DSA2101 Project

Introduction

Music is an integral part of a society's culture which is constantly changing and evolving. Music trends can be displayed through the The Billboard Hot 100, which is "the music industry standard record chart in the United States for songs" (Mock, 2021), leading us to explore the `billboard.csv` and `audio_features.csv` dataset from TidyTuesday. `billboard.csv` shows the values of the song's position each week from 1958 to 2021, instances that the song appears on the billboard and details of the performers, while `audio_features.csv` contains information of the song features based on Spotify metrics. By incorporating data from both tables, we aim to gain insights into the underlying musical characteristics that may contribute to chart success.

Reading in Data

```
tuesdata <- tidytuesdayR::tt_load('2021-09-14')
```

```
##
##  Downloading file 1 of 2: `billboard.csv`
##  Downloading file 2 of 2: `audio_features.csv`
```

```
billboard <- tuesdata$billboard
audio_features <- tuesdata$audio_features
```

Descriptive Statistics and Data Cleaning

Upon downloading the dataset from TidyTuesday, we explored the two datasets using `summary` which allowed us to validate the data type (which were all correct, but we used lubridate to make the `week_id` variable more accesible) and count the number of NAs of each variable (only a small proportion of the billboard dataset had NA values). Before using `left_join` to aggregate `billboard.csv` and `audio_features.csv` dataset, we performed `anti_join` on the key, `song_id`, and realised we had to standardise `song_id` by changing them into lower case (e.g., "Taylor Swift" and "taylor swift" should be the same value) in order to properly join the dataset.

Upon joining the dataset, we realised the `spotify_genre` variable required re-formatting and cleaning—each song had multiple genres and there were 1146 unique genres, with some sharing parts of the same name (e.g. 'k-pop' and 'pop rock' share 'pop'). Therefore, we first expanded the dataset by duplicating each song based on the number of genres. Then, we referenced a similar data analysis method on the same dataset and decided to group our genres together based on the main few genres: pop, rap, rock, jazz, metal, disco, edm, soul and blues (Lee et al., 2018). If a genre contains the name of one of the main genres, we group it under that genre (e.g. 'k-pop' will be in 'pop') but if more than one main genre appears, we group the genre under fusion (e.g. 'pop rock' to 'fusion'), which means more than a type of main genre under the genre column. (What is fusion music, 2023)

Clean Audio Features

```
audio_clean <- audio_features %>%
  mutate(song_id = str_to_lower(song_id))
```

Clean Billboard and join data

```
#' Helper functions
#'
##' Removes outer brackets, '[', ']' of `spotify_genres` column
remove_casing <- function(x){
  return(substr(x, 2, nchar(x)-1))
  }

##' Merges specific genres into a general one
##'
##' For example, 'k-pop' and 'j-pop' would be replaced by 'pop' instead
merge_big_genres <- function(x){
  if(is.na(x)){
    return(x)
  }
  if(str_detect(x, "pop") == TRUE){
    return("pop")
  }
  else if(str_detect(x, "rap") == TRUE){
    return("rap")
  }
  else if(str_detect(x, "rock") == TRUE){
    return("rock")
  }
  else if(str_detect(x, "jazz") == TRUE){
    return("jazz")
  }
  else if(str_detect(x, "metal") == TRUE){
    return("metal")
  }
  else if(str_detect(x, "disco") == TRUE){
    return("disco")
  }
  else if(str_detect(x, "hip hop") == TRUE){
    return("hip hop")
  }
  else if(str_detect(x, "contemporary") == TRUE){
    return("contemporary")
  }
  else if(str_detect(x, "blues") == TRUE){
    return("blues")
  }
  else if(str_detect(x, "soul") == TRUE){
    return("soul")
  }
  return(x)
}

##' Merges genres that involve two or more general genres into the 'fusion' genre
##'
##' For example, 'pop rock' and 'jazz metal' would be replaced by 'fusion' instead
merge_into_fusion <- function(x){
  count = 0
```

```
  if(is.na(x)){
    return(x)
  }
  if (str_detect(x, "fusion") == TRUE){
    return ("fusion")
  }
  types = c("pop | pop", "rap | rap", "rock | rock", "jazz | jazz", "metal | metal", "disco | disco", "
  for(i in types){
    if(str_detect(x, i)==TRUE){
      count = count + 1
      if (count == 2){
        return("fusion")
      }
    }
  }
  return(x)
}

##' Normalises values in a vector
normalise_ms <- function(vector){
  min_m = min(vector, na.rm = TRUE)
  max_m = max(vector, na.rm = TRUE)
  vector_norm = ifelse(!is.na(vector), (vector-min_m)/(max_m-min_m), NA)
  return(vector_norm)
}

#' Data Cleaning
#'
##' Dataframe that joins `billboard` & `audio_clean` tables together.
##'
##' @description
##' * Column `spotify_genre` has been split such that each row only has one genre under itself
##' * Contains normalised `spotify_track_duration_ms` column
##' * Addition of `year` column
merged <- billboard %>%

  #' Joining of tables `billboard` & `audio_clean`
  mutate(song_id = str_to_lower(song_id)) %>%
  left_join(audio_clean, by = "song_id", suffix = c("_billboard", "_audio")) %>%
  select(-c(url,performer_audio, spotify_track_id, song_audio, spotify_track_preview_url)) %>%
  mutate_at(vars(performer_billboard), str_to_lower) %>%

  #' Splitting of genres
  mutate_at(vars(spotify_genre), remove_casing) %>%
  mutate(spotify_genre = strsplit(as.character(spotify_genre), ",")) %>%
  unnest(spotify_genre) %>%
  mutate(spotify_genre = str_trim(gsub("'", "", spotify_genre))) %>%

  mutate_at(vars(week_id), mdy) %>%
  mutate(year = year(week_id),
         duration_mins = normalise_ms(spotify_track_duration_ms))

##' Dataframe that contains merged genres via the use of `merge_into_fusion()` and
```

```
##'  `merge_big_genres()` functions
##'
merged_genres <- merged %>%
  mutate(spotify_genre = lapply(spotify_genre, merge_into_fusion)) %>%
  mutate(spotify_genre = lapply(spotify_genre, merge_big_genres)) %>%
  mutate_at(vars(spotify_genre), as.character)
```

Question 1: How have music trends changed over the years?

Music is a cultural force that reflects the changes in a society (Rabinowitch, 2020). In this question, we hope to understand how music trends have transformed over the years based on the top genres and discover pivotal moments in the music scene, such as the emergence of new genres and the use of technology. Therefore, this question will make use of the `year` variable of our merged dataset and various features of the song such as the genre and song characteristics.

Methodology

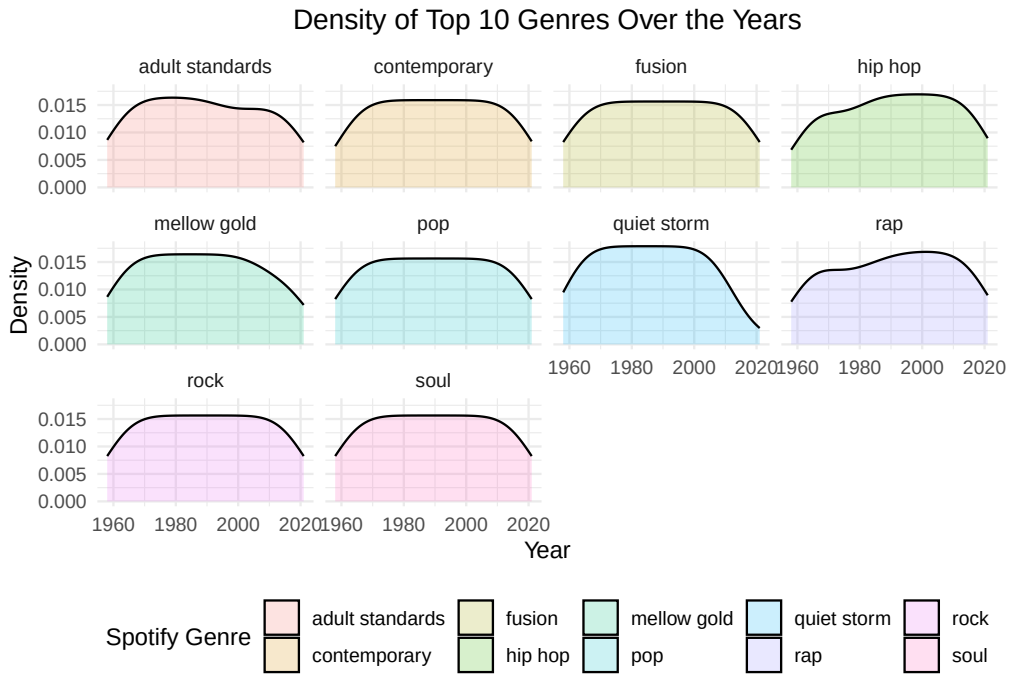
Using the cleaned `spotify_genre` variable, we first calculated the total number of instances each genre has appeared in the whole dataset to measure its overall popularity. Next, we extracted the top 10 genres over the entire dataset based on the number of instances and proceeded to examine the distribution of these top 10 genres over the years. Since we are working with a continuous data type, we used a density plot for this visualisation.

While researching how music genres have changed over the years, we learnt that 1991 "marked the most significant revolution in the history of modern pop music" (Thompson, 2015) due to the sudden rise in popularity of the rap genre. Therefore, we chose to use a butterfly plot for our second visualisation because the plot can visually show the differences in song characteristics for billboard songs before and after 1991. Selecting variables from the `audio_feature` datset such as danceability, speechiness, acousticness and eight other characteristics, we summarised the filtered data of each characteristic (using mainly the `mean` function) as shown in the function `create_sum` and displayed their values on the butterfly plot.

Plot 1a

```
##' Dataframe that contains the extraction of the top genres over the entire `merged_genres` dataset
top_genres <- merged_genres %>%
  group_by(spotify_genre) %>%
  summarize(n_songs = n()) %>%
  mutate(rank = rank(desc(n_songs))) %>%
  filter(rank <=10) %>%
  pull(spotify_genre)

##' Dataframe that contains spread of the top genres in `top_genres` over the years
top_overyears <- merged_genres %>%
  filter(spotify_genre %in% top_genres) %>%
  group_by(year, spotify_genre) %>%
  summarize(n_songs = n()) %>%
  arrange(year, desc(n_songs))

##' Creation of density plot
ggplot(top_overyears, aes(x = year, fill = spotify_genre)) +
  geom_density(alpha = 0.2) +
  facet_wrap(~ spotify_genre) +
  theme_minimal() +
```

```
  labs(
    title = "Density of Top 10 Genres Over the Years",
    x = "Year",
    y = "Density",
    fill = "Spotify Genre") +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "bottom")
```



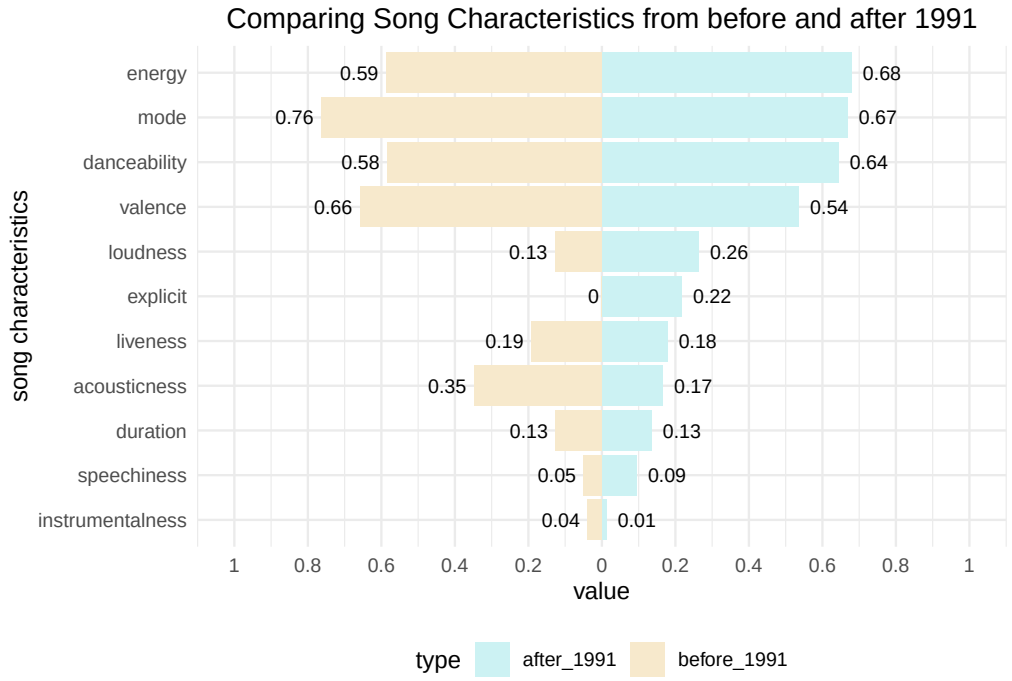
Plot 1b

```
#' Helper functions
#'
##' Summarises and obtain means of spotify audio_features from Dataframe
create_sum <- function(data){
  new <- data %>%
    select(-c(spotify_genre)) %>%
    distinct() %>%
    summarise(
      danceability = mean(danceability, na.rm = TRUE),
      energy = mean(energy, na.rm = TRUE),
      speechiness = mean(speechiness, na.rm = TRUE),
      acousticness = mean(acousticness, na.rm = TRUE),
      instrumentalness = mean(instrumentalness, na.rm = TRUE),
      liveness = mean(liveness, na.rm = TRUE),
      valence = mean(valence, na.rm = TRUE),
      explicit = sum(spotify_track_explicit, na.rm = TRUE)/n(),
      loudness = mean(ifelse(!is.na(loudness), 10^(loudness/10), NA), na.rm = TRUE),
      mode = mean(mode, na.rm = TRUE),
      duration = mean(duration_mins, na.rm = TRUE)
      )
  return(new)
}
```

```r
##' Transposes the dataframes & then row binds
##'
##' Takes in two summarised dataframes and their names
create_butterfly_data <- function(right, left, right_name, left_name){
  right_t <- right %>%
    gather(key="features", value="value") %>%
    mutate(type = right_name)
  left_t <- left %>%
    gather(key="features", value="value") %>%
    mutate(type = left_name, value = - 1*value)
  butterfly_data <- rbind(right_t, left_t)
  return(butterfly_data)
}
```

```r
##' Splitting of `merged` dataframe into two based on `year`
before_1991 <- merged %>%
  filter(year <= 1991)

after_1991 <- merged %>%
  filter(year > 1991)

butterfly_data <- create_butterfly_data(create_sum(after_1991), create_sum(before_1991), "after_1991",

##' Pull features in order of descending values (based on after_1991)
features_by_desc_values <- butterfly_data %>% filter(type == "after_1991") %>%
  arrange(value) %>% pull(features)

##' Creation of butterfly plot
ggplot(butterfly_data) +
  geom_bar(
    aes(x = factor(features, level = features_by_desc_values), y = value, fill = type),
    stat = 'identity'
    ) +
  scale_fill_manual(values = c("after_1991" = "#ccf2f3", "before_1991" = "#f7e9cc")) +
  geom_text(
    data = subset(butterfly_data, type == "after_1991"),
    aes(
      x = features,
      y = value,
      label = round(abs(value),2)
      ),
    size = 3,
    vjust = .5,
    hjust = -0.3
    ) +
  geom_text(
    data = subset(butterfly_data, type == "before_1991"),
    aes(x = features, y = value, label = round(abs(value),2)),
    size = 3,
    vjust = .5,
    hjust = 1.2
    ) +
  coord_flip(ylim = c(-1,1)) +
```

```
  scale_y_continuous(
    breaks = seq(-1,1,0.2),
    labels = paste0(as.character(c(rev(seq(0,1,0.2)),seq(0.2,1,0.2)), 'm'))
    ) +
  theme_minimal() +
  labs(
    y = "value",
    x = "song characteristics",
    title = "Comparing Song Characteristics from before and after 1991"
    ) +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "bottom")
```



Discussion

We can see that adult standards, quiet storm and mellow gold show a declining trend in recent years. Contrastingly, hip hop and rap have been showing an increasing trend starting in 1980s. Though the three genres are slowly fading away with the rise of hip hop and rap, their presence in the early years is still significant enough for these genres to remain in the top 10 genres across the dataset. Corresponding with Thompson's article in 2015, rap, hip hop and pop have maintained a high density in recent years, with pop consistently being a popular genre throughout the years.

Looking at the second visualisation, we learn that while some characteristics (such as energy, mode, danceability, liveness and duration) have minimal changes before and after 1991, loudness, speechiness and proportion of explicit songs have significantly increased by around 100% or more. Song characteristics such as acousticness and instrumentalness on the other hand, decreased by around 50% or more. Potential reasons that lead to this shift in music preference might be because it is easier for artists to "create a beat on a computer and drop in a voice note" and publish the music, as well as the use of streaming platforms like SoundCloud and Spotify which allow listeners to discover new types of songs (Bruner, 2018). In conclusion, both visualisations show that musical trends do change with time as technology and societal preferences influence the popularity of a song.

Question 2: How do music preferences vary based on the months of a year?

In this question, we decided to explore if there are recurring patterns in the top 100 billboard music. As human beings, our emotions are naturally influenced by the changing seasons (Behnke et.al., 2021), which prompted us to investigate music trends for each month over a decade from 2011 to 2020. By examining how music preferences shift throughout the year, we hope to uncover the relations between our emotional states and the changing seasons through the music we listen to.

Methodology

In the first visualisation, we aim to find how the average absolute change in song rankings changes each month using `week_position` and `prev_week_pos` which we created using `lag` function. We created our own variable despite `billboard` having a `previous_week_position` variable because we found one entry with the same `song_id` and `week_id` which might affect the validity of the `previous_week_position` (namely 'Unchained MelodyThe Righteous Brothers'). This shows us how much music preferences are changing each month and lets us identify possible trends. We do this by finding the absolute difference between a song's current position and its position in the previous week, and then taking the mean of the sum of these absolute differences for each month. To better contextualise this dataset, we look at the proportion of songs that stay on the billboard for less than four weeks and include the total number of songs in that month, which gives us more insight into the volatility of music taste and popularity.

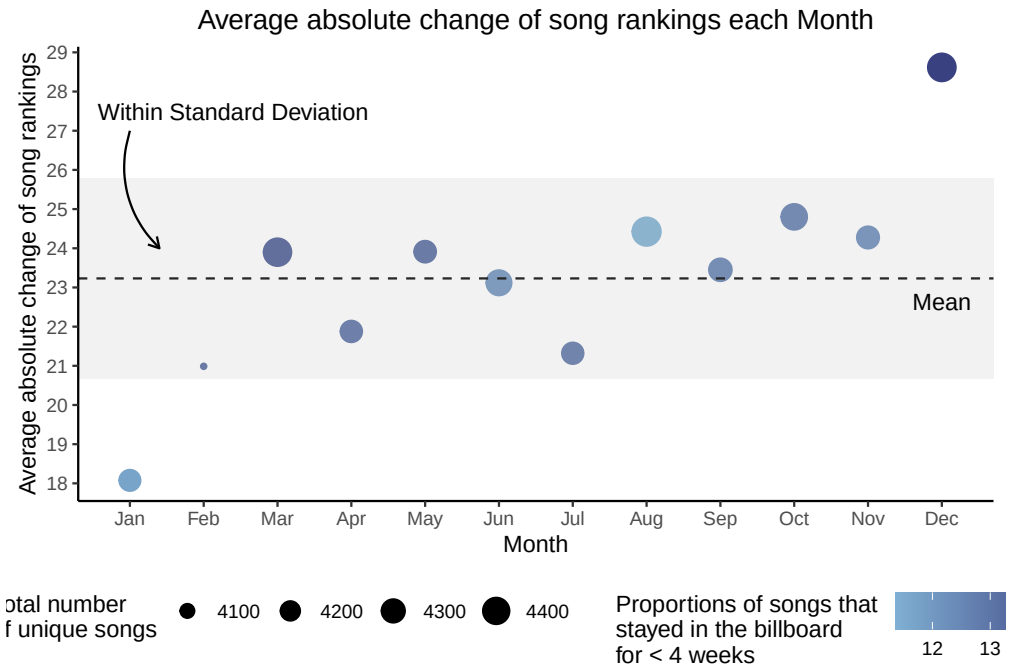
For the second visualisation, we used `valence`, `month` and `year` as the main variables to examine how the average music positiveness of songs change each month within 10 years. For the `valence` variable, songs with high-valence are happier, cheerier, and more euphoric (positive), whereas songs with low-valence are more depressing and sad (negative) (Mock, 2021). We grouped the data by `month` and `year` then summarized the mean valence of the songs from the grouped data by excluding the NA value. We used boxplot for this visualisation because it provides a visual summary of the numeric values at a glance. Hence, it will be easier for us to visualise the data and compare the mean of the yearly mean valence, by month.

Plot 2a

```
##' Dataframe that selects only the necessary distinct rows from year 2011-2020
unique_songs <- merged %>%
  filter(year %in% c(2011:2020)) %>%
  select(c(year, week_id, song_id, week_position, peak_position)) %>%
  mutate(month = month(week_id, label = TRUE, abbr = TRUE), week = day(week_id)) %>%
  distinct()

##' Dataframe with the average absolute change in song rankings and the respective year
ranking_change <- unique_songs %>%
  group_by(month) %>%
  group_by(song_id) %>%
  arrange(week_id) %>%
  arrange(month, song_id) %>%
  #' Lag the week_position to obtain previous week position of song ranking
  mutate(prev_week_pos = lag(week_position)) %>%
  na.omit() %>%
  ungroup(month, song_id) %>%
  mutate(diff = abs(prev_week_pos-week_position)) %>%
  group_by(month) %>%
  summarise(abs_change = mean(sum(diff)/length(unique(song_id)), na.rm = TRUE))

##' Data frame with the total number of songs that stay in the top 100 billboards for less than 4 weeks
```

```r
##'
##' It is also joined with `ranking_change`
drop_out <- unique_songs %>%
  group_by(month, song_id) %>%
  summarise(count = n()) %>%
  filter(count < 4) %>%
  group_by(month) %>%
  summarise(count_drop_out = n()) %>%
  full_join(ranking_change)

##' Dataframe that contains the join of `drop_out` and `ranking_change` and has a count of total number
unique_songs_df <- unique_songs %>%
  group_by(month) %>%
  summarise(count_total = n()) %>%
  full_join(drop_out) %>%
  mutate(proportion_drop_out = 100*count_drop_out/count_total)

##' Obtaining statistics for plotting
mean_abs_change <-mean(unique_songs_df$abs_change)
sd_abs_change <- sd(unique_songs_df$abs_change)

##' Creation of Scatterplot
ggplot(unique_songs_df) +
  geom_point(aes(x = month, y = abs_change, size = count_total, color = proportion_drop_out)) +
  scale_color_continuous(trans = "reverse") +
  theme_classic() +
  labs(
    y = "Average absolute change of song rankings",
    x = "Month",
    title = "Average absolute change of song rankings each Month",
    color = "Proportions of songs that \nstayed in the billboard \nfor < 4 weeks",
    size = "Total number \nof unique songs"
    ) +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5)) +
  scale_color_gradient(low = "#80b1d3", high = "#3a4180") +
  scale_y_continuous(breaks= seq(15,30,by=1)) +
  geom_hline(yintercept = mean_abs_change, linetype = 2) +
  annotate(
    'rect',
    xmin = 0.3,
    xmax = 12.7,
    ymin = mean_abs_change - sd_abs_change,
    ymax = mean_abs_change + sd_abs_change,
    alpha=.2,
    fill='grey'
    ) +
  annotate(
    "text",
    x = 12,
    y = mean_abs_change - 0.6,
    label = "Mean"
    ) +
  annotate(
```

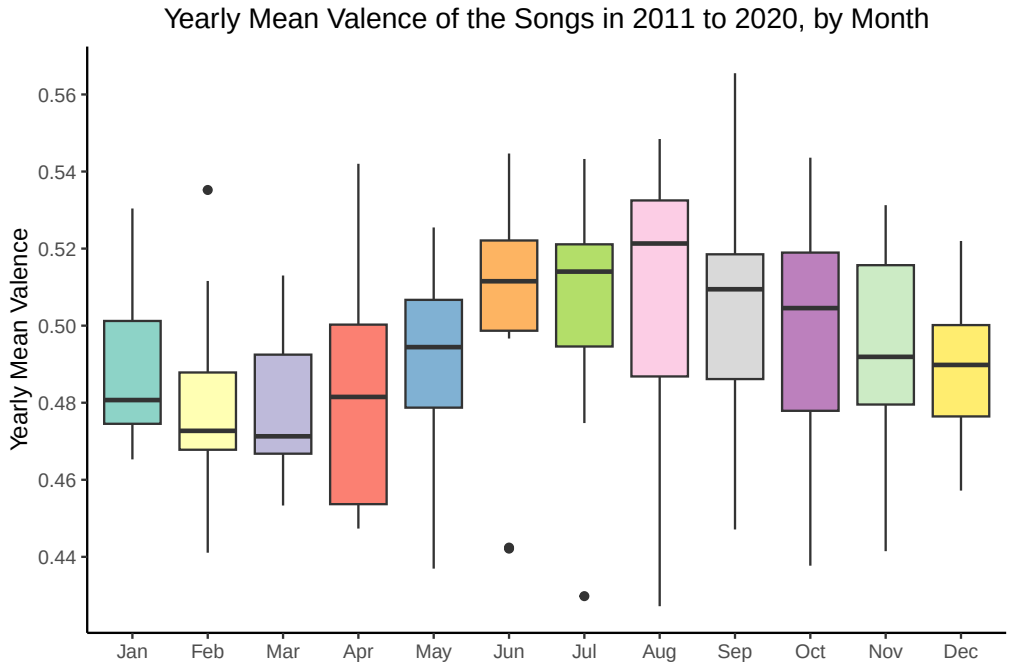
```
    geom = "curve",
    x = 1,
    y = 27,
    xend = 1.4,
    yend = 24,
    curvature = .3,
    arrow = arrow(length = unit(2, "mm"))
    ) +
  annotate("text", x = 2.4, y = 27.5, label="Within Standard Deviation")
```



Plot 2b

```
##' Dataframe that selects only the necessary distinct rows from year 2011-2020 and contains
##'  summarised values of `mean_valence`
billboard_valence_month <- merged %>%
  mutate(month = month(week_id, label = TRUE, abbr = TRUE)) %>%
  select(year, month, song_billboard, valence) %>%
  distinct() %>%  filter(year %in% c(2011:2020)) %>%
  group_by(month, year) %>%
  summarise(mean_valence= mean(valence, na.rm=TRUE),.groups='drop')
```

```
##' Creation of Boxplot
ggplot(billboard_valence_month, aes(x=month, y=mean_valence, fill=month)) +
  geom_boxplot() +
  labs(x="",
       y="Yearly Mean Valence",
       title ="Yearly Mean Valence of the Songs in 2011 to 2020, by Month") +
  scale_y_continuous(breaks = seq(0.42,0.58,by=0.02)) +
  scale_fill_brewer(type="qual", palette = "Set3") +
  theme_classic() +
  theme(legend.position = "none", plot.title = element_text(hjust=0.5))
```



Discussion

In December and January, there are significant differences in the average absolute change of song rankings compared to the other months. December also has a higher proportion of songs that stay on the billboard for less than four weeks and a greater number of unique songs, indicating greater volatility in the top 100 rankings. Conversely, January has a smaller proportion of "short-lived" songs and fewer unique songs, showing that there is less volatility. While we cannot determine a causation between the number/proportion of songs and average absolute change of song rankings (it is only a correlation), these differences may be attributed to the increased music consumption in December, especially for holiday-themed or festive songs, as well as the release of year-end "best of" lists, awards and movies (Ordanini, & Nunes, 2016). In contrast, January sees people returning to work or school after the holidays, leading to less focus on chart trends and music listening.

For the second visualisation, we observe that the song valence is higher in the summer and early autumn (Jun to Oct) than in the late autumn to early spring (Nov to May). This supports the theory that seasonal changes may impact people's musical preferences since people choose to listen to happier songs (with higher valence) during the summer because it is a lively season in contrast to the calm winter season (Lagarto, 2022). Moreover, the valence level of the songs during the late autumn to winter season (November and December) is slightly higher than in January to March, since November and December are close to holiday seasons (Christmas and New Year), meaning that people may tend to listen to more festive songs. Although the range of valence values are around 0.5, showing that the top 100 billboard songs are generally both positive and negative, the box plot can clearly show where the yearly valence averages are distributed and displays a wave-like pattern throughout the year. Therefore, both visualisations are able to show a general pattern in music preference over the years of 2011 to 2020.

Reference

- BBC. (n.d.). What is fusion music?. Retrieved April 13, 2023, from https://www.bbc.co.uk/bitesize/guides/zddbhbk/revision/1#:~:text=Fusion%20music%20is%20a%20blend,and%20blues%20and%20country%20music.
- Behnke, M., Overbye, H., Pietruch, M., & Kaczmarek, L. D. (2021). How seasons, weather, and part of day influence baseline affective valence in laboratory research participants? PLOS ONE, 16(8), e0256430. https://doi.org/10.1371/journal.pone.0256430

- Britannica. (2023, March 2). Jazz-rock. Retrieved April 13, 2023, from https://www.britannica.com/art/jazz-rock
- Bruner, R. (2018, January 25). Kendrick Lamar to Migos: How Rap Became Sound of Mainstream | Time. TIME. Retrieved April 13, 2023, from https://time.com/5118041/rap-music-mainstream/

- Lagarto, I. (2022, November 20). Songs and the senses: Why our music preferences shift with the seasons. Retrieved April 13, 2023, from https://www.thepalmettopanther.com/songs-and-the-senses-why-our-music-preferences-shift-with-the-seasons/
- Lee, J., Wen, Y., & Schaich, K. (2018, September 17). Billboard. Retrieved April 13, 2023, from https://github.com/kevinschaich/billboard
- Ordanini, A., & Nunes, J. C. (2016, June). From fewer blockbusters by more superstars to more blockbusters by fewer superstars: How technological innovation has impacted convergence on the music chart. Retrieved April 13, 2023, from https://www.sciencedirect.com/science/article/abs/pii/S0167811615001081
- Rabinowitch, T.-C. (2020). The potential of music to effect social change. Music & Science, 3, 205920432093977. https://doi.org/10.1177/2059204320939772
- Thomas Mock (2021, September 14). Tidy Tuesday: Top 100 Billboard. Retrieved April 13, 2023, from https://github.com/rfordatascience/tidytuesday/blob/master/data/2021/2021-09-14/readme.md.
- Thompson, D. (2015, May 8). In 1991, Rap Changed Pop Music Forever. The Atlantic. Retrieved April 13, 2023, from https://www.theatlantic.com/culture/archive/2015/05/1991-the-most-important-year-in-music/392642/