R Final Project

Lucas Copsey

12/9/2019

Introduction

Music produced today is often short-lived. A new song might quickly reach the top of the charts only to sharply drop in popularity when the next hit song is released. Using data from Spotify, one of the most popular music streaming services, we will find the typical lifespan of top songs, as well as what songs remain at the top the longest.

Packages Required

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
# dplyr for data manipulation
# readr for reading/saving data
# stringr for joining strings
# purrr for error handling
# magrittr for piping
# ggplot2 for data visualization
```

Data Preparation

Spotify's top charts are readily available to download via https://spotifycharts.com/regional. Using the provided "Download to CSV" URL, I wrote functions to download the global, United States, and United Kingdom daily top 200 charts for a specified date. The data is available from January 1, 2017 to present, however a few dates are missing. I addressed this by replacing them with empty data frames.

```
# URL setup
base_url <- 'https://spotifycharts.com/regional'
global <- '/global/daily/'
usa <- '/us/daily/'
uk <- '/gb/daily/'
endpoint <- '/download'

# Set start and end dates
start_date <- as.Date("2017-01-01")
end_date <- as.Date("2019-12-06")

# Initialize empty dataframes
top200_global <- data.frame()
top200_usa <- data.frame()
top200_uk <- data.frame()

pull_charts_global <- function(date) {
  # Build URL
  global_url <- str_c(base_url, global, format.Date(date), endpoint)
  # Pull data from Spotify
  top200_global <- read.csv(global_url, encoding = "UTF-8")
  # Remove header rows and add date column
  top200_global <- top200_global %>% filter(X != "Position") %>% mutate(Date = date)
}

pull_charts_usa <- function(date) {
  # Build URL
  usa_url <- str_c(base_url, usa, format.Date(date), endpoint)
  # Pull data from Spotify
  top200_usa <- read.csv(usa_url, encoding = "UTF-8")
  # Remove header rows and add date column
  top200_usa <- top200_usa %>% filter(X != "Position") %>% mutate(Date = date)
}

pull_charts_uk <- function(date) {
  # Build URL
  uk_url <- str_c(base_url, uk, format.Date(date), endpoint)
  # Pull data from Spotify
  top200_uk <- read.csv(uk_url, encoding = "UTF-8")
  # Remove header rows and add date column
  top200_uk <- top200_uk %>% filter(X != "Position") %>% mutate(Date = date)
}

# Some days are missing, replace with empty data frame
pull_charts_global <- possibly(pull_charts_global, otherwise = top200_global <- data.frame())
pull_charts_usa <- possibly(pull_charts_usa, otherwise = top200_usa <- data.frame())
pull_charts_uk <- possibly(pull_charts_uk, otherwise = top200_uk <- data.frame())
```

Using a while loop, I then iterated through each date for the three regions, combining each day with the previous to create 3 complete data frames. This takes 30-40 minutes so I exported the data to avoid repeating this process.

```r
# Initialize date as start date
date <- start_date

# Pull charts for each day from start to end dates
while (date <= end_date) {
  # Combine to create one large data frame
  top200_global <- top200_global %>% rbind(pull_charts_global(date))
  top200_usa <- top200_usa %>% rbind(pull_charts_usa(date))
  top200_uk <- top200_uk %>% rbind(pull_charts_uk(date))
  date <- date + 1
}

# Assign meaningful column names
names <- c("Position", "Track_Name", "Artist", "Streams", "URL", "Date")
colnames(top200_global) <- names
colnames(top200_usa) <- names
colnames(top200_uk) <- names

# Save data
top200_global %>% write_csv("./Data/topcharts_global.csv")
top200_usa %>% write_csv("./Data/topcharts_usa.csv")
top200_uk %>% write_csv("./Data/topcharts_uk.csv")
```

```r
# Load data
top200_global <- read_csv("./Data/topcharts_global.csv")
top200_usa <- read_csv("./Data/topcharts_usa.csv")
top200_uk <- read_csv("./Data/topcharts_uk.csv")
```

I dropped the URL variable as I won't be using it, added a variable identifying the region, and combine the data into one data frame to avoid repetition of code.

The variables we are interested in are:

- Position - rank of track out of 200
- Track_Name - name of track
- Artist - artist of track
- Streams - number of streams
- Date - year/month/day of data
- Region - region of data

```r
global_data <- top200_global %>%
  select(-URL) %>%
  mutate(Region = "Global")
usa_data <- top200_usa %>%
  select(-URL) %>%
  mutate(Region = "USA")
uk_data <- top200_uk %>%
  select(-URL) %>%
  mutate(Region = "UK")
```

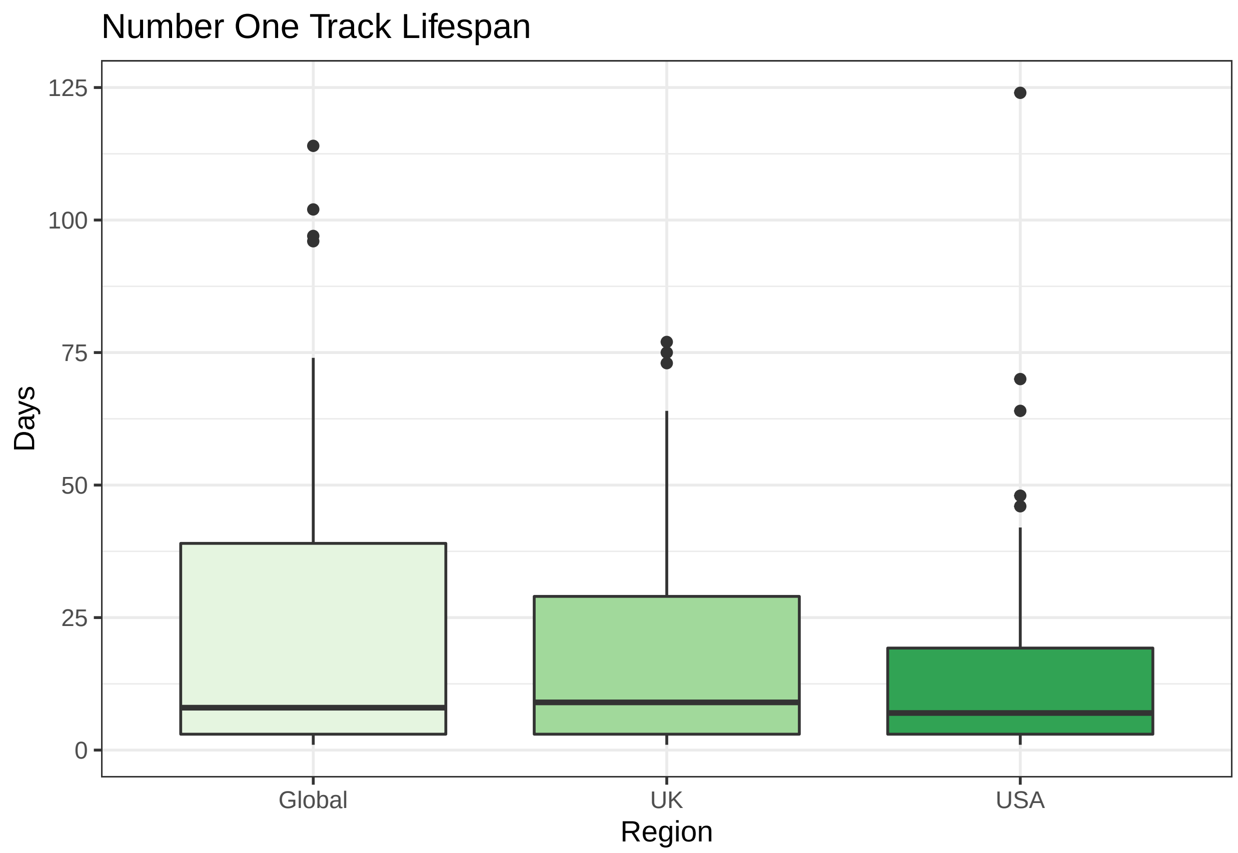
```
total_data <- rbind(global_data, usa_data, uk_data)

head(total_data)
```

```
## # A tibble: 6 x 6
##   Position Track_Name                          Artist       Streams Date       Region
##      <dbl> <chr>                               <chr>          <dbl> <date>     <chr>
## 1        1 Starboy                             The Weeknd   3135625 2017-01-01 Global
## 2        2 Closer                              The Chainsm~ 3015525 2017-01-01 Global
## 3        3 Let Me Love You                     DJ Snake     2545384 2017-01-01 Global
## 4        4 Rockabye (feat. Sean Paul & A~ Clean Bandit 2356604 2017-01-01 Global
## 5        5 One Dance                           Drake        2259887 2017-01-01 Global
## 6        6 Fake Love                           Drake        2137437 2017-01-01 Global
```

Exploratory Data Analysis

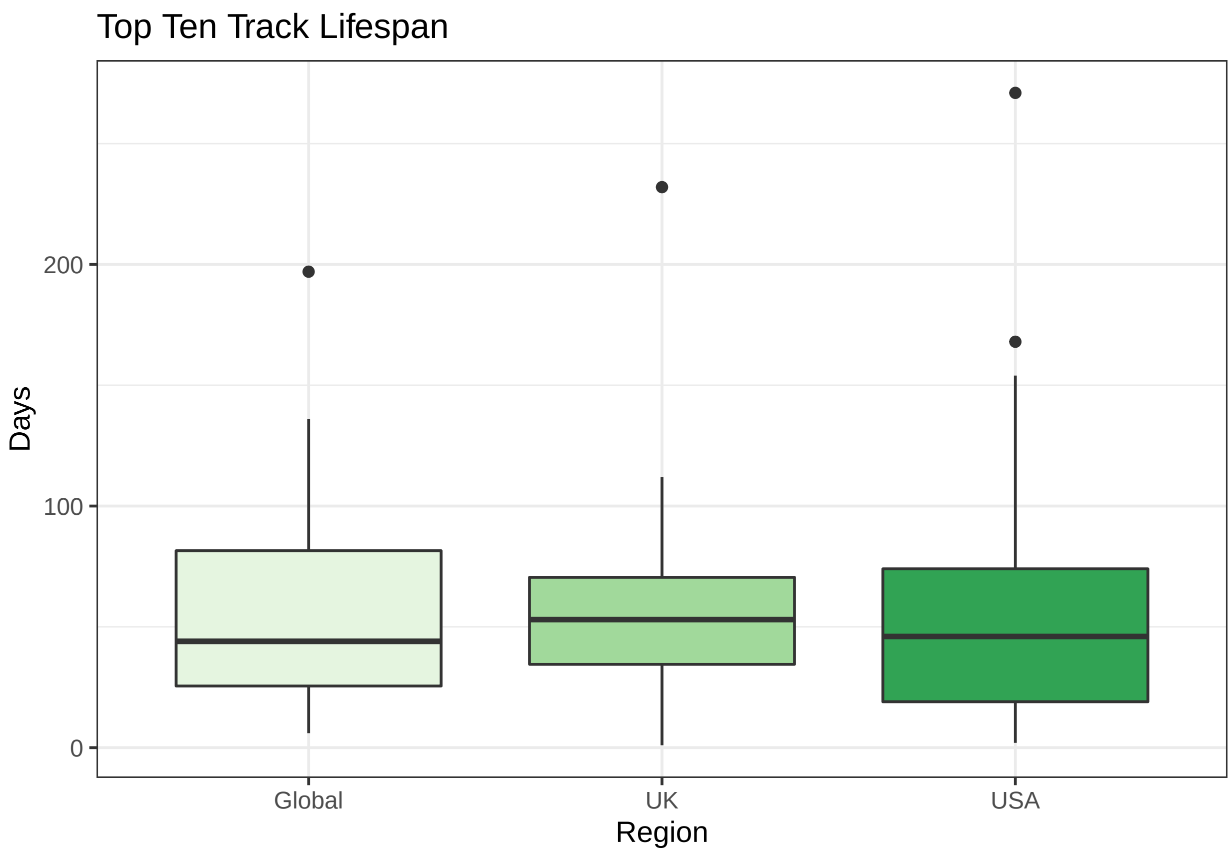
First I grouped the data by the three regions, and filtered to the tracks that have been in the number one position. I then counted how many days each track has been number one and used a box plot to show the distribution. We see that while a few outliers have held the number one position for as many as 124 days, the median number of days most tracks hold this position is 7-9 days.

```
num1data <- total_data %>%
  group_by(Region) %>%
  filter(Position == 1) %>%
  count(Track_Name)

ggplot(num1data, aes(x = Region, y = n)) +
  geom_boxplot(aes(fill = Region)) +
  theme_bw() +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Greens") +
  labs(title = "Number One Track Lifespan", y = "Days")
```



I then looked at just those tracks that have been number one, this time counting the number of days they were in positions two thru ten. We see again that a few outliers remain high in the charts for an extended time period, as many as 271 days, while the median number of days the previously number one tracks remain in the top ten is 44-53 days.

```
top10data <- total_data %>%
  semi_join(num1data, by = c("Track_Name", "Region")) %>%
  group_by(Region) %>%
  filter(Position > 1 & Position <= 10) %>%
  count(Track_Name)

ggplot(top10data, aes(x = Region, y = n)) +
  geom_boxplot(aes(fill = Region)) +
  theme_bw() +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Greens") +
  labs(title = "Top Ten Track Lifespan", y = "Days")
```



Looking at the outliers, we find the tracks and artists holding the number one position for the most days.

```
num1tracks <- total_data %>%
  filter(Region == "Global", Position == 1) %>%
  count(Track_Name) %>%
  arrange(desc(n)) %>%
  top_n(5)

num1artist <- total_data %>%
  filter(Region == "Global", Position == 1) %>%
  count(Artist) %>%
  arrange(desc(n)) %>%
  top_n(5)

num1tracks
```

```
## # A tibble: 5 x 2
##   Track_Name            n
##   <chr>             <int>
## 1 rockstar            114
## 2 Señorita            102
## 3 Shape of You         97
## 4 Despacito - Remix    96
## 5 God's Plan           74
```

```
num1artist
```

```
## # A tibble: 5 x 2
##   Artist            n
##   <chr>         <int>
## 1 Post Malone     174
## 2 Drake           158
## 3 Ed Sheeran      139
## 4 Ariana Grande   121
## 5 Shawn Mendes    102
```

Summary

What we've found is that while a select number of tracks hold their position as number one for longer periods of time, most only remain there around a week, and then remain in the top 10 for another month or two. Presumably this trend continues as tracks fall further down the chart and new tracks take their places. The artist holding the number one position the most was Post Malone, and the track was his song "rockstar".

This analysis is somewhat limited as additional data about each track could provide insight as to what factors make a song top the charts. Additionally, this analysis only gives a general idea of the track lifespan. Further analysis might provide additional insights.