sheeran	11
Taylor Swift	9
Bad Bunny	6
Harry Styles	6
The Weeknd	5
SZA	4
Arctic Monkeys	3
Avicii	3
Dua Lipa	3
Imagine Dragons	3
Kendrick Lamar	3
Olivia Rodrigo	3

```
df_grp_artist_yr <- df %>%
    group_by(released_year) %>%
    filter(streams == max(streams)) %>%
    select(released_year, `artist(s)_name`) %>%
    arrange(desc(released_year))
colnames(df_grp_artist_yr) <- c("Year", "Artist Names")
kable(df_grp_artist_yr, align = "lccrr")
```

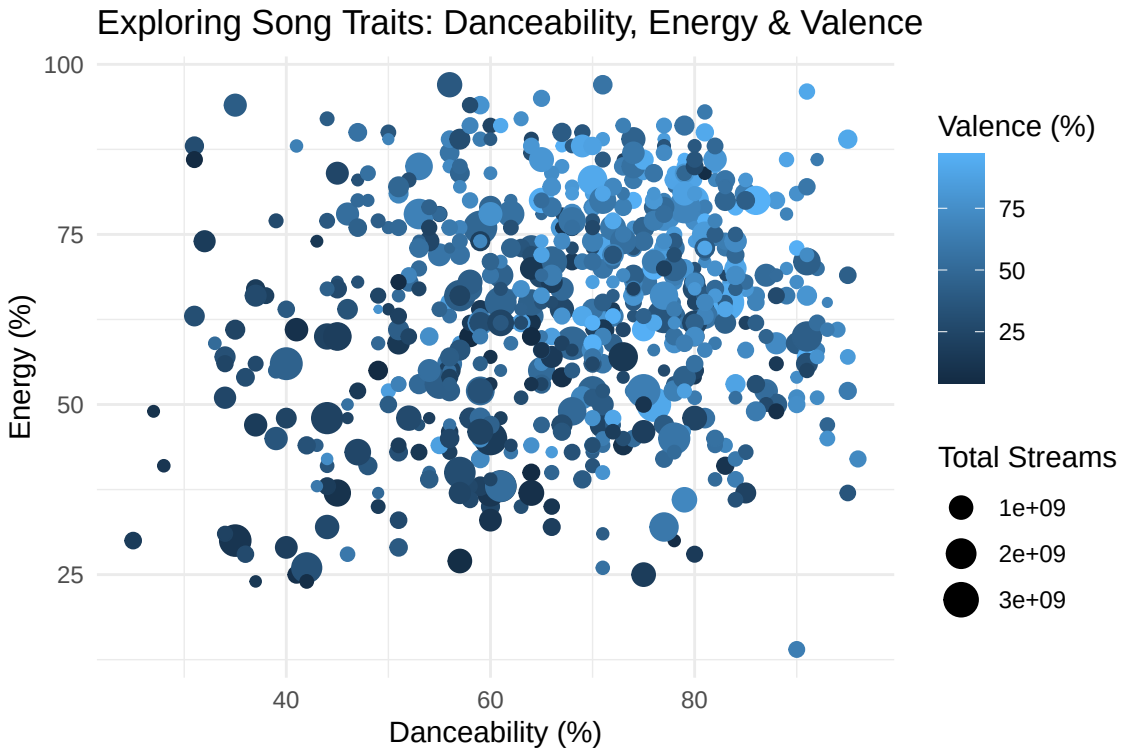
Top artist of each years

Year	Artist Names
2023	Eslabon Armado, Peso Pluma
2022	Chencho Corleone, Bad Bunny
2021	Justin Bieber, The Kid Laroi
2020	Dua Lipa, DaBaby
2019	Dua Lipa
2018	Post Malone, Swae Lee
2017	Ed Sheeran

Year	Artist Names
2016	Drake, WizKid, Kyla
2015	Justin Bieber
2014	Ed Sheeran
2013	Hozier

```
ggplot(df, aes(x = `danceability_%`, y = `energy_%`, color = `valence_%`,
    size = streams)) + geom_point() + labs(x = "Danceability (%)",
    y = "Energy (%)", color = "Valence (%)", size = "Total Streams",
    title = "Exploring Song Traits: Danceability, Energy & Valence") +
    theme_minimal()
```

How does songs streaming varies in terms of dance, valence and energy?



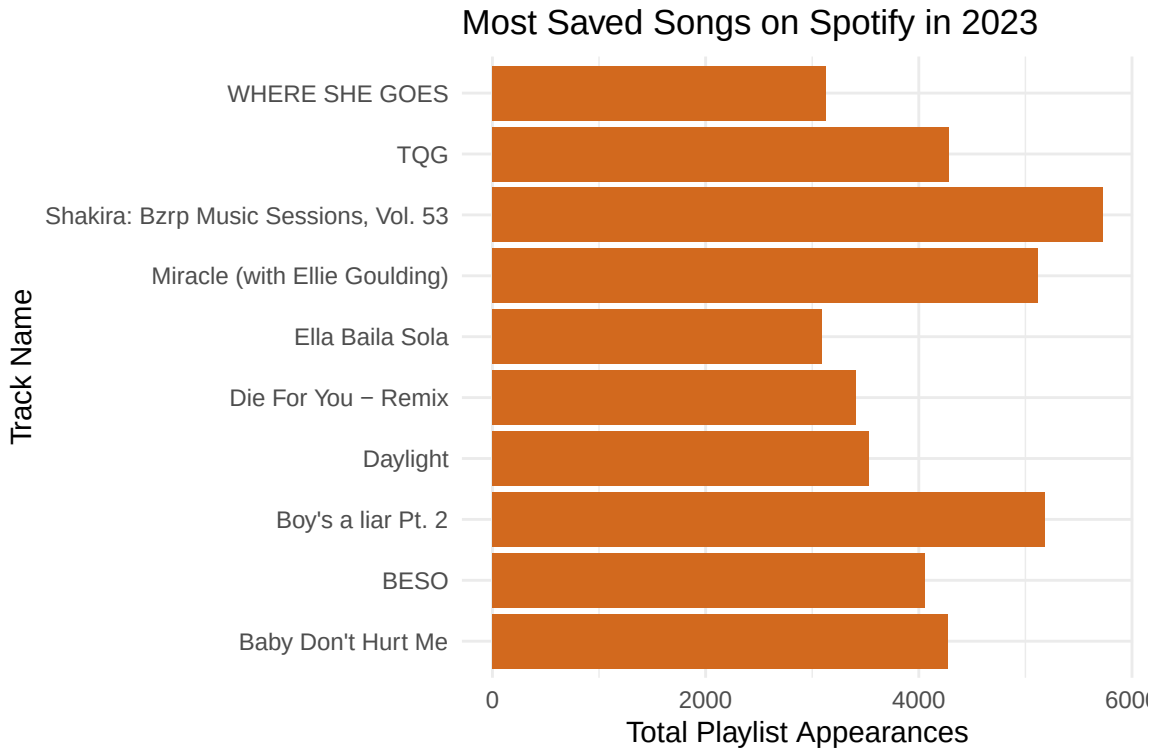
Interpretations & Insights

- The points are widely dispersed, so there is *no clear linear trend between danceability and energy* of music, means it is not always correlated. This indicates that *highly energetic song might not always be danceable.*
- Size of points tells the streaming intensity. Since it is also distributed and large, this indicates that *hit songs exist across different levels* of energy and danceability.
- Color of the points tells variance, darker it is *sad or neutral cells more popular* whereas, *lighter indicates trends of happy songs.*
- Color of these points becomes lighter, as danceablitity increases. Which means danceable songs makes people feel happier and tune with.

```
df_track_2023 <- df %>%
    filter(released_year == "2023") %>%
```

```
    select(track_name, in_spotify_playlists) %>%
    arrange(desc(in_spotify_playlists)) %>%
    top_n(10, in_spotify_playlists)
# graph
ggplot(df_track_2023, aes(x = in_spotify_playlists, y = track_name)) +
    geom_col(fill = "chocolate") + labs(x = "Total Playlist Appearances",
    y = "Track Name", title = "Most Saved Songs on Spotify in 2023") +
    theme_minimal()
```

Top 10 Songs added to Playlist on Spotify in 2023




```
df_track_yr <- df %>%
    group_by(released_year) %>%
    filter(in_spotify_playlists == max(in_spotify_playlists)) %>%
    select(released_year, track_name, in_spotify_playlists) %>%
    arrange(desc(released_year))
colnames(df_track_yr) <- c("Released Year", "Track Name", "No. of Playlist Appearance")
kable(df_track_yr, align = "lccrr")
```

Top Most Saved Song each Year on Spotify (2013-23)

Released Year	Track Name	No. of Playlist Appearance
2023	Shakira: Bzrp Music Sessions, Vol. 53	5724
2022	I'm Good (Blue)	12482
2021	STAY (with Justin Bieber)	17050
2020	Levitating (feat. DaBaby)	15894
2019	Don't Start Now	27119
2018	Sunflower - Spider-Man: Into the Spider-Verse	24094

Released Year	Track Name	No. of Playlist Appearance
2017	HUMBLE.	33206
2016	One Dance	43257
2015	Love Yourself	22730
2014	Thinking Out Loud	33032
2013	Get Lucky - Radio Edit	52898

Interpretations & Insights All these songs are Pop/Hip-Hop, which indicates that people prefer danceable songs to cope up with their daily work life.

```
df_chart <- df %>%
    filter(released_year == "2023" & in_spotify_charts == 1)
df_chart$track_name
```

Tracks that backed #1 on Spotify in 2023

```
## [1] "Hummingbird (Metro Boomin & James Blake)"
## [2] "Mejor Que Yo"
```

```
df_chart_yr <- df %>%
    group_by(released_year) %>%
    filter(in_spotify_charts == 1) %>%
    select(released_year, track_name) %>%
    arrange(desc(released_year))
colnames(df_chart_yr) <- c("Released Year", "Track Name")
kable(df_chart_yr, align = "lccrr")
```

Tracks that backed #1 on Spotify over the years (2013-23)

Released Year	Track Name
2023	Hummingbird (Metro Boomin & James Blake)
2023	Mejor Que Yo
2022	Un Verano Sin Ti
2022	Music For a Sushi Restaurant
2022	Keep Driving
2022	Daydreaming
2021	ELEVEN
2019	Adore You
2019	Golden
2019	San Lucas
2017	HUMBLE.

```
df_chart_2023 <- df %>%
    filter(released_year == "2023" & in_spotify_charts <= 10 &
        in_spotify_charts != 0) %>%
    select(track_name, in_spotify_charts) %>%
    arrange(in_spotify_charts)
colnames(df_chart_2023) <- c("Track Name", "Chart Ranks")
kable(df_chart_2023, align = "lccrr")
```

Tracks that backed top 10 position on Spotify charts in 2023

Track Name	Chart Ranks
Hummingbird (Metro Boomin & James Blake)	1
Mejor Que Yo	1
Snow On The Beach (feat. More Lana Del Rey)	2
Oi Balde - Ao Vivo	2
on the street (with J. Cole)	2
Heaven	2
Never Felt So Alone	3
Annihilate (Spider-Man: Across the Spider-Verse) (Metro Boomin & Swae Lee, Lil Wayne, Offset)	4
Princess Diana (with Nicki Minaj)	4
Mami Chula	4
Rosa Pastel	4
People Pt.2 (feat. IU)	4
If We Ever Broke Up	4
Sial	4
Self Love (Spider-Man: Across the Spider-Verse) (Metro Boomin & Coi Leray)	5
Abcdario	5
Stand By Me (feat. Morgan Wallen)	5
Seu Brilho Sumiu - Ao Vivo	5
Erro Gostoso - Ao Vivo	5
Phir Aur Kya Chahiye (From “Zara Hatke Zara Bachke”)	6
Lilith (feat. SUGA of BTS) (Diablo IV Anthem)	6
Search & Rescue	6
Dijeron Que No La Iba Lograr	6
Tere Vaaste (From “Zara Hatke Zara Bachke”)	8
Gol Bolinha, Gol Quadrado 2	8
Mi Bello Angel	8
Cheques	8
Las Morras	8
Nosso Quadro	9
Shoong! (feat. LISA of BLACKPINK)	9
REMIX EXCLUSIVO	9
Chanel	10
Cupid	10
Di Que Si	10
Watch This - ARIZONATEARS Pluggnb Remix	10

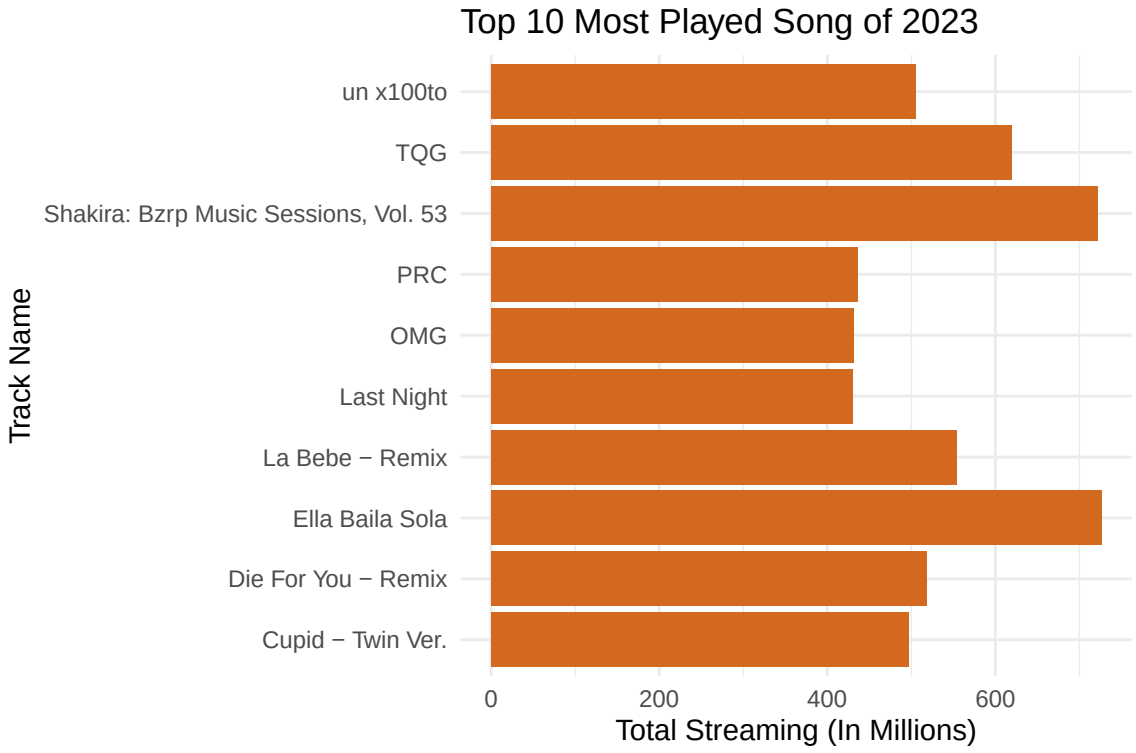
```
df_top_song <- df %>%
    group_by(released_year) %>%
    filter(streams == max(streams)) %>%
    reframe(released_year, track_name, streams = as.integer(streams/10^6)) %>%
    arrange(desc(released_year))
colnames(df_top_song) <- c("Release Year", "Track Name", "Streamings (In Millions)")
kable(df_top_song, align = "lccrr")
```

Most streamed song of each year (2013-23)

Release Year	Track Name	Streamings (In Millions)
2023	Ella Baila Sola	725
2022	Me Porto Bonito	1440
2021	STAY (with Justin Bieber)	2665
2020	Levitating (feat. DaBaby)	1802
2019	Don't Start Now	2303
2018	Sunflower - Spider-Man: Into the Spider-Verse	2808
2017	Shape of You	3562
2016	One Dance	2713
2015	Love Yourself	2123
2014	Thinking Out Loud	2280
2013	Take Me To Church	2135

```
df_top_song_2023 <- df %>%
    filter(released_year == "2023") %>%
    reframe(track_name, streams = streams/10^6) %>%
    arrange(desc(streams)) %>%
    top_n(10, streams)
# plot
ggplot(df_top_song_2023, aes(x = streams, y = track_name)) +
    geom_col(fill = "chocolate") + labs(x = "Total Streaming (In Millions)",
    y = "Track Name", title = "Top 10 Most Played Song of 2023") +
    theme_minimal()
```

Top 10 most streamed song of in 2023



Statistical Analysis

```r
df_numeric <- df[, c("artist_count", "streams", "bpm", "danceability_%",
    "valence_%", "energy_%", "acousticness_%", "instrumentalness_%",
    "liveness_%", "speechiness_%", "in_spotify_playlists")]
corr <- cor(df_numeric)
corr
```

Correlation Analysis

```
##                      artist_count     streams          bpm danceability_%
## artist_count           1.00000000 -0.07807971 -0.082385178     0.21702904
## streams               -0.07807971  1.00000000 -0.049369034    -0.06651020
## bpm                   -0.08238518 -0.04936903  1.000000000    -0.11294992
## danceability_%         0.21702904 -0.06651020 -0.112949916     1.00000000
## valence_%              0.12252251 -0.03591224  0.039279582     0.40627590
## energy_%               0.14763204 -0.08724604  0.012151217     0.15158016
## acousticness_%        -0.12671885  0.06539444 -0.043577477    -0.23273482
## instrumentalness_%    -0.06189578 -0.02926153  0.006028457    -0.10278569
## liveness_%             0.01644715 -0.05092836 -0.006383745    -0.07606709
## speechiness_%          0.10462494 -0.10322890  0.055033329     0.18117733
## in_spotify_playlists  -0.03438183  0.82434001 -0.050045321    -0.04410162
##                        valence_%    energy_% acousticness_% instrumentalness_%
## artist_count          0.122522514  0.14763204   -0.126718853       -0.061895777
## streams              -0.035912237 -0.08724604    0.065394436       -0.029261533
## bpm                   0.039279582  0.01215122   -0.043577477        0.006028457
## danceability_%        0.406275901  0.15158016   -0.232734821       -0.102785690
## valence_%             1.000000000  0.36936648   -0.079987519       -0.152326794
## energy_%              0.369366481  1.00000000   -0.549176393       -0.047391985
## acousticness_%       -0.079987519 -0.54917639    1.000000000        0.047921162
## instrumentalness_%   -0.152326794 -0.04739199    0.047921162        1.000000000
## liveness_%            0.003403899  0.12306792   -0.052030288       -0.046953706
## speechiness_%         0.057276724 -0.05501038   -0.007261277       -0.094412402
## in_spotify_playlists  0.007426094 -0.02588492   -0.009931856       -0.018244986
##                       liveness_% speechiness_% in_spotify_playlists
## artist_count          0.016447146   0.104624937         -0.034381832
## streams              -0.050928365  -0.103228901          0.824340012
## bpm                  -0.006383745   0.055033329         -0.050045321
## danceability_%       -0.076067088   0.181177333         -0.044101618
## valence_%             0.003403899   0.057276724          0.007426094
## energy_%              0.123067919  -0.055010381         -0.025884918
## acousticness_%       -0.052030288  -0.007261277         -0.009931856
## instrumentalness_%   -0.046953706  -0.094412402         -0.018244986
## liveness_%            1.000000000  -0.024481593         -0.038669865
## speechiness_%        -0.024481593   1.000000000         -0.081607299
## in_spotify_playlists -0.038669865  -0.081607299          1.000000000
```

Interpretations & Insights

- ***Streams vs. Playlist Presence***: Strong positive correlation between streams and songs being in Spotify playlists (0.82).
- ***Danceability vs. Valence & Energy***: Songs that are more danceable tend to be more energetic and have a happier. mood (valence). Danceability is positively correlated with valence (0.41) and energy (0.15).
- ***Streams vs. Danceability, Energy, and Valence***: Viral songs or widely streamed tracks are not necessarily the most danceable, energetic, or happy, other factors like artist popularity, marketing,

and playlist inclusion play a bigger role. Weak negative correlation between streams and danceability (-0.067),energy (-0.087),and valence (-0.036).

- ***Speechiness vs. Energy & Valence***: Speechiness is positively correlated with energy (0.18) but weakly correlated with valence (0.057). Which means that songs with more speech-like qualities (e.g., rap) often have higher energy but are not necessarily associated with happy/positive emotions.
- ***Energy vs. Acousticness***: High-energy songs tend to have less acoustic elements, as they rely more on electronic production and beats. Therefore, strong negative correlation exists between energy and acoustics (-0.54).
- ***Instrumentalness vs. Danceability & Valence***: There is strong negative correlation between instrumentalness and danceability (-0.10) and valence (-0.15). Which means that songs with lyrics (lower instrumentalness) are often more engaging, danceable, and emotionally expressive.

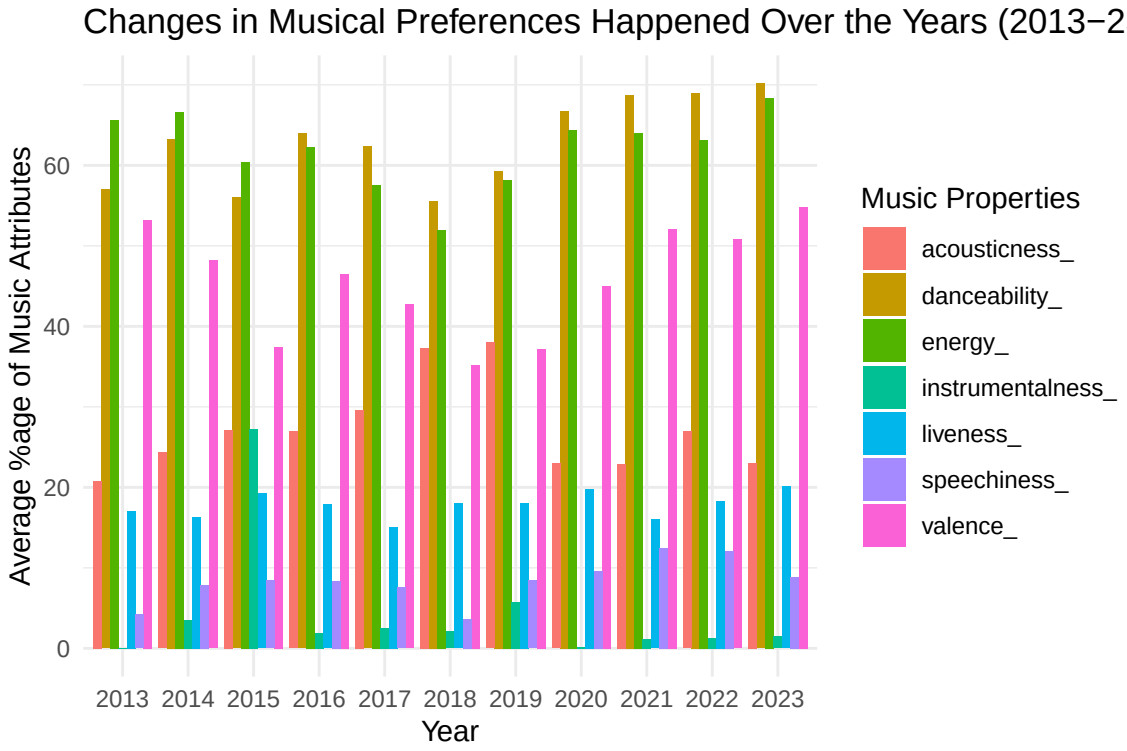
```
df_pref_yr <- df %>%
    group_by(released_year) %>%
    summarise(danceability_ = mean(`danceability_%`), valence_ = mean(`valence_%`),
        energy_ = mean(`energy_%`), acousticness_ = mean(`acousticness_%`),
        instrumentalness_ = mean(`instrumentalness_%`), liveness_ = mean(`liveness_%`),
        speechiness_ = mean(`speechiness_%`)) %>%
    arrange(desc(released_year)) %>%
    top_n(11, released_year)

df_pref_tr_yr <- df_pref_yr %>%
    pivot_longer(cols = ends_with("_"), names_to = "Song_Preferences",
        values_to = "avg_%age")
df_pref_tr_yr
```

What global changes in musical preferences happened over the years

```
## # A tibble: 77 x 3
##    released_year Song_Preferences  `avg_%age`
##    <chr>         <chr>                  <dbl>
##  1 2023          danceability_          70.1
##  2 2023          valence_               54.9
##  3 2023          energy_                68.4
##  4 2023          acousticness_          23.0
##  5 2023          instrumentalness_       1.49
##  6 2023          liveness_              20.1
##  7 2023          speechiness_            8.86
##  8 2022          danceability_          69.0
##  9 2022          valence_               50.8
## 10 2022          energy_                63.2
## # i 67 more rows
```

```
# Graph based on table above. Change in preference
ggplot(df_pref_tr_yr, aes(x = released_year, y = `avg_%age`,
    fill = Song_Preferences)) + geom_bar(stat = "identity", position = "dodge") +
    labs(x = "Year", y = "Average %age of Music Attributes",
        fill = "Music Properties", title = "Changes in Musical Preferences Happened Over the Years (201
    theme_minimal()
```



Interpretations & Insights

- There is steady *rise in danceability of music* being created overtime. With the rise in streaming platforms and social media trends (e.g., TikTok dance challenges) customer preferences for highly danceable music has increased.
- *Energy of music has fairly steady rise*, which means Pop, hip-hop, and EDM have maintained their dominance, keeping energy levels relatively stable.
- In recent years there has been *rise in valence attribute of the music*. Might be an effect of pandemic, when users preferences shifted towards more uplifting and positive music.
- *Decline of acousticness & instrumentalness* over the years, shows preference for electronically produced music over acoustic or instrumental tracks. Due to rise in digital production techniques and AI-generated music, the attributes like acousticness and instrumentalness has been decreased from music these days.
- *Speechiness & Liveness is very low*, which indicates speech-heavy tracks (such as rap) remain popular but do not dominate, and liveness is not a major factor in mainstream music trends.

```
# H0:There is no significant difference in the average
# number of streams between songs in the Major key and the
# Minor key.  Ha: There is a significant difference in the
# average number of streams between songs in the Major and
# Minor keys.
df %>%
    select(streams, mode) %>%
    filter(mode %in% c("Major", "Minor")) %>%
    drop_na(streams) %>%
    t.test(streams ~ mode, data = ., alternative = "two.sided",
        mu = 0, var.equal = FALSE, conf.level = 0.95)
```

Hypothesis Testing

```
##
##  Welch Two Sample t-test
##
## data:  streams by mode
## t = 1.6052, df = 666.55, p-value = 0.1089
## alternative hypothesis: true difference in means between group Major and group Minor is not equal to
## 95 percent confidence interval:
##  -13919348 138638868
## sample estimates:
## mean in group Major mean in group Minor
##           462756925           400397165
```

```
# Null hypothesis (HO) is not rejected as p-value is
# greater than 0.05
```

Interpretations & Insights

- The p-value is greater than 0.05, meaning there is not enough evidence to reject the null hypothesis at a 95% confidence level.
- Since p-value results are not statistically significant, therefore null hypothesis is not rejected. There is no significant difference in the average number of streams between songs in the Major key and the Minor key.
- Mean for major key songs (462756925) is slightly higher, therefore there difference is not statistically significant.
- Null hypothesis is not rejected. Though, there is no strong evidence that songs in major keys perform significantly better than those in minor keys in terms of streams.

Conclusion

This project analyzed the Spotify dataset to understand trends in musical preferences and factors influencing song popularity. We explored various song properties such as danceability, energy, valence, and key mode to assess their impact on streaming performance.A Two-Sample t-test showed no significant difference in average streams between songs in major and minor keys, indicating that key mode alone does not determine a song's success.Furthermore, the analysis of musical trends from 2013 to 2023 highlighted a shift in listener preferences, with increasing emphasis on danceability and energy.While musical characteristics contribute to a song's appeal, other factors like artist influence, marketing strategies, and playlist placements likely play a crucial role in driving streams. This study provides valuable insights into the evolving landscape of music consumption.