Spotify Data Analysis

Khoi Trinh

2023-02-21

Analyzing My Spotify Streaming History

All of the codes are based on this article

First, here are the required library

```
# REQUIRED LIBRARIES
library(jsonlite)
library(lubridate)
library(gghighlight)
library(spotifyr)
library(tidyverse)
library(knitr)
library(ggplot2)
library(plotly)
```

Let's read in our data, you can find how to get your own spotify data here

```
streamHistory0 <- data.frame(fromJSON("StreamingHistory0.json", flatten = TRUE))
streamHistory1 <- data.frame(fromJSON("StreamingHistory1.json", flatten = TRUE))
streamHistory2 <- data.frame(fromJSON("StreamingHistory2.json", flatten = TRUE))
streamHistory3 <- data.frame(fromJSON("StreamingHistory3.json", flatten = TRUE))
streamHistory4 <- data.frame(fromJSON("StreamingHistory4.json", flatten = TRUE))
streamHistory5 <- data.frame(fromJSON("StreamingHistory5.json", flatten = TRUE))

streamHistory <- Map(c, streamHistory0, streamHistory1, streamHistory2,
                     streamHistory3, streamHistory4, streamHistory5)
```

On what day did I listen to more or less music?

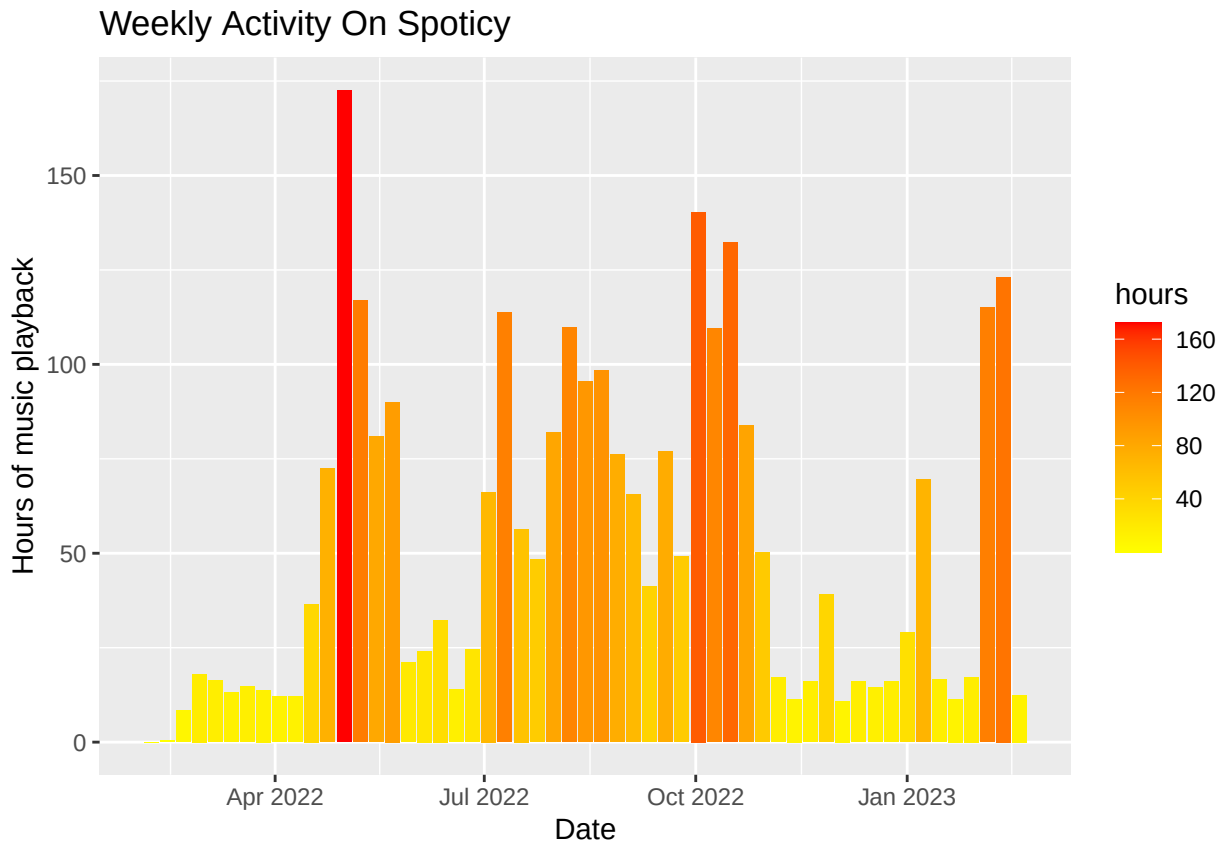
```
# Add date and time
mySpotify <- streamHistory %>%
  as_tibble() %>%
  mutate_at("endTime", ymd_hm) %>%
  mutate(endTime = endTime - hours(6)) %>%
  mutate(date = floor_date(endTime, "day") %>% as_date,
         seconds = msPlayed / 1000, minutes = seconds / 60)

# Playback activity per week and hours
streamingHours <- mySpotify %>%
  filter(date >= "2020-01-01") %>%
```

```
  group_by(date) %>%
  group_by(date = floor_date(date, "week")) %>%
  summarize(hours = sum(minutes) / 60) %>%
  arrange(date) %>%
  ggplot(aes(x = date, y = hours)) +
  geom_col(aes(fill = hours)) +
  scale_fill_gradient(low = "yellow", high = "red") +
  labs(x= "Date", y= "Hours of music playback") +
  ggtitle("Weekly Activity On Spoticy")
```

Let's view the plot

```
streamingHours
```



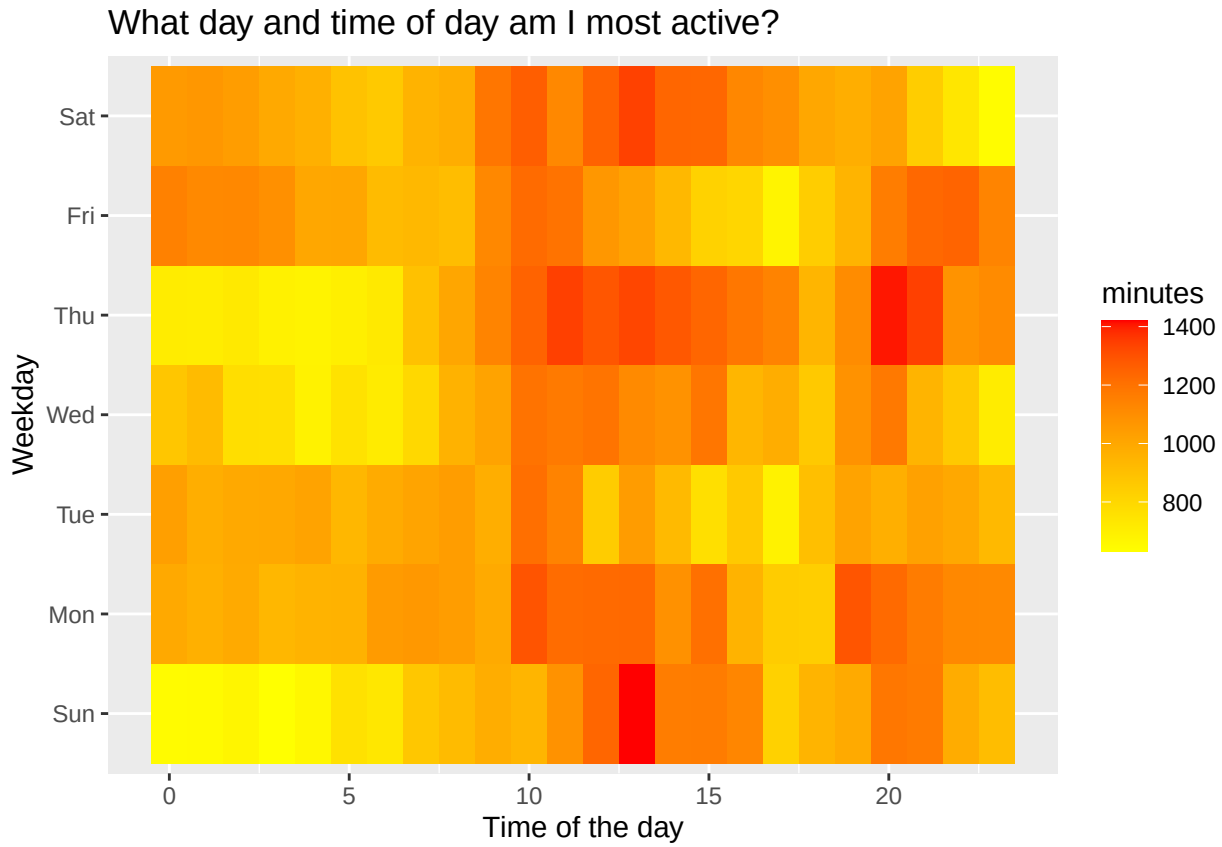
What time of the day am I most active?

```
hoursDay <- mySpotify %>%
  filter(date >= "2019-01-01") %>%
  group_by(date, hour = hour(endTime), weekday = wday(date, label = TRUE))%>%
  summarize(minutesListened = sum(minutes))
```

```
## `summarise()` has grouped output by 'date', 'hour'. You can override using the
## `.groups` argument.
```

```
hoursDay %>%
  group_by(weekday, hour) %>%
  summarize(minutes = sum(minutesListened)) %>%
  ggplot(aes(x = hour, weekday, fill = minutes)) +
  geom_tile() +
  scale_fill_gradient(low = "yellow", high = "red") +
  labs(x= "Time of the day", y= "Weekday") +
  ggtitle("What day and time of day am I most active?")
```

```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```



Here is the graph

```
hoursDay
```

```
## # A tibble: 4,449 x 4
## # Groups:   date, hour [4,449]
##    date        hour weekday minutesListened
##    <date>     <int> <ord>              <dbl>
##  1 2022-02-10    10 Thu                3.21
##  2 2022-02-18    19 Fri                0.989
##  3 2022-02-18    21 Fri                3.18
##  4 2022-02-19    18 Sat                5.06
```

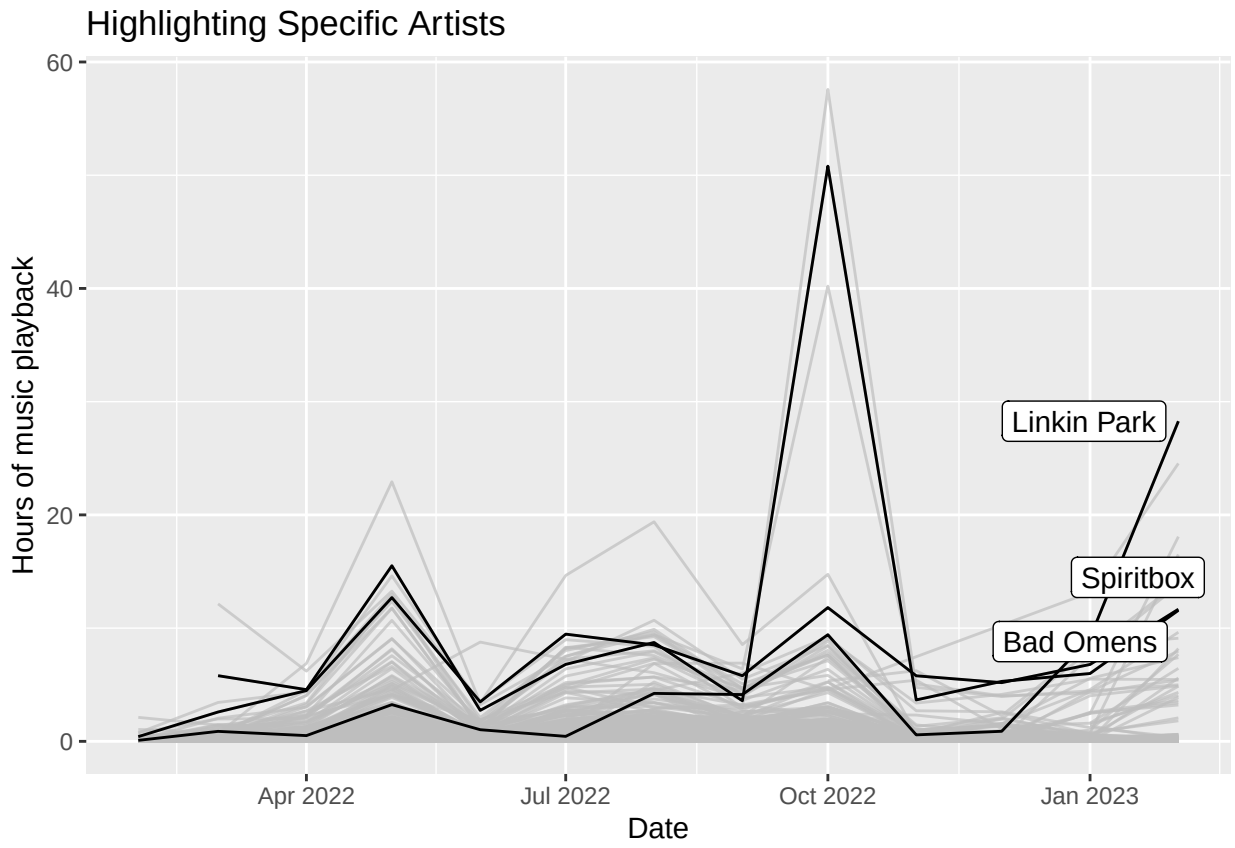
```
##  5 2022-02-19    22 Sat              20.9
##  6 2022-02-20    15 Sun               0.148
##  7 2022-02-20    16 Sun              40.7
##  8 2022-02-20    20 Sun               5.18
##  9 2022-02-20    21 Sun              39.1
## 10 2022-02-20    22 Sun               9.04
## # ... with 4,439 more rows
```

How about streaming time by a specific artist?

```
hoursArtist <- mySpotify %>%
  group_by(artistName, date = floor_date(date, "month")) %>%
  summarize(hours = sum(minutes) / 60) %>%
  ggplot(aes(x = date, y = hours, group = artistName)) +
  labs(x= "Date", y= "Hours of music playback") +
  ggtitle("Highlighting Specific Artists") +
  geom_line() +
  gghighlight(artistName == "Bad Omens" || artistName == "Spiritbox"
              || artistName == "Linkin Park")
```

Let's look at the graph

```
hoursArtist
```



Finally, let's get my most listened to artist(s) in 2022

```
topArtists <- mySpotify %>%
  filter(date >= "2022-01-01") %>%
  group_by(artistName) %>%
  summarize(minutesListened = sum(minutes)) %>%
  filter(minutesListened >= 1200) %>%
  ggplot(aes(x = artistName, y = minutesListened)) +
  geom_col(aes(fill = minutesListened)) +
  scale_fill_gradient(low = "yellow", high = "red") +
  labs(x= "Artist", y= "Minutes of music playback") +
  ggtitle("My Most Listened To Artists", "> 20 hours listened") +
  theme(axis.text.x = element_text(angle = 90))
```

And the plot

```
topArtists
```