

FINAL PROJECT “SPOTIFY DATA”

GSI Intro to Big Data and Data Mining

The University of Texas at Austin

*Zhaowen Fan
Rafael Ignacio Gonzalez Chong*

Table of Contents

<i>Project Definition</i>	3
<i>Project Implementation</i>	4
- <i>Cleaning the data</i>	4
- <i>Finding the most popular artist and album of each month</i>	5
- <i>Finding the underlying listening habits</i>	6
- <i>The output from the console</i>	7
- <i>The visualizations</i>	8
<i>Appendices (Code)</i>	9

Project Definition:

About:

This dataset contains a detailed Spotify listening history of a user. Each record captures information about a specific track played on Spotify, including metadata about the track, playback behavior, and the user's interaction with the platform. It spans multiple years and provides insights into user behavior, track preferences, and session dynamics.

Objective:

Investigate how user listening habits (e.g., song completion rates, shuffle usage, skip behavior) and song popularity vary depending on the shuffle setting (on or off) and identify factors that influence whether a song is skipped or played in full.

We aim to learn which songs or artists are associated with higher engagement (e.g., longer playtime or fewer skips) and whether contextual factors like shuffle mode or autoplay impact user interaction.

This project combines Data Wrangling/Data Manipulation and Exploratory Data Analysis (EDA) as its primary learning models. We will aggregate data to compute metrics like skip rates and song completion percentages, visualize trends, and explore relationships between variables.

Expectations:

We expect that user behavior will vary significantly depending on the settings. For instance, we anticipate that songs started via autoplay are more likely to be skipped than manually selected songs, as the autoplay is playing songs without surveillance, but under expectation. Additionally, shuffle mode might correlate with higher skip rates due to less predictable song sequences, so that the song might not be liked by the listener. Our expectations are based on our behavior in music listening. However, the data may reveal unexpected patterns, such as specific artists or time periods driving higher completion rates.

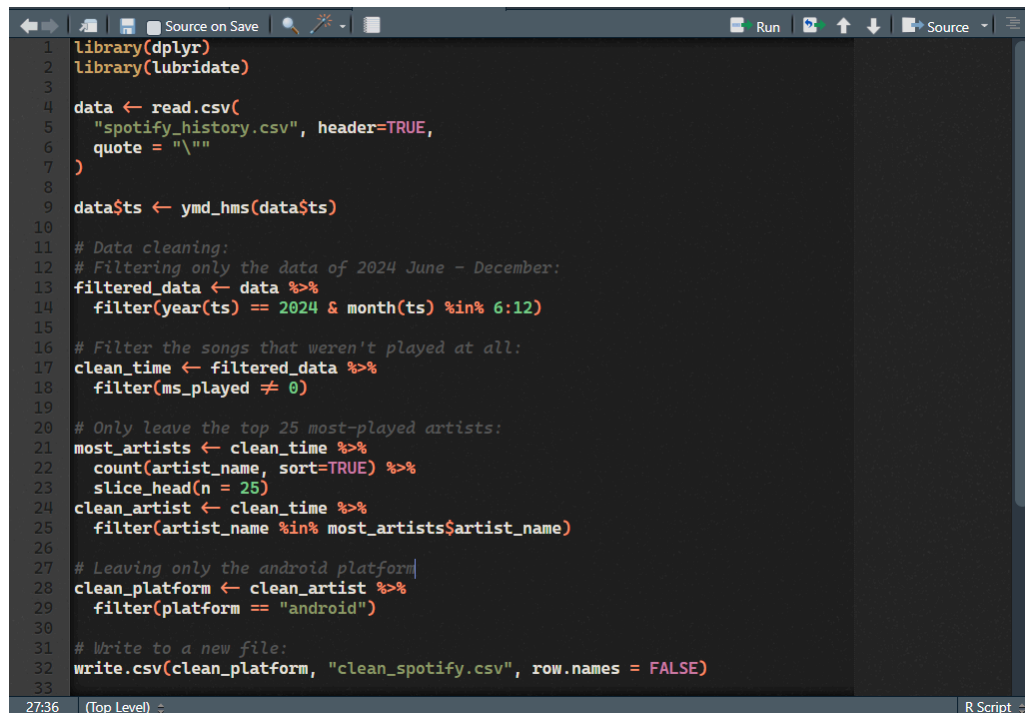
Evaluation and Assessment:

The project's will be evaluated based on the clarity and depth of analysis derived from the data. For the EDA component, we will assess correctness by ensuring data manipulations (e.g., skip rate calculations) are accurate and visualizations (e.g., histograms of listen habits, heatmaps of skip rates by platform) align with computed metrics. For hypothesis tests, we will use statistical significance (e.g., p-value < 0.05 for a t-test) to validate findings, such as differences in skip rates across conditions. We expect the EDA to perform robustly, as it relies on straightforward aggregations and visualizations. We hope our project can answer the research question with clear, data-driven insights.

Project Implementation:

1. Cleaning the data:

As we are supposed to use a data set with less than 2000 rows, we have cut the original data with the following code:



```
1 library(dplyr)
2 library(lubridate)
3
4 data <- read.csv(
5   "spotify_history.csv", header=TRUE,
6   quote = "\""
7 )
8
9 data$ts <- ymd_hms(data$ts)
10
11 # Data cleaning:
12 # Filtering only the data of 2024 June - December:
13 filtered_data <- data %>%
14   filter(year(ts) == 2024 & month(ts) %in% 6:12)
15
16 # Filter the songs that weren't played at all:
17 clean_time <- filtered_data %>%
18   filter(ms_played != 0)
19
20 # Only leave the top 25 most-played artists:
21 most_artists <- clean_time %>%
22   count(artist_name, sort=TRUE) %>%
23   slice_head(n = 25)
24 clean_artist <- clean_time %>%
25   filter(artist_name %in% most_artists$artist_name)
26
27 # Leaving only the android platform
28 clean_platform <- clean_artist %>%
29   filter(platform == "android")
30
31 # Write to a new file:
32 write.csv(clean_platform, "clean_spotify.csv", row.names = FALSE)
33
```

2. Finding the most popular artist and album of each month:

- Top Artists:

```
# A tibble: 7 × 3

# Groups:   month [7]

  month artist_name      total_time_ms
  <fct> <chr>          <int>
1 Jun   Queen           2155221
2 Jul   The Beatles      2302834
3 Aug   The Killers        14726642
4 Sep   The Beatles         8385680
5 Oct   The Beatles        12843693
6 Nov   Kendrick Lamar     12669366
7 Dec   John Mayer          497368
```

- Top Albums:

```
# A tibble: 7 × 3

# Groups:   month [7]

  month album_name      total_time_ms
  <fct> <chr>          <int>
1 Jun   Wish You Were Here  811077
2 Jul   Past Masters        759534
3 Aug   Arrival              3727597
4 Sep   Plastic Ono Band    2069091
5 Oct   To Pimp A Butterfly  8078446
6 Nov   GNX                  9698803
7 Dec   Night Visions        412892
```

3. Finding the underlying listening habits:

We analysed the pattern based on the relationships between reason start and skipped, reason end and skipped, and shuffle and skipped. To calculate the skip rate, the dataset was grouped by the combinations of reason_start and reason_end, and the mean of the skipped variable within each group was computed to represent the proportion of skipped tracks for each reason pair.

According to the output from the console, we have found several intriguing underlying patterns:

reason_start vs. skipped:

- Certain start reasons (like "trackdone" or "forwardbtn") tend to have lower skip rates, implying intentional listening.
- Others, like "apload" or "clickrow", often show higher skip rates, suggesting passive starts or exploratory clicks.

reason_end vs. skipped:

- Reasons like "endplay" or "trackdone" are typically not associated with skipping — they imply full consumption.
- "backbtn", "logout", or "fwdbtn" often correlate with higher skip rates, especially when the track is interrupted.

shuffle vs. skipped:

- On average, users tend to skip more often when shuffle is enabled.
- This could reflect that shuffled tracks don't always match the listener's momentary preference.

4. The output from the console:

```
> table(data$reason_start, data$skipped)
```

	FALSE	TRUE
appload	64	15
backbtn	5	9
clickrow	52	46
fwdbtn	125	385
playbtn	3	4
remote	16	0
trackdone	1104	149
trackerror	4	0
unknown	0	1

```
> table(data$reason_end, data$skipped)
```

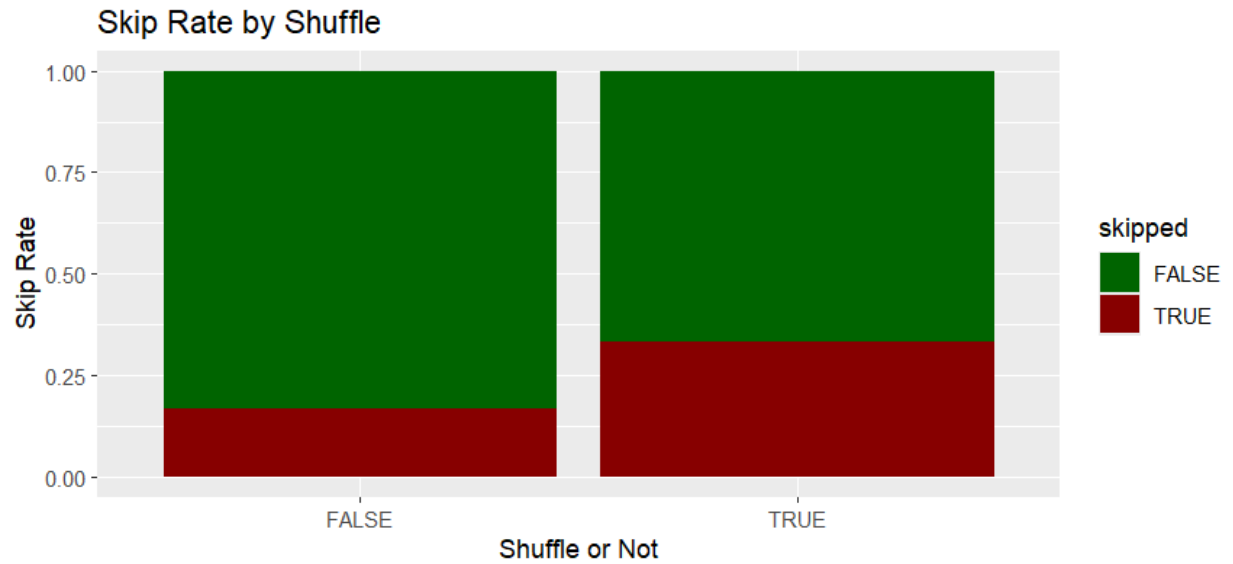
	FALSE	TRUE
backbtn	0	18
endplay	0	95
fwdbtn	0	496
logout	107	0
trackdone	1254	0
unexpected-exit	1	0
unexpected-exit-while-paused	1	0
unknown	10	0

```
> table(data$shuffle, data$skipped)
```

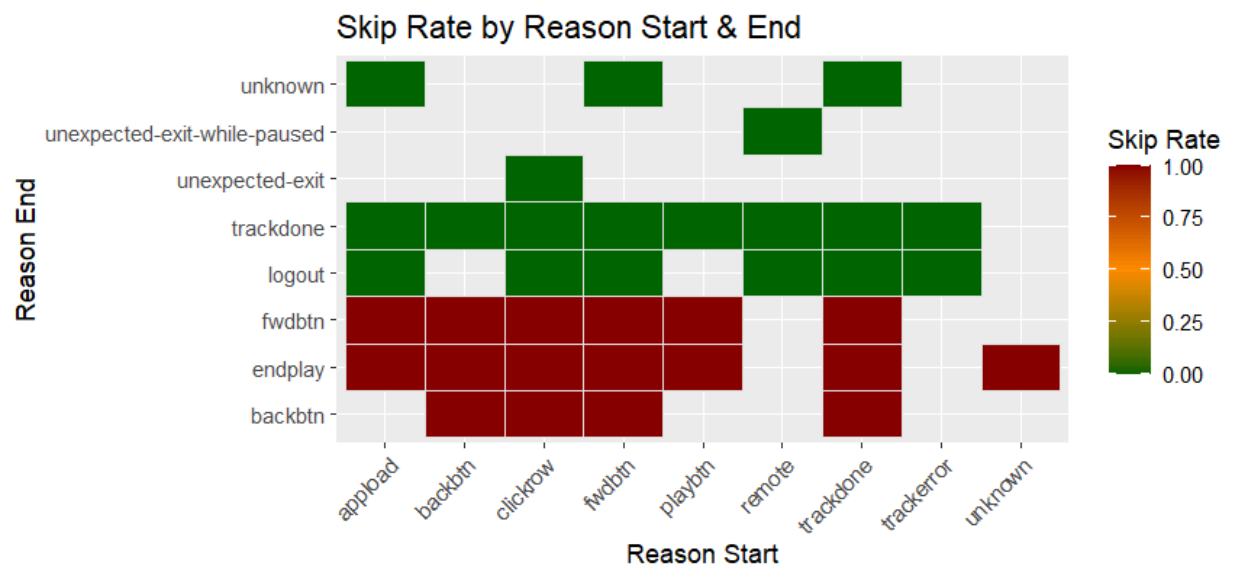
	FALSE	TRUE
FALSE	242	48
TRUE	1131	561

5. The Visualizations:

Bar Chart of Skip Rates by Shuffle Mode:



Heatmap of Skip Rates by Reason Start & End:



Appendices (Code):

```
library(dplyr)
library(ggplot2)
library(lubridate)
Sys.setlocale("LC_TIME", "English_United States.1252")
data <- read.csv("clean_spotify.csv", header=TRUE)
data$ts <- ymd_hms(data$ts)

top_artists <- data %>%
  mutate(month=format(ts, "%b")) %>%
  mutate(month=factor(
    month,
    levels=c("Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
  )) %>%
  group_by(month, artist_name) %>%
  summarise(total_time_ms=sum(ms_played), .groups="drop") %>%
  arrange(month, desc(total_time_ms)) %>%
  group_by(month) %>%
  slice_max(order_by=total_time_ms, n = 1)

print(top_artists)

data %>%
  mutate(month=format(ts, "%b")) %>%
  mutate(month=factor(
    month,
    levels=c("Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
  )) %>%
  group_by(month, album_name) %>%
  summarise(total_time_ms=sum(ms_played), .groups="drop") %>%
  arrange(month, desc(total_time_ms)) %>%
  group_by(month) %>%
```

```

slice_max(order_by=total_time_ms, n = 1)
str(data[, c("reason_start", "reason_end", "shuffle", "skipped")])

data$reason_start <- as.factor(data$reason_start)
data$reason_end <- as.factor(data$reason_end)
data$shuffle <- as.logical(data$shuffle)
data$skipped <- as.logical(data$skipped)

table(data$reason_start, data$skipped)
table(data$reason_end, data$skipped)
table(data$shuffle, data$skipped)

ggplot(data, aes(x = shuffle, fill = skipped)) +
  geom_bar(position = "fill") +
  labs(title = "Skip Rate by Shuffle", y = "Skip Rate", x = "Shuffle or Not") +
  scale_fill_manual(values = c("darkgreen", "darkred"))

reason_skip <- data %>%
  group_by(reason_start, reason_end) %>%
  summarise(skip_rate = mean(skipped))

ggplot(reason_skip, aes(x = reason_start, y = reason_end, fill = skip_rate)) +
  geom_tile(color = "grey") +
  scale_fill_gradient2(
    low = "darkgreen", mid = "darkorange", high = "darkred", midpoint = 0.5
  ) +
  labs(
    title = "Skip Rate by Reason Start & End",
    x = "Reason Start",
    y = "Reason End",
    fill = "Skip Rate"
  )

```

The source codes are uploaded together with the file.

Final Project

GSI Intro to Big Data and Data Mining

by

Zhaowen Fan & Rafael Ignacio Gonzalez Chong

Data Source:

[Top Spotify Listening History Songs in Countries](#)