Introduction
Data Load
Data Cleaning
Summary
- Summary statistics of user scores
- Summary statistics of metacritic scores
- Summary statistics of her top 3 albums based on user and metacritic scores
- Summary statistics of number of lyrics

Question 1: Does number of lyrics or album duration affect the appeal of Taylor's albums?
- Introduction to Question 1
- Methodology
- Visualisation
- Discussion

Question 2: Is the energy score of Taylor Swift tracks an indicator of their appeal?
- Introduction to Question 2
- Methodology
- Visualisation
- Discussion

Question 3: Does choice of words in her lyrics affect the appeal of Taylor's albums?
- Introduction to Question 3
- Methodology
- Visualisation
- Discussion

Overall conclusion
Teamwork
References
Introduction
Data Load
Data Cleaning
Summary
- Summary statistics of user scores
- Summary statistics of metacritic scores
- Summary statistics of her top 3 albums based on user and metacritic scores
- Summary statistics of number of lyrics

Question 1: Does number of lyrics or album duration affect the appeal of Taylor's albums?
- Introduction to Question 1
- Methodology
- Visualisation
- Discussion

Question 2: Is the energy score of Taylor Swift tracks an indicator of their appeal?
- Introduction to Question 2
- Methodology
- Visualisation
- Discussion

Question 3: Does choice of words in her lyrics affect the appeal of Taylor's albums?
- Introduction to Question 3

Methodology
Visualisation
Discussion
Overall conclusion
Teamwork
References

DSA2101 Project: Taylor Swift

Annika Law Jie Yu (A0257136A)
Beatriz Khoh (A0257343A)
Hon Mun Dai Darren (A0234121Y)
Tan Shayne (A0259184U)
Zhang Changfeng (A0245873U)

2024-11-15

Introduction

Our group chose to explore the Taylor Swift dataset to investigate the factors contributing to her remarkable popularity. As the Eras Tour is reported to be one of the highest-grossing world tours, Swift's music continues to captivate audiences with her songs and top charts globally. One question that interests us the most is - what are the factors contributing to Taylor Swift's widespread public reception? We will look into the potential reasons by analyzing patterns in her songs. As such, we aim to reveal insights with 3 visualisations - scatter plot of number of lyrics and duration, histogram and violin plots of energy scores, and barplots of her most frequently used lyrics. Given that Swift has over 200 songs, our group will analyse the data across her different albums for clearer comparisons.

Our main dataset is taylor_all_songs which has her entire discography - her songs, album that the song belongs to and factors such as danceability score, energy score, liveness and loudness. It also includes lyrics where it displays the song's lyrics. The other dataset which we have used is taylor_albums which summarises her album release history with the album names, metacritic scores and user scores.

```
library(tidyverse)
library(readxl)
library(stringr)
library(lubridate)
library(tidytuesdayR)
library(taylor)
library(tidytext)
library(grid)
library(ggplot2)
library(viridis)
library(gridExtra)
library(ggthemes)
library(ggrepel)
library(cowplot)
```

Data Load

```
tuesdata <- tidytuesdayR::tt_load('2023-10-17')
taylor_all_songs <- tuesdata$taylor_all_songs
taylor_albums <- tuesdata$taylor_albums

taylor_albums_clean = taylor_albums %>%
  na.omit()
```

Data Cleaning

```
taylor_albums_clean = taylor_albums %>%
  na.omit() %>%  ## remove the 2 ep releases
  mutate(metacritic_score = metacritic_score / 10,
         average_score = (metacritic_score + user_score) / 2) %>%
  arrange(desc(average_score)) ## finding best album by taking average of scores

#cleaning based on below checks
taylor_all_songs_clean = taylor_all_songs %>%
  filter(!is.na(album_name) | !is.na(album_release)) %>%
  filter(ep == FALSE)

#check uniqueness of key
taylor_albums_clean %>% summarise(n = n_distinct(album_name, album_release))
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    12
```

```
taylor_all_songs_clean %>% summarise(n = n_distinct(album_name, album_release))
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    12
```

```
#Check missing values in key
taylor_albums_clean %>% filter(is.na(album_name)|is.na(album_release))
```

```
## # A tibble: 0 × 6
## # ℹ 6 variables: album_name <chr>, ep <lgl>, album_release <date>,
## #   metacritic_score <dbl>, user_score <dbl>, average_score <dbl>
```

```
taylor_all_songs_clean %>% filter(is.na(album_name)|is.na(album_release))
```

```
## # A tibble: 0 × 29
## # ℹ 29 variables: album_name <chr>, ep <lgl>, album_release <date>,
## #   track_number <dbl>, track_name <chr>, artist <chr>, featuring <chr>,
## #   bonus_track <lgl>, promotional_release <date>, single_release <date>,
## #   track_release <date>, danceability <dbl>, energy <dbl>, key <dbl>,
## #   loudness <dbl>, mode <dbl>, speechiness <dbl>, acousticness <dbl>,
## #   instrumentalness <dbl>, liveness <dbl>, valence <dbl>, tempo <dbl>,
## #   time_signature <dbl>, duration_ms <dbl>, explicit <lgl>, key_name <chr>, …
```

```
#check unmatched rows
taylor_all_songs_clean %>% anti_join(taylor_albums_clean, by = c("album_name", "album_release"))
```

```
## # A tibble: 0 × 29
## # ℹ 29 variables: album_name <chr>, ep <lgl>, album_release <date>,
## #   track_number <dbl>, track_name <chr>, artist <chr>, featuring <chr>,
## #   bonus_track <lgl>, promotional_release <date>, single_release <date>,
## #   track_release <date>, danceability <dbl>, energy <dbl>, key <dbl>,
## #   loudness <dbl>, mode <dbl>, speechiness <dbl>, acousticness <dbl>,
## #   instrumentalness <dbl>, liveness <dbl>, valence <dbl>, tempo <dbl>,
## #   time_signature <dbl>, duration_ms <dbl>, explicit <lgl>, key_name <chr>, …
```

```
#Joining 2 tables
taylor_data = taylor_all_songs_clean %>%
  left_join(taylor_albums_clean, by = c("album_name", "album_release")) %>%
  select(album_name, track_name, danceability, energy, speechiness,
         acousticness, instrumentalness, liveness, valence, duration_ms, metacritic_score, user_score, average_scor
e, album_release)

# 3 songs from Midnights album with missing (NA) values for danceability, liveness, energy etc.
# filter out these rows because the NA values3 will affect later calculations of mean
taylor_data = taylor_data %>% filter(!is.na(danceability))
```

```
#Lyrics data
data = taylor::taylor_all_songs #tibbles of lyrics within a tibble
lyrics = data %>%
  select(album_name, track_name, lyrics) %>%
  unnest(lyrics) %>% ## unnest nested lyrics tibble
  select(album_name, track_name, lyric) %>%
  group_by(album_name) %>%
  summarize(lyric = str_c(lyric, collapse = " ")) %>% #Combine all lyrics for a song
  unnest_tokens(word, lyric) %>% # separate each word - 1 row per word
  count(album_name, word) %>% # count number of combinations of album_name with same word
  ungroup()
```

Summary

Summary statistics of user scores

```
summary_user_score = taylor_albums_clean %>%
  select(album_name, user_score)
summary(summary_user_score)
```

```
##   album_name          user_score
##  Length:12          Min.   :8.200
##  Class :character   1st Qu.:8.375
##  Mode  :character   Median :8.500
##                     Mean   :8.583
##                     3rd Qu.:8.900
##                     Max.   :9.000
```

Summary statistics of metacritic scores

```
summary_meta_score = taylor_albums_clean %>%
  select(album_name, metacritic_score)
summary(summary_meta_score)
```

```
##   album_name        metacritic_score
##  Length:12          Min.   :6.700
##  Class :character   1st Qu.:7.525
##  Mode  :character   Median :7.800
##                     Mean   :7.925
##                     3rd Qu.:8.500
##                     Max.   :9.100
```

Summary statistics of her top 3 albums based on user and metacritic scores

```
top_albums = taylor_albums_clean %>%
  arrange(desc(average_score)) %>%
  select("Album Name" = album_name, "Average Score" = average_score)

print(head(top_albums, 3))
```

```
## # A tibble: 3 × 2
##   `Album Name`           `Average Score`
##   <chr>                            <dbl>
## 1 Red (Taylor's Version)            9.05
## 2 folklore                          8.9
## 3 evermore                          8.7
```

Summary statistics of number of lyrics

```
no_of_words = lyrics %>%
  summarise(unique = n_distinct(word))
cat("Taylor Swift uses a total of", no_of_words$unique[1], "unique words in her entire discography")
```

```
## Taylor Swift uses a total of 5057 unique words in her entire discography
```

Question 1: Does number of lyrics or album duration affect the appeal of Taylor’s albums?

Introduction to Question 1

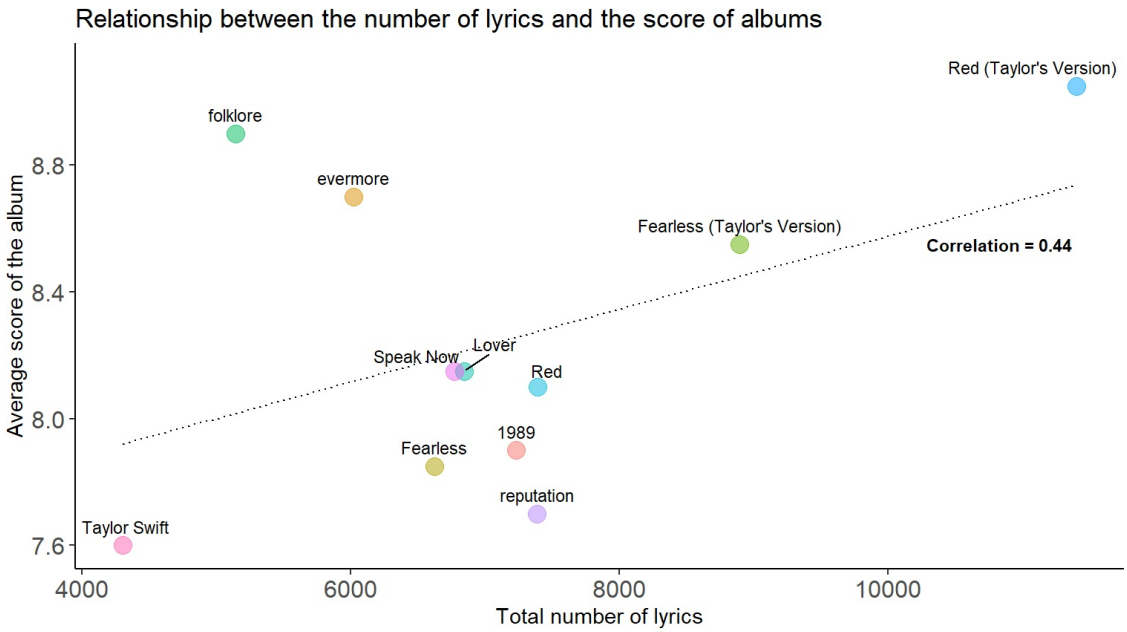
In the first visualisation, we want to explore the relationship between Taylor Swift’s album scores and two factors: total number of lyrics and album duration.

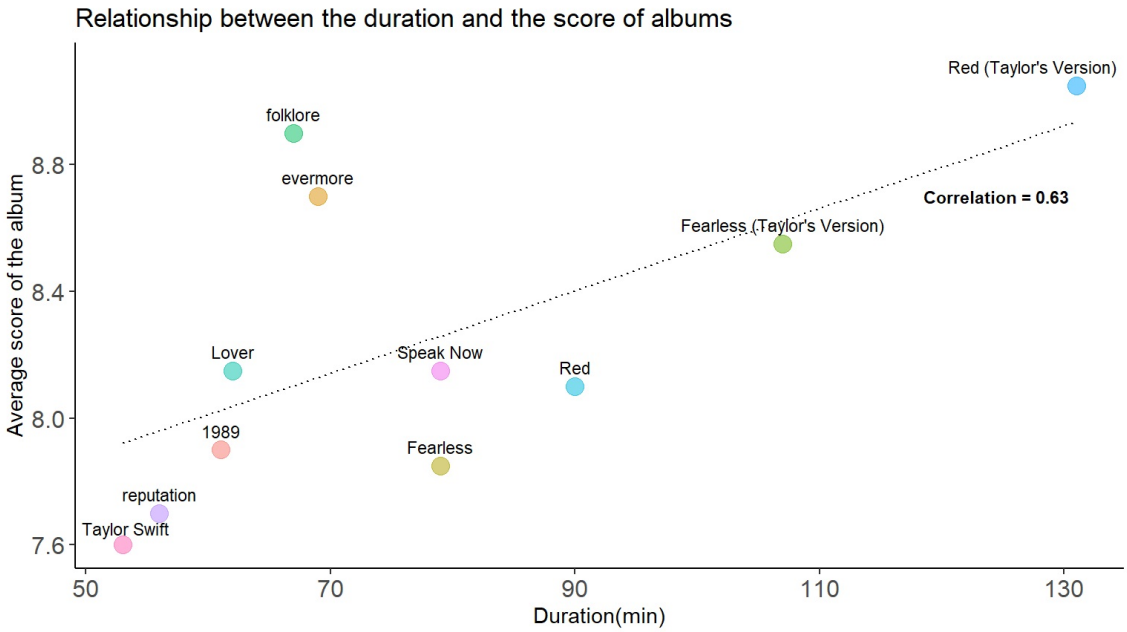
Methodology

First, we summarized the total number of lyrics and total duration for each album. We removed the album “Midnight” from this summary due to missing data in lyrics and duration. Next, we performed an inner join with the taylor_albums_clean table, which contains the average score for each album. Finally, we created two scatter plots to examine the relationships. The first plot shows the average album score versus the total number of lyrics, while the second plot shows the average album score versus the album duration. The albums are labeled to highlight corresponding points, and the trendlines with correlation coefficients help assess the strength of these relationships.

Visualisation

```r
vs1_1 = taylor_all_songs_clean %>%
  group_by(album_name) %>%
  summarise(total_duration = sum(duration_ms)) %>% # summerise the total duration
  na.omit() %>% # omit the album that have unknown duration which is midnight
  mutate(total_duration = round(total_duration/(1000*60), 0)) # change unit from ms to min

vs1_2 = lyrics %>%
  group_by(album_name) %>%
  summarise(total_n = sum(n)) # summarise total lyrics

vs1_3 = vs1_2 %>%
  inner_join(taylor_albums_clean, by = "album_name") %>%
  inner_join(vs1_1, by = "album_name") %>%
  relocate(total_duration, .after = total_n) %>%
  arrange(desc(average_score))

vs1_4 <- vs1_3 %>%
  filter(!album_name %in% c("folklore", "evermore")) # remove these two albums for further analysis

cor1 <- with(vs1_3, cor(total_n, average_score, use = "complete.obs")) # coefficient for lyrics amount
cor2 <- with(vs1_3, cor(total_duration, average_score, use = "complete.obs")) # # coefficient for duration
cor12 <- with(vs1_4, cor(total_n, average_score, use = "complete.obs")) # coefficient for lyrics amount after remov
ing folklore,evermore
cor22 <- with(vs1_4, cor(total_duration, average_score, use = "complete.obs")) # coefficient for duration after rem
oving folklore,evermore

p1 = ggplot(vs1_3, aes(x = total_n, y = average_score)) +
  geom_point(aes(color = album_name), alpha = 0.5, size = 4) +
  geom_smooth(method = "lm", formula = y ~ x,
              color = "black", se = FALSE,
              lty = 3, lwd = 0.5) +
  annotate("text",
           x = max(vs1_3$total_n) * 0.95,
           y = 8.55,
           label = sprintf("Correlation = %.2f", cor1),
           size = 3,
           fontface = "bold") +
  geom_text_repel(aes(label = album_name), size = 3, nudge_y = 0.06) +
  guides(color = "none") +
  labs(title = "Relationship between the number of lyrics and the score of albums",
       x = "Total number of lyrics",
       y = "Average score of the album") +
  theme_classic() +
  theme(axis.text = element_text(size = 12))

p2 = ggplot(vs1_3, aes(x = total_duration, y = average_score)) +
  geom_point(aes(color = album_name), alpha = 0.5, size = 4) +
  geom_smooth(method = "lm", formula = y ~ x,
              color = "black", se = FALSE,
              lty = 3, lwd = 0.5) +
  annotate("text",
           x = max(vs1_3$total_duration) * 0.95,
           y = 8.7,
           label = sprintf("Correlation = %.2f", cor2),
           size = 3,
           fontface = "bold") +
  geom_text_repel(aes(label = album_name), size = 3, nudge_y = 0.06) +
  guides(color = "none") +
  labs(title = "Relationship between the duration and the score of albums",
       x = "Duration(min)",
       y = "Average score of the album") +
  theme_classic() +
  theme(axis.text = element_text(size = 12))

grid.arrange(p1, p2, nrow = 2)
```





Discussion

In the first graph, the trend shows a positive relationship between the total number of lyrics in an album and its average score. The relationship is not really strong, with a correlation coefficient of 0.44, but it still suggests that albums with a higher lyric count may tend to receive higher scores. However, this correlation is moderate, meaning that although there is a general trend, lyric amount alone is not a strong predictor of album score. Special cases like “folklore” and “evermore” have high scores with fewer lyrics, suggesting that other factors may contribute significantly to an album’s score.

The second graph shows a stronger positive relationship between album duration and score, with a moderately strong correlation coefficient of 0.63. This indicates that album duration is a more reliable indicator of higher scores compared to the number of lyrics. Longer albums, such as “Red (Taylor’s Version)” and “Fearless (Taylor’s Version),” tend to receive better scores, implying that people may be more satisfied with the albums with longer duration or more songs. However, the special cases of “folklore” and “evermore” still have high scores with shorter duration.

In conclusion, while both lyric amount and album duration show positive correlations with album scores, album duration seems to be a more reliable predictor of higher scores. However, the moderate correlation with lyric amount and the moderately strong correlation with album duration show that both factors have influence but are not the most significant drivers of an album’s success. High-scoring albums like “folklore” and “evermore,” which have fewer lyrics and shorter duration, indicate that other factors may play a more significant role in influencing the album scores.

One possible reason for the success of “folklore” and “evermore” is the timing of their release during the COVID-19 pandemic, a period when people had to spend more time at home and turned to indoor activities such as listening to music for comfort and relax. Moreover, the introspective and hopeful themes of these albums also resonated deeply with listeners during this time, possibly boosting the albums’ popularity and resulting in higher scores.(American Songwriter, 2021)

To further understand this possible effect, the correlation coefficients were recalculated after removing the “folklore” and “evermore” from the data. Without these albums, the correlation coefficients increased to 0.90 for lyric amount and 0.93 for album duration, indicating much stronger relationships between these factors and album score. This shows that “folklore” and “evermore” are indeed “outliers”, with their high scores likely influenced by other factors. Consequently, while the albums with more lyrics or longer duration generally receive higher scores from this analysis, context, thematic resonance and other factors can also significantly impact an album’s success, as seen with “folklore” and “evermore”. Therefore, we will continue to explore other possible factors in the following text.

Question 2: Is the energy score of Taylor Swift tracks an indicator of their appeal?

Introduction to Question 2

In our second visualisation, we were interested in figuring out if we can determine the popularity, (indicated by the metacritic scores) of Taylor Swift's albums by looking at their energy scores.

Methodology

In this visualisation, we created 2 violin plots that show the trend of energy scores in Taylor Swift songs across different albums. The objective of this visualisation is to explore how the energy scores of Swift's albums vary, and how these variations correlate with her albums' popularity. The energy score ranges from 0.0 to 1.0, and is a measure of tracks' intensity and activity.

For both graphs, the horizontal axis shows the energy score, and the vertical axis lists albums released by Taylor Swift in chronological order by release date. We decided to use a gradient to colour the violin plot based on the metacritic scores of each album, reflecting their popularity rankings. deeper red indicates a higher metacritic score and hence higher popularity rank, whereas lighter reds (closer to white) indicate lower metacritic scores and hence lower popularity rank.

We chose to use the violin plot as it shows an easily interpretable distribution of energy scores across each album. This allows viewers to quickly identify the mean, while visually showing the spread of energy scores in each album (i.e. wider violin bulge for albums with more songs with certain energy scores).

Visualisation

Preliminary Data Cleaning

```
# sorting the albums by release date
sorted_albums = taylor_data[order(taylor_data$album_release), ]
sorted_albums = unique(as.character(sorted_albums$album_name))
#print(sorted_albums)

# sort by metacritic score
by_meta = summary_meta_score #from summary stats
by_meta = unique(as.character(by_meta$album_name))
#print(summary_meta_score)

# sort the albums by mean energy score
mean_data = taylor_data %>%
  group_by(album_name) %>%
  # create mean_energy column
  mutate(mean_energy = mean(energy)) %>%
  ungroup() %>%
  mutate(
    # order album_names by mean_energy
    album_name = factor(album_name, levels = unique(album_name[order(mean_energy)])),
    # create new columns album_pop and release_order:
    ## order album_pop by meta score (aka by_meta)
    ## order release_order by release date (aka sorted_albums)
    album_pop = factor(album_name, levels = by_meta),
    release_order = factor(album_name, levels = sorted_albums))

# calculate overall mean
overall_mean = mean(mean_data$mean_energy)
# print(overall_mean)

# create a color palette
n_albums = length(unique(mean_data$album_name))
colours = colorRampPalette(c("red","white"))(n_albums)
```

violin plot with TV albums

```
# remove tv albums version
no_tv = mean_data %>%
  filter(!(album_name %in% c("Red (Taylor's Version)","Fearless (Taylor's Version)")))

#colors
n_albums_2 = length(unique(no_tv$album_name))
colours2 = colorRampPalette(c("red","white"))(n_albums_2)

plot1 = ggplot(mean_data, aes(x = energy, y = release_order, fill = album_pop)) +
  geom_violin() +
  xlim(0.0,1.0) +
  # black dots to indicate mean_energy for each album
  stat_summary(fun = mean, geom = "point", shape = 20, color = "black", size = 3) +
  # vertical line for overall_mean
  geom_vline(xintercept = overall_mean, linetype = "solid", color = "blue", size = 0.6) +
  # labels
  labs(title = "Including TV", x = "Energy Score", y = "Album Name by Release Date") +
  # adjust colors and labels
  scale_fill_manual(values = colours) +
  theme(axis.text.y = element_text(size = 12))+
  theme_minimal() +
  theme(legend.position = "none",
        axis.title.y = element_text(size = 8),
        axis.title.x = element_text(size = 8),
        axis.text.y = element_text(size = 5),
        axis.text.x = element_text(size = 6))
```

violin plot without TV albums

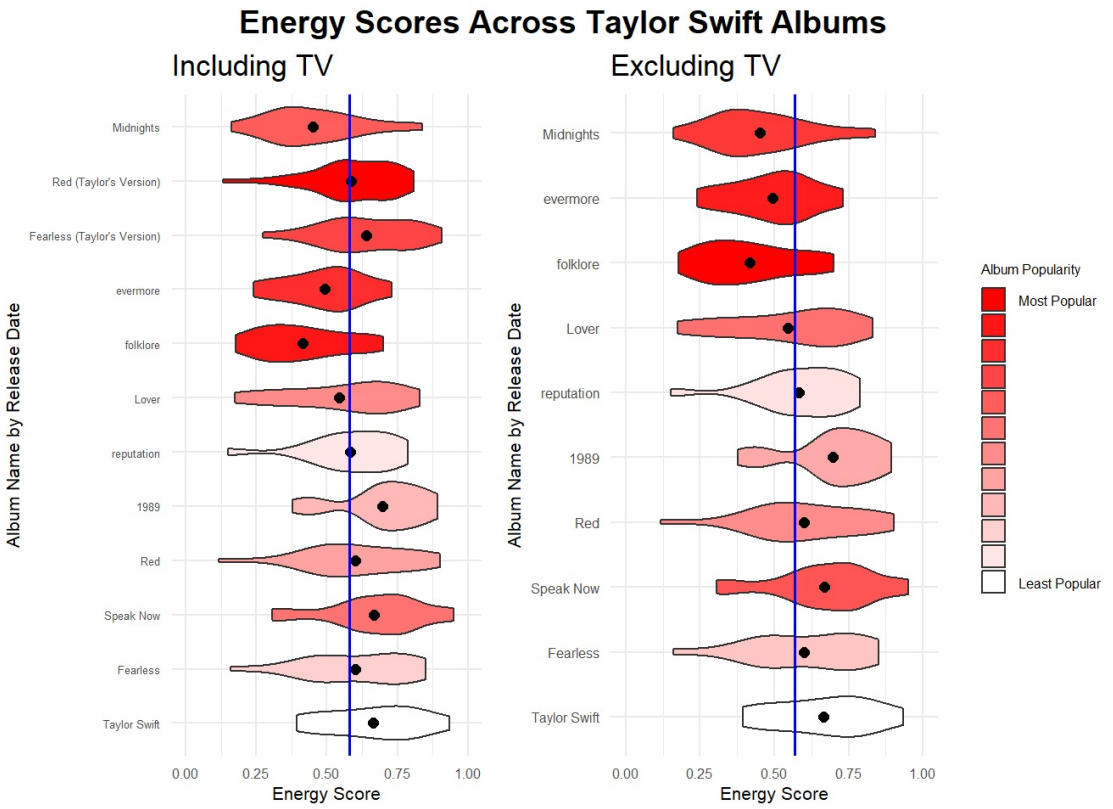
```
#new overall_mean
new_overall_mean = mean(no_tv$mean_energy)

plot2 = ggplot(no_tv, aes(x = energy, y = release_order, fill = album_pop)) +
  geom_violin() +
  xlim(0.0,1.0) +
  # black dots to indicate mean_energy for each album
  stat_summary(fun = mean, geom = "point", shape = 20, color = "black", size = 3) +
  # vertical line for new_overall_mean
  geom_vline(xintercept = new_overall_mean, linetype = "solid", color = "blue", size = 0.6) +
  # labels
  labs(title = "Excluding TV", x = "Energy Score", y = "Album Name by Release Date") +
  # adjust colors + labels
  scale_fill_manual(values = colours2) +
  theme_minimal() +
  theme(legend.position = "none",
        axis.title.y = element_text(size = 8),
        axis.title.x = element_text(size = 8),
        axis.text.y = element_text(size = 6),
        axis.text.x = element_text(size = 6))
```

creating legend

```
# Create a plot to extract the legend
legend_plot = ggplot(mean_data, aes(x = energy, y = release_order, fill = album_pop)) +
  geom_violin() +
  scale_fill_manual(values = colours, labels = c("Most Popular","","","","","","","","","","","Least Popular")) +
  theme_minimal() +
  labs(fill = "Album Popularity") +
  theme(legend.title = element_text(size = 6),
        legend.text = element_text(size = 6),
        legend.key.size = unit(0.4, "cm"),
        legend.spacing.y = unit(0.4, "cm"))
```

final plot

```
# arrange plots side-by-side for comparison
shared_legend = get_legend(legend_plot)

# Arrange plots and legend
final_plot = plot_grid(
  plot1, plot2, shared_legend,
  ncol = 3,
  rel_widths = c(1.1, 1, 0.4)
)

# Add title on top
grid.arrange(
  arrangeGrob(final_plot, top = textGrob("Energy Scores Across Taylor Swift Albums",
                gp = gpar(fontsize = 14, fontface = "bold"))))
```



Discussion

A key aspect of our visualisation lies in the side by side comparison between the graph containing all 12 of Swift's albums, and the graph containing only original releases. In other words, the graph on the right does not include albums that fall under Taylor's Version (TV) which are rerecorded versions of old original albums. We chose to do this to avoid redundancy, as their energy scores are similar to their original versions, leading to skewed trends.

As laid out in the legend of our graph, white represents songs in her debut album 'Taylor Swift', and the red gets deeper as the release dates of her albums become more recent, with 'Midnights' being her most recent release. We arranged the albums according to album release dates along the vertical axis. For each violin shape, a black dot indicates the mean energy score for each album. As previously mentioned, this is helpful for quick identification of central values and the skewness of the distributions.

'1989' has the highest mean energy of 0.697, and the largest width at an energy level of 0.7, suggesting that this album has tracks with high energy. 'folklore' on the other hand has the lowest mean energy of about 0.416, and is a more recent release. Considering the deeper red colours of the 4 violins at the bottom of the graph, this is interesting as we can see that albums with lower energy scores are generally more recently released. This could be due to a shift in Taylor Swift's production preferences as she matures. She may now favour slower and calmer music which amplifies the emotional depth of her lyrics, contrasted with the powerful and high-energy hits from her younger days.

We ended up concluding that it is possible that energy is a factor that contributes to the popularity of Taylor Swift's music. 'folklore' and 'evermore' have the lowest and 3rd lowest mean energy scores respectively, while having the highest popularity according to their metacritic scores. While this indicates that lower energy pop songs draw more listeners, there are other factors that may have affected the popularity of her songs. It is possible that the lower energy levels in her popular albums could have attracted listeners of genres like R&B and ambient music. Additionally, the popularity of 'folklore' and 'evermore' could have been due to the strategic release timing and their surrounding marketing tactics. 'folklore' and 'evermore' were released in 2020 and 2022 respectively, and in 2021, she re-released 2 of her earlier albums. (Hung, n.d.) The renewed interest in her discography could have contributed to the success of her lower energy albums, even though there are stylistic deviations in her music.

Question 3: Does choice of words in her lyrics affect the appeal of Taylor's albums?

Introduction to Question 3

In the 3rd visualisation, we wanted to find out if the choice of words was a factor in the rating of the album. We will compare the most commonly-used words between the top and bottom 3 albums ranked based on the average of the user and metacritic scores. By identifying the commonly-used words, we can analyse their underlying themes and frequency of occurrence, allowing us to determine if there is a specific choice of words that makes her albums more appealing.

Methodology

```
#Find best albums
ranked_albums = taylor_albums_clean %>%
  mutate(rank = min_rank(desc(average_score)))

data("stop_words")
non_words = c("a", "oh", "ooh", "ah", "ahh", "la", "di", "ha", "da", "yeah")
third_vis = lyrics %>%
  anti_join(stop_words, by = "word") %>% #remove common words from lyrics
  filter(!word %in% non_words) %>%
  filter(album_name != "1989 (Taylor's Version)",  # remove these albums because not in tidytuesday data
         album_name != "Speak Now (Taylor's Version)",
         album_name != "THE TORTURED POETS DEPARTMENT",
         album_name != "Beautiful Eyes",
         album_name != "The Taylor Swift Holiday Collection",
         !is.na(album_name)) %>%
  left_join(ranked_albums, by = "album_name") %>%
  filter(rank <= 3 | rank >= 10) %>% # choosing top and bottom 3 albums
  mutate(class = ifelse(rank <= 3, "Top", "Bottom")) %>% #Find top and bottom 3 albums and do analysis on them only
  group_by(class, word) %>%
  summarise(frequency = sum(n)) %>% ##find total count of each word in the different classes
  arrange(desc(frequency), .by_group = TRUE) %>%
  mutate(rank_frequency = row_number()) %>%
  slice_min(rank_frequency, n = 20)  # take only top 10 words of each class
```

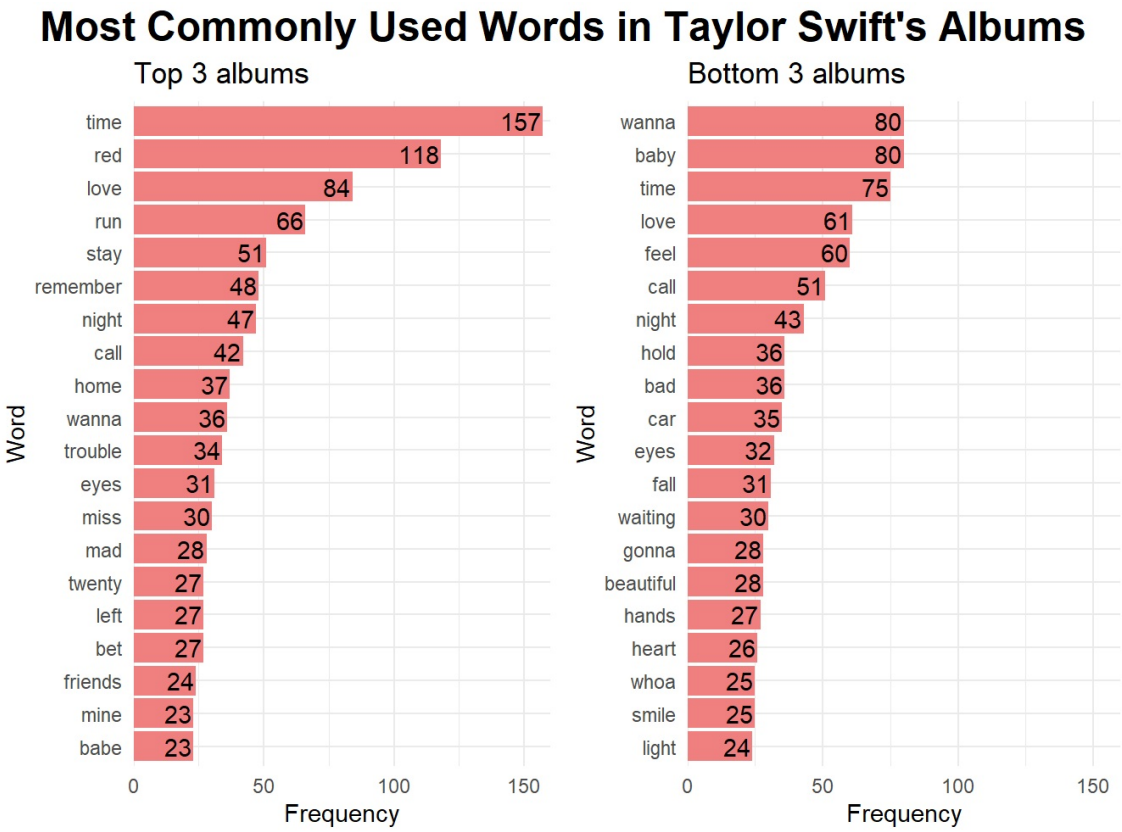
First, we further cleaned the data to remove albums like “1989 (Taylor’s Version)”, “Speak Now (Taylor’s Version)” as they are not found in the orginal tidy tuesday dataset, we also removed non-words like “oh”, “ooh”, “ah”, etc. We removed these words as they are not words that could give us any insight into the appeal of Taylor’s albums.

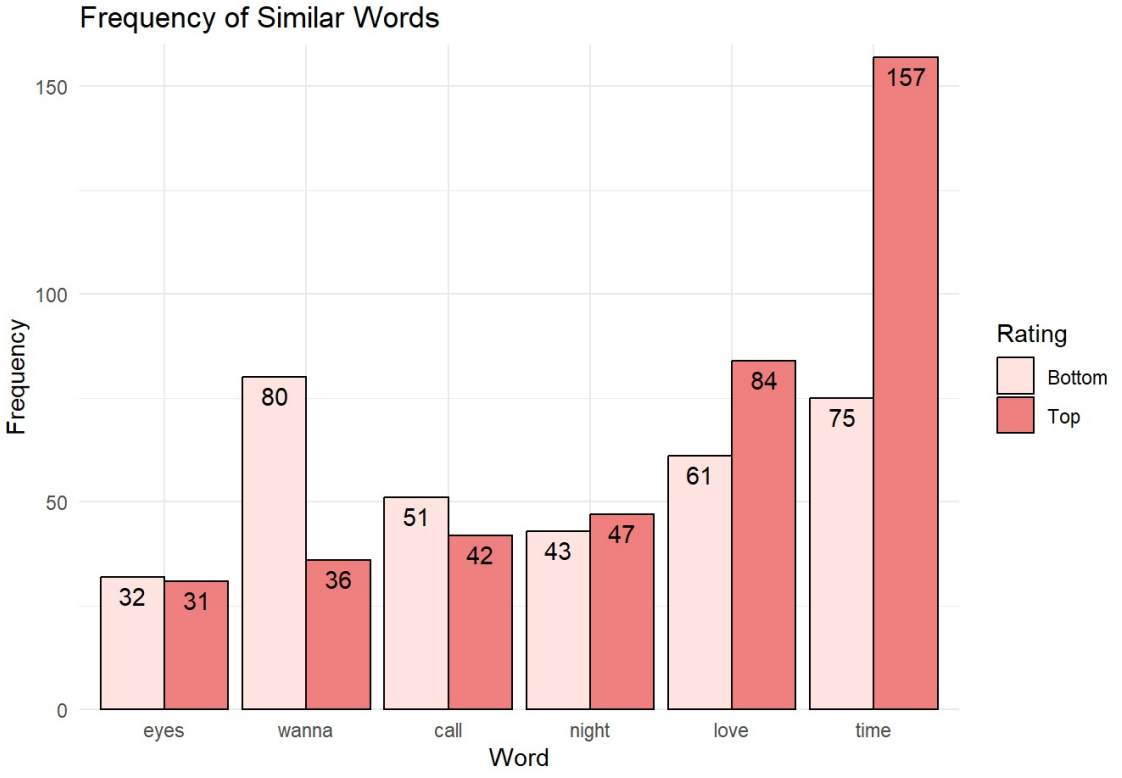
For the visualisation, we first plotted the count of the top 20 words used in the top and bottom 3 rated albums in a side-by-side bar plot. By drawing data from the very top and bottom rated albums, we hope to show a clearer difference when comparing choice of words. Subsequently, at the bottom of the visualisation, we have a vertical bar plot to compare the count of similar between the top and bottom 3 albums.

The bar plots makes it easy to identify the more frequently used words in the top and bottom 3 albums, especially since we only have one categorical variable. The side-by-side bar plots allows for easy comparison between the magnitude of repetition of words between the top and bottom 3 albums. Within the most frequent words for the top and bottom albums, there are a few similar words and to compare the magnitude of count of these words easily, we have another barplot at the bottom of the visualisation.

Visualisation

```
third_vis_top = third_vis %>%
  filter(class == "Top") %>%
  mutate(word = reorder(word, frequency))

third_vis_bottom = third_vis %>%
  filter(class == "Bottom") %>%
  mutate(word = reorder(word, frequency))

top_plot = ggplot(data = third_vis_top, aes(x = word, y = frequency)) +
  geom_col(fill = "lightcoral") +
  coord_flip() +
  geom_text(aes(label = frequency), hjust = "right", nudge_y = -0.7) +
   scale_y_continuous(expand = expansion(mult = c(0,0)),
                      limits = c(0,160),
                      breaks = seq(0,150, by = 50)) +
  labs(title = "Top 3 albums", x = "Word", y = "Frequency") +
  theme(axis.text.y = element_text(size = 12))+
  theme_minimal()

bottom_plot = ggplot(data = third_vis_bottom, aes(x = word, y = frequency)) +
  geom_col(fill = "lightcoral") +
  coord_flip() +
  geom_text(aes(label = frequency), hjust = "right", nudge_y = -0.7) +
   scale_y_continuous(expand = expansion(mult = c(0,0)),
                      limits = c(0,160),
                      breaks = seq(0,150, by = 50)) +
  labs(title = "Bottom 3 albums", x = "Word", y = "Frequency")+
  theme(axis.text.y = element_text(size = 12)) +
  theme_minimal()

third_vis_combined = third_vis_top %>%
  semi_join(third_vis_bottom, by = "word")
bottom_common = third_vis_bottom %>%
  filter(word %in% third_vis_combined$word)
third_vis_combined = third_vis_combined %>%
  rbind(bottom_common)

combined_plot = ggplot(data = third_vis_combined,
                       aes(x = word, y = frequency)) +
  geom_col(aes(fill = class), position  = "dodge", color = "black") +
  # coord_flip() +
  geom_text(aes(label = frequency, group = class),
            color = "black", position = position_dodge(width = 0.9),
            vjust = 1.5) +
   scale_y_continuous(expand = expansion(mult = c(0,0)),
                      limits = c(0,160),
                      breaks = seq(0,150, by = 50)) +
  labs(title = "Frequency of Similar Words", x = "Word", y = "Frequency") +
  scale_fill_manual(values = c("mistyrose", "lightcoral"), name = "Rating") +
  theme(axis.text.x = element_text(size = 30)) +
  theme_minimal()

grid.arrange(top_plot, bottom_plot,combined_plot, nrow = 2,layout_matrix= rbind(c(1,2), 3), top = textGrob("Most Co
mmonly Used Words in Taylor Swift's Albums", gp = gpar(fontsize = 18, fontface = "bold")))
```





Discussion

From the side-by-side bar plots at the top, we found that the top albums have frequently used words like “time” (157 occurrences) and “remember” (48 occurrences), which suggests a nostalgic theme. Whereas for the bottom albums, words like “baby” (80 occurrences) and “wanna” (80 occurrences) suggest a more casual and romantic theme. From the bottom barplot, we see that the word “time” is also commonly used in the bottom albums but with a much smaller frequency (75 occurrences). From these findings, it may be possible to establish that albums with nostalgic themes are more well-liked by Taylor Swift’s audience.

From the side-by-side bar plots, for the most frequent words in top 3 albums, the magnitude of repetition (157, 118, 84) is a lot higher than that of the bottom 3 albums (80,80,75). This could be due to a possible relationship that stronger repetition of lyrics could lead to songs and hence her albums being more well-liked. (Nunes et al., 2014)

Overall conclusion

Based on our results obtained, there are varying factors that could explain Taylor Swift’s tremendous success. From Visualisation 1, it is clear that generally, a greater total number of lyrics and longer duration of her albums has led to the album being received more favorably, excluding the outliers of “folklore” and “evermore” that was discussed earlier. Furthermore, Visualisation 2 shows that her albums with a lower energy score also tends to be

more popular amongst the public. Lastly, Visualisation 3 shows that albums that tend to have more nostalgic themes and repetition of words in the lyrics are more well-received. As such, we believe that the following factors contribute to the popularity and public reception of her albums:

- Increased number of lyrics
- Longer duration of songs
- Lower energy scores
- Themes of nostalgia
- Repetition of words

Teamwork

- Annika Law Jie Yu - Introduction, Conclusion, Summary statistics
- Zhang Changfeng - Visualisation 1
- Beatriz Khoh - Visualisation 2
- Tan Shayne - Visualisation 2
- Hon Mun Dai, Darren - Data cleaning, Visualisation 3

References

Hung, S. (n.d.). Swedishcharts.com - Discography Taylor Swift. https://swedishcharts.com/showinterpret.asp?interpret=Taylor+Swift (https://swedishcharts.com/showinterpret.asp?interpret=Taylor+Swift) Nunes, J. C., Ordanini, A., & Valsesia, F. (2014). The power of repetition: repetitive lyrics in a song increase processing fluency and drive market success. Journal of Consumer Psychology, 25(2), 187–199. https://doi.org/10.1016/j.jcps.2014.12.004 (https://doi.org/10.1016/j.jcps.2014.12.004)
American Songwriter. (2021, January 5). How Taylor Swift's 'folklore' and 'evermore' got us through COVID. American Songwriter. https://americansongwriter.com/how-taylor-swifts-folklore-and-evermore-got-us-through-covid/ (https://americansongwriter.com/how-taylor-swifts-folklore-and-evermore-got-us-through-covid/)
Yahoo Entertainment. (2023, July 24). Taylor Swift celebrates 'folklore' on its 4th anniversary, calls the album a 'return to escapism'. Yahoo Entertainment. https://www.yahoo.com/entertainment/taylor-swift-celebrates-folklore-4th-135628983.html (https://www.yahoo.com/entertainment/taylor-swift-celebrates-folklore-4th-135628983.html)