Business Insight Report

POP MUSIC ANALYSIS BY USING LYRICS

Piya Thavornwong

MSBA: Cohort 5 - Valencia

Piya Thavornwong
MSBA Cohort 5 - Valencia

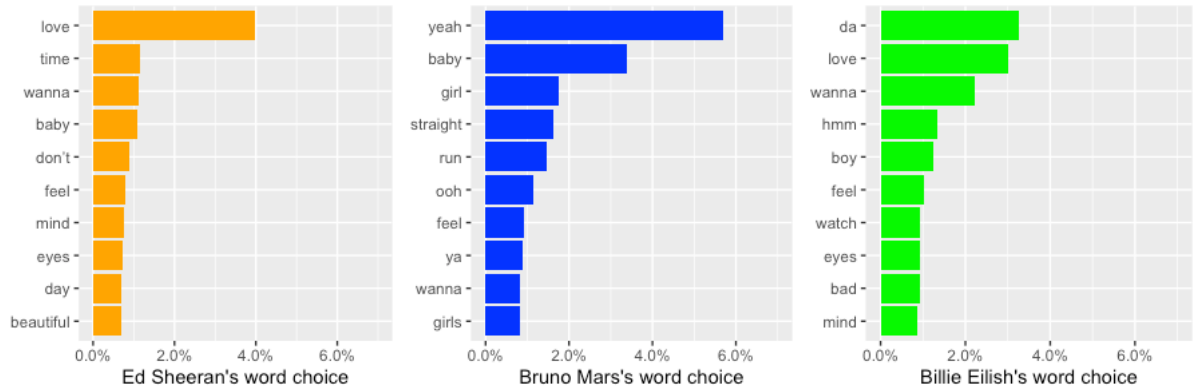
Business Insight Report

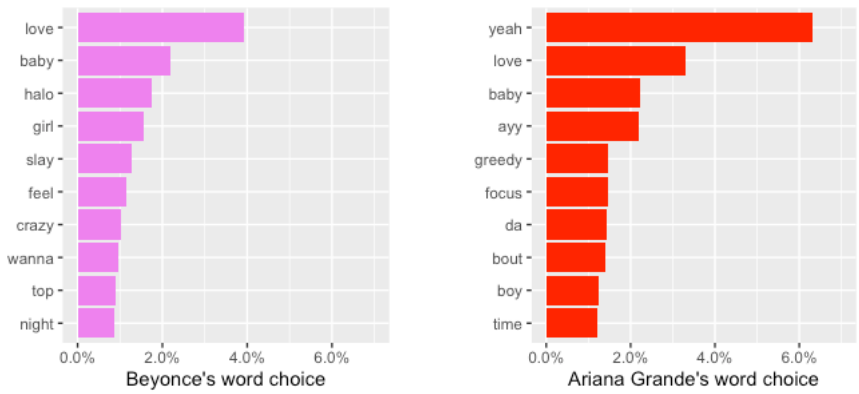
I decided to study some of the most well-known artists, such as Ed Sheeran, Bruno Mars, Billie Eilish, Beyonce, and Ariana Grande, to see what the secret behind their success in the music career is. They are all recognized by people in pop music, which is the most popular genre in music. This business report will analyze words that they put in their songs to see what insights behind it are.

Data collection

The data used in this report is based on lyrics from www.genius.com, which is one of the biggest song lyrics collection. This website shows lines of text, which is unstructured data. In the stage of collecting data, I scraped it by using the "geniusr" package in R (Henderson, 2019) to get lyrics from the URL. Approximately 20-30 songs were selected from each artist to make sure that the outcome will not rely on one specific song. I created a data frame for each artist by binding lyrics from each of their songs together by rows.

After that, I tokenized the data by converting lyrics into only one word for each row. Next, I deleted all stop words from data because these words will not give us insights from it, such as I, you, the, and many more. We can count the words that appear in lyrics to see what are the words that they like to put in their songs. I decided to convert numbers of words into a percentage by dividing it by total words. Because the numbers of words used for each artist are not the same, the percentage form will help us compare each of them.



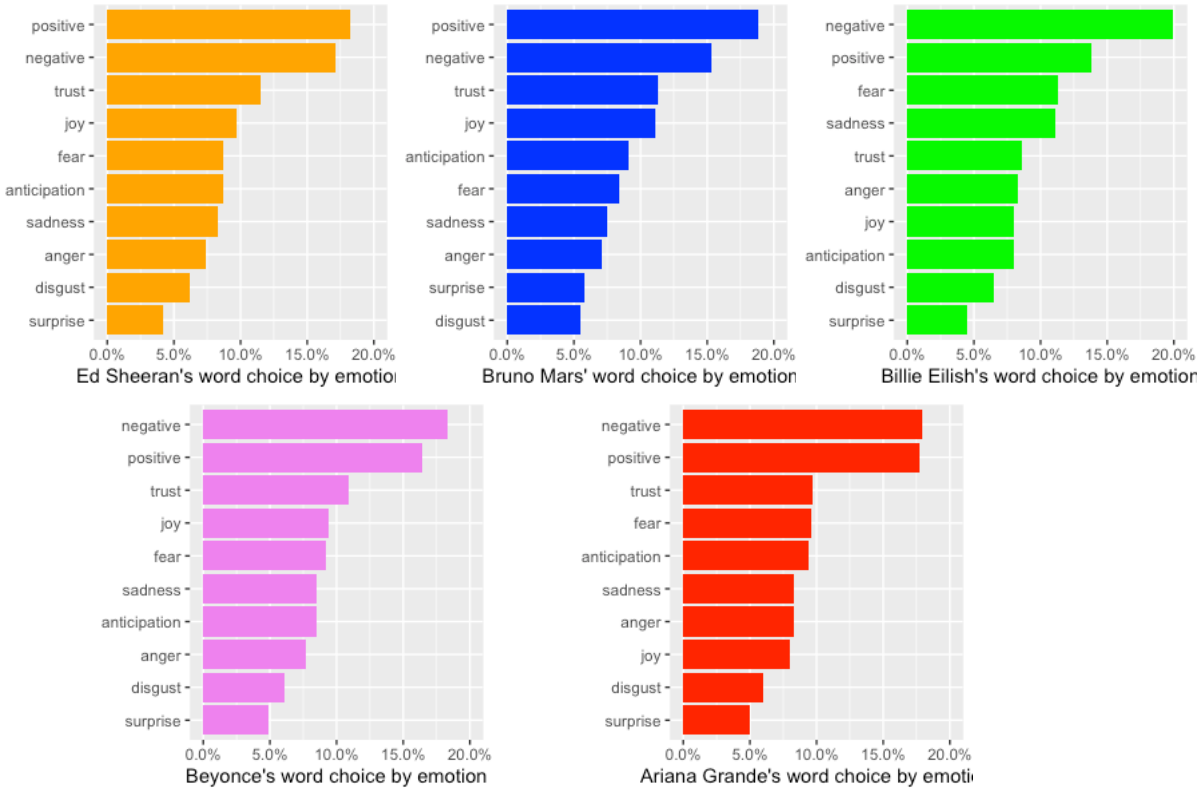


From the bar chart above, we can see that the word “love” appears in the top rank in the most chart. So, we can know that they like to make love songs. Moreover, most of them use the word “baby” very often to make their songs more romantic. One more point that we can see is that Bruno Mars, Billie Eilish, and Ariana Grande like to sing words with no meaning, such as “ooh”, “ya”, “da”, and “ayy”. They put those words with melodies to make their song catchy when people hear it.

Lyric’s emotion analysis

Next, let’s focus on Billie Eilish, 18 years old artist who just won the most Grammy awards in this year (Record of the Year, Album of the Year, Song of the Year, Best new artist, Best pop vocal album). What is the success behind her song? And how she differentiates herself from other artists. We will analyze it by getting feelings of the words in her lyrics.

To get the feeling from the words, I used the dictionary “nrc” to specify the emotion in words and then count it to see the trends of emotion in their lyrics.



From the bar chart above, we can see that emotions in Billie Eilish’s lyrics are different from the other artists. There is a difference in the use of positive and negative words between artists. Ed Sheeran and Bruno Mars like to use positive words more than negative words. Even though Beyonce and Ariana Grande have negative words more than positive words, there is no significant difference between them. On the other hand, Billie Eilish uses negative words (20%) way more than positive words (13%). Moreover, 3rd ranking of emotion for all of them is trust except Billie Eilish. She tends to use words related to fear and sadness. It might be a way to attract more people to follow her. The trends of emotion may be changing from trust and joy to fear and sadness.

Correlogram analysis

In the final step, let's compare the word in lyrics between Billie Eilish and other artists by using a correlogram.

In the Grammy award this year, Ariana Grande and Billie Eilish were nominated in 4 same awards. However, Billie Eilish was the one that won all of them. In this chart, we can compare the difference between word selection in their lyrics. Let's focus on Billie Eilish and Ariana Grande. Words like "bad", "hurt", "bored", and "afraid" tend to appear more in Billie Eilish songs, which reinforce the insight that we got from the previous section that Billie Eilish's lyrics are more in fear and sadness emotion. If we compare her with other artists, we can notice that the word "da" was used by Billie Eilish more than the rest. It is one of the signatures from many of her songs, including "Bad Guy", winner of Grammy awards in Record of the Year, and Song of the Year.

Conclusion

There are four main insights from this analysis that you should consider if you want to become successful in the pop music industry. First, love songs are the most popular topic of songs. Second, try to create some words that can combine with the melody to make the songs catchy. Third, the trends of emotion in lyrics are changing from positive, trust, and joy to negative, fear, and sadness. Forth, try to create your signature by putting some unique words to make people remember. However, there might be other factors such as notes, melodies, and chords that should take into consideration because these factors will affect people's feeling when they listen to music.

Reference

Ewen Henderson. (2019, January 20). Package ‘geniusr’. r-project. Retrieved from https://cran.r-project.org/web/packages/geniusr/geniusr.pdf

Appendix

R Script

```
# Import library
#install.packages('geniusr')
library(geniusr)
library(tidyverse)
library(tidytext)


# Create a list of URL that contains lyrics from Ed Sheeran
sheeran_url <- c("https://genius.com/Ed-sheeran-shape-of-you-lyrics",
            "https://genius.com/Ed-sheeran-perfect-lyrics",
            "https://genius.com/Ed-sheeran-photograph-lyrics",
            "https://genius.com/Ed-sheeran-castle-on-the-hill-lyrics",
            "https://genius.com/Ed-sheeran-happier-lyrics",
            "https://genius.com/Ed-sheeran-galway-girl-lyrics",
            "https://genius.com/Ed-sheeran-thinking-out-loud-lyrics",
            "https://genius.com/Ed-sheeran-dive-lyrics",
            "https://genius.com/Ed-sheeran-the-a-team-lyrics",
            "https://genius.com/Ed-sheeran-beautiful-people-lyrics",
            "https://genius.com/Ed-sheeran-i-see-fire-lyrics",
            "https://genius.com/Ed-sheeran-you-need-me-i-dont-need-you-live-at-the-live-room-
lyrics",
            "https://genius.com/Ed-sheeran-give-me-love-lyrics",
            "https://genius.com/Ed-sheeran-lego-house-lyrics",
            "https://genius.com/Ed-sheeran-sing-lyrics",
            "https://genius.com/Ed-sheeran-bloodstream-lyrics",
            "https://genius.com/Ed-sheeran-dont-lyrics",
            "https://genius.com/Ed-sheeran-supermarket-flowers-lyrics",
            "https://genius.com/Ed-sheeran-south-of-the-border-lyrics",
            "https://genius.com/Ed-sheeran-eraser-lyrics",
            "https://genius.com/Ed-sheeran-beautiful-people-lyrics",
            "https://genius.com/Ed-sheeran-new-man-lyrics",
            "https://genius.com/Ed-sheeran-how-would-you-feel-paean-lyrics",
            "https://genius.com/Ed-sheeran-remember-the-name-lyrics")

# Create a list of URL that contains lyrics from Bruno Mars
mars_url <- c("https://genius.com/Bruno-mars-thats-what-i-like-lyrics",
           "https://genius.com/Bruno-mars-versace-on-the-floor-lyrics",
           "https://genius.com/Bruno-mars-24k-magic-lyrics",
           "https://genius.com/Bruno-mars-when-i-was-your-man-lyrics",
           "https://genius.com/Bruno-mars-finesse-lyrics",
           "https://genius.com/Bruno-mars-just-the-way-you-are-lyrics",
           "https://genius.com/Bruno-mars-grenade-lyrics",
           "https://genius.com/Bruno-mars-locked-out-of-heaven-lyrics",
           "https://genius.com/Bruno-mars-count-on-me-lyrics",
           "https://genius.com/Bruno-mars-treasure-lyrics",
```

```
          "https://genius.com/Bruno-mars-the-lazy-song-lyrics",
          "https://genius.com/Bruno-mars-talking-to-the-moon-lyrics",
          "https://genius.com/Bruno-mars-it-will-rain-lyrics",
          "https://genius.com/Bruno-mars-marry-you-lyrics",
          "https://genius.com/Bruno-mars-young-girls-lyrics",
          "https://genius.com/Bruno-mars-perm-lyrics",
          "https://genius.com/Bruno-mars-calling-all-my-lovelies-lyrics",
          "https://genius.com/Bruno-mars-if-i-knew-lyrics",
          "https://genius.com/Bruno-mars-straight-up-and-down-lyrics",
          "https://genius.com/Bruno-mars-liquor-store-blues-lyrics",
          "https://genius.com/Bruno-mars-rest-of-my-life-lyrics",
          "https://genius.com/Bruno-mars-runaway-baby-lyrics",
          "https://genius.com/Bruno-mars-natalie-lyrics",
          "https://genius.com/Bruno-mars-the-other-side-lyrics")

# Create a list of URL that contains lyrics from Billie Eilish
eilish_url <- c("https://genius.com/Billie-eilish-when-the-partys-over-lyrics",
           "https://genius.com/Billie-eilish-bad-guy-lyrics",
           "https://genius.com/Billie-eilish-everything-i-wanted-lyrics",
           "https://genius.com/Billie-eilish-idontwannabeyouanymore-lyrics",
           "https://genius.com/Billie-eilish-bury-a-friend-lyrics",
           "https://genius.com/Billie-eilish-wish-you-were-gay-lyrics",
           "https://genius.com/Billie-eilish-ocean-eyes-lyrics",
           "https://genius.com/Billie-eilish-i-love-you-lyrics",
           "https://genius.com/Billie-eilish-you-should-see-me-in-a-crown-lyrics",
           "https://genius.com/Billie-eilish-all-the-good-girls-go-to-hell-lyrics",
           "https://genius.com/Billie-eilish-xanny-lyrics",
           "https://genius.com/Billie-eilish-copycat-lyrics",
           "https://genius.com/Billie-eilish-bellyache-lyrics",
           "https://genius.com/Billie-eilish-watch-lyrics",
           "https://genius.com/Billie-eilish-my-strange-addiction-lyrics",
           "https://genius.com/Billie-eilish-my-boy-lyrics",
           "https://genius.com/Billie-eilish-listen-before-i-go-lyrics",
           "https://genius.com/Billie-eilish-hostage-lyrics",
           "https://genius.com/Billie-eilish-six-feet-under-lyrics",
           "https://genius.com/Billie-eilish-ilomilo-lyrics",
           "https://genius.com/Billie-eilish-come-out-and-play-lyrics",
           "https://genius.com/Billie-eilish-bitches-broken-hearts-lyrics",
           "https://genius.com/Billie-eilish-8-lyrics",
           "https://genius.com/Billie-eilish-party-favor-lyrics",
           "https://genius.com/Billie-eilish-61818-lyrics",
           "https://genius.com/Billie-eilish-goodbye-lyrics",
           "https://genius.com/Billie-eilish--lyrics",
           "https://genius.com/Billie-eilish-bored-lyrics",
           "https://genius.com/Billie-eilish-when-i-was-older-lyrics",
           "https://genius.com/Billie-eilish-the-end-of-the-world-lyrics",
           "https://genius.com/Billie-eilish-fingers-crossed-lyrics",
           "https://genius.com/Billie-eilish-shes-broken-lyrics",
           "https://genius.com/Billie-eilish-see-through-lyrics")

# Create a list of URL that contains lyrics from Beyonce
```

```
beyonce_url <- c("https://genius.com/Beyonce-drunk-in-love-lyrics",
                 "https://genius.com/Beyonce-formation-lyrics",
                 "https://genius.com/Beyonce-partition-lyrics",
                 "https://genius.com/Beyonce-mine-lyrics",
                 "https://genius.com/Beyonce-hold-up-lyrics",
                 "https://genius.com/Beyonce-sorry-lyrics",
                 "https://genius.com/Beyonce-if-i-were-a-boy-lyrics",
                 "https://genius.com/Beyonce-pray-you-catch-me-lyrics",
                 "https://genius.com/Beyonce-all-night-lyrics",
                 "https://genius.com/Beyonce-flawless-lyrics",
                 "https://genius.com/Beyonce-halo-lyrics",
                 "https://genius.com/Beyonce-dont-hurt-yourself-lyrics",
                 "https://genius.com/Beyonce-listen-lyrics",
                 "https://genius.com/Beyonce-crazy-in-love-lyrics",
                 "https://genius.com/Beyonce-love-on-top-lyrics",
                 "https://genius.com/Beyonce-freedom-lyrics",
                 "https://genius.com/Beyonce-blow-lyrics",
                 "https://genius.com/Beyonce-rocket-lyrics",
                 "https://genius.com/Beyonce-sandcastles-lyrics",
                 "https://genius.com/Beyonce-7-11-lyrics",
                 "https://genius.com/Beyonce-6-inch-lyrics",
                 "https://genius.com/Beyonce-love-drought-lyrics",
                 "https://genius.com/Beyonce-pretty-hurts-lyrics")

# Create a list of URL that contains lyrics from Ariana Grande
ariana_url <- c("https://genius.com/Ariana-grande-thank-u-next-lyrics",
                "https://genius.com/Ariana-grande-7-rings-lyrics",
                "https://genius.com/Ariana-grande-god-is-a-woman-lyrics",
                "https://genius.com/Ariana-grande-side-to-side-lyrics",
                "https://genius.com/Ariana-grande-no-tears-left-to-cry-lyrics",
                "https://genius.com/Ariana-grande-breathin-lyrics",
                "https://genius.com/Ariana-grande-break-up-with-your-girlfriend-im-bored-lyrics",
                "https://genius.com/Ariana-grande-imagine-lyrics",
                "https://genius.com/Ariana-grande-needy-lyrics",
                "https://genius.com/Ariana-grande-into-you-lyrics",
                "https://genius.com/Ariana-grande-dangerous-woman-lyrics",
                "https://genius.com/Ariana-grande-one-last-time-lyrics",
                "https://genius.com/Ariana-grande-the-way-lyrics",
                "https://genius.com/Ariana-grande-problem-lyrics",
                "https://genius.com/Ariana-grande-focus-lyrics",
                "https://genius.com/Ariana-grande-almost-is-never-enough-lyrics",
                "https://genius.com/Ariana-grande-break-free-lyrics",
                "https://genius.com/Ariana-grande-greedy-lyrics",
                "https://genius.com/Ariana-grande-ghostin-lyrics",
                "https://genius.com/Ariana-grande-sweetener-lyrics",
                "https://genius.com/Ariana-grande-nasa-lyrics",
                "https://genius.com/Ariana-grande-rem-lyrics",
                "https://genius.com/Ariana-grande-in-my-head-lyrics",
                "https://genius.com/Ariana-grande-the-light-is-coming-lyrics",
                "https://genius.com/Ariana-grande-goodnight-n-go-lyrics",
                "https://genius.com/Ariana-grande-fake-smile-lyrics",
```

```
            "https://genius.com/Ariana-grande-bloodline-lyrics")


# Scraping all lyrics from Ed Sheeran by using geniusr
datalist = list()
for (i in 1: length(sheeran_url)) {
  dat <- scrape_lyrics_url(song_lyrics_url = sheeran_url[i])
  datalist[[i]] <- dat # add it to your list
}
sheeran_df = do.call(rbind, datalist)

# Scraping all lyrics from Bruno Mars geniusr
datalist = list()
for (i in 1: length(mars_url)) {
  dat <- scrape_lyrics_url(song_lyrics_url = mars_url[i])
  datalist[[i]] <- dat # add it to your list
}
mars_df = do.call(rbind, datalist)

# Scraping all lyrics from Billie Eilish
datalist = list()
for (i in 1: length(eilish_url)) {
  dat <- scrape_lyrics_url(song_lyrics_url = eilish_url[i])
  datalist[[i]] <- dat # add it to your list
}
eilish_df = do.call(rbind, datalist)

# Scraping all lyrics from Beyonce geniusr
datalist = list()
for (i in 1: length(beyonce_url)) {
  dat <- scrape_lyrics_url(song_lyrics_url = beyonce_url[i])
  datalist[[i]] <- dat # add it to your list
}
beyonce_df = do.call(rbind, datalist)

# Scraping all lyrics from Ariana Grande
datalist = list()
for (i in 1: length(ariana_url)) {
  dat <- scrape_lyrics_url(song_lyrics_url = ariana_url[i])
  datalist[[i]] <- dat # add it to your list
}
ariana_df = do.call(rbind, datalist)


# Tokenize lyrics from Ed Sheeran
sheeran_struc <- sheeran_df %>%
  unnest_tokens(word,line) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
print(sheeran_struc)
```

```
# Tokenize lyrics from Bruna Mars
mars_struc <- mars_df %>%
  unnest_tokens(word,line) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
print(mars_struc)

# Tokenize lyrics from Billie Eilish
eilish_struc <- eilish_df %>%
  unnest_tokens(word,line) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
print(eilish_struc)

# Tokenize lyrics from Beyonce
beyonce_struc <- beyonce_df %>%
  unnest_tokens(word,line) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
print(beyonce_struc)

# Tokenize lyrics from Ariana Grande
ariana_struc <- ariana_df %>%
  unnest_tokens(word,line) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
print(ariana_struc)


# Conver to percentage
freq_sheeran_word <- sheeran_struc %>%
                mutate(proportion = n/sum(n))

# Conver to percentage
freq_mars_word <- mars_struc %>%
                mutate(proportion = n/sum(n))

# Conver to percentage
freq_eilish_word <- eilish_struc %>%
                mutate(proportion = n/sum(n))

# Conver to percentage
freq_beyonce_word <- beyonce_struc %>%
                mutate(proportion = n/sum(n))

# Conver to percentage
freq_ariana_word <- ariana_struc %>%
                mutate(proportion = n/sum(n))

# Visualize the word's frequency in bar chart
```

```
freq_sheeran_word %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, proportion) , proportion))+
  geom_col(show.legend = FALSE, fill = 'orange', alpha = 1)+
  scale_fill_brewer(direction = -1, palette = "Blues")+
  scale_y_continuous(limits = c(0,0.07), labels = scales::percent) +
  labs(y="Ed Sheeran's word choice", x=NULL)+
  coord_flip()

# Visualize the word's frequency in bar chart
freq_mars_word %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, proportion) , proportion))+
  geom_col(show.legend = FALSE, fill = 'blue', alpha = 1)+
  scale_y_continuous(limit = c(0,0.07), labels = scales::percent) +
  labs(y="Bruno Mars's word choice", x=NULL)+
  coord_flip()

# Visualize the word's frequency in bar chart
freq_eilish_word %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, proportion) , proportion))+
  geom_col(show.legend = FALSE, fill = 'green', alpha = 1)+
  scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
  labs(y="Billie Eilish's word choice", x=NULL)+
  coord_flip()

# Visualize the word's frequency in bar chart
freq_beyonce_word %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, proportion) , proportion))+
  geom_col(show.legend = FALSE, fill = 'violet', alpha = 1)+
  scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
  labs(y="Beyonce's word choice", x=NULL)+
  coord_flip()

# Visualize the word's frequency in bar chart
freq_ariana_word %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, proportion) , proportion))+
  geom_col(show.legend = FALSE, fill = 'red', alpha = 1)+
  scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
  labs(y="Ariana Grande's word choice", x=NULL)+
  coord_flip()



# Inner join with nrc dictionary and create a proportion column
sheeran_feeling <- sheeran_struc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(sentiment, sort = TRUE) %>%
```

```
  mutate(proportion = n/sum(n))
print(sheeran_feeling)

# Inner join with nrc dictionary and create a proportion column
mars_feeling <- mars_struc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(sentiment, sort = TRUE) %>%
  mutate(proportion = n/sum(n))
print(mars_feeling)

# Inner join with nrc dictionary and create a proportion column
eilish_feeling <- eilish_struc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(sentiment, sort = TRUE) %>%
  mutate(proportion = n/sum(n))
print(eilish_feeling)

# Inner join with nrc dictionary and create a proportion column
beyonce_feeling <- beyonce_struc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(sentiment, sort = TRUE) %>%
  mutate(proportion = n/sum(n))
print(beyonce_feeling)

# Inner join with nrc dictionary and create a proportion column
ariana_feeling <- ariana_struc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(sentiment, sort = TRUE) %>%
  mutate(proportion = n/sum(n))
print(ariana_feeling)


# Plotting bar chart
sheeran_feeling %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(sentiment=reorder(sentiment, proportion)) %>%
  ggplot(aes(sentiment , proportion))+
  geom_col(show.legend = FALSE, fill = 'orange', alpha = 1)+
  scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
  labs(y="Ed Sheeran's word choice by emotion", x=NULL)+
  coord_flip()

# Plotting bar chart
mars_feeling %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(sentiment=reorder(sentiment, proportion)) %>%
  ggplot(aes(sentiment , proportion))+
  geom_col(show.legend = FALSE, fill = 'blue', alpha = 1)+
  scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
```

```
  labs(y="Bruno Mars' word choice by emotion", x=NULL)+
  coord_flip()

# Plotting bar chart
eilish_feeling %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(sentiment=reorder(sentiment, proportion)) %>%
  ggplot(aes(sentiment , proportion))+
  geom_col(show.legend = FALSE, fill = 'green', alpha = 1)+
  scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
  labs(y="Billie Eilish's word choice by emotion", x=NULL)+
  coord_flip()

# Plotting bar chart
beyonce_feeling %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(sentiment=reorder(sentiment, proportion)) %>%
  ggplot(aes(sentiment , proportion))+
  geom_col(show.legend = FALSE, fill = 'violet', alpha = 1)+
  scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
  labs(y="Beyonce's word choice by emotion", x=NULL)+
  coord_flip()

# Plotting bar chart
ariana_feeling %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(sentiment=reorder(sentiment, proportion)) %>%
  ggplot(aes(sentiment , proportion))+
  geom_col(show.legend = FALSE, fill = 'red', alpha = 1)+
  scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
  labs(y="Ariana Grande's word choice by emotion", x=NULL)+
  coord_flip()


# Binding all artists by rows and calculate the probability that each words appear
frequency <- bind_rows(mutate(sheeran_struc, Artist="Ed Sheeran"),
                       mutate(mars_struc, Artist= "Bruno Mars"),
                       mutate(eilish_struc, Artist="Billie Eilish"),
                       mutate(beyonce_struc, Artist = 'Beyonce'),
                       mutate(ariana_struc, Artist='Ariana Grande')) %>%
  group_by(Artist) %>%
  mutate(proportion = n/sum(n))%>%
  select(-n) %>%
  spread(Artist, proportion) %>%
  gather(Artist, proportion, `Ed Sheeran`, `Bruno Mars`, `Beyonce`, `Ariana Grande`)


# Create a correlogram
```

```
ggplot(frequency, aes(x=proportion, y=`Billie Eilish`,
                      color = abs(`Billie Eilish`- proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word),check_overlap = TRUE, vjust=0.15) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  facet_wrap(~Artist, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "Billie Eilish", x=NULL)
```

R Output

```
> # Import library
> #install.packages('geniusr')
> library(geniusr)
> library(tidyverse)
> library(tidytext)
>
>
> # Create a list of URL that contains lyrics from Ed Sheeran
> sheeran_url <- c("https://genius.com/Ed-sheeran-shape-of-you-lyrics",
+                  "https://genius.com/Ed-sheeran-perfect-lyrics",
+                  "https://genius.com/Ed-sheeran-photograph-lyrics",
+                  "https://genius.com/Ed-sheeran-castle-on-the-hill-lyrics",
+                  "https://genius.com/Ed-sheeran-happier-lyrics",
+                  "https://genius.com/Ed-sheeran-galway-girl-lyrics",
+                  "https://genius.com/Ed-sheeran-thinking-out-loud-lyrics",
+                  "https://genius.com/Ed-sheeran-dive-lyrics",
+                  "https://genius.com/Ed-sheeran-the-a-team-lyrics",
+                  "https://genius.com/Ed-sheeran-beautiful-people-lyrics",
+                  "https://genius.com/Ed-sheeran-i-see-fire-lyrics",
+                       "https://genius.com/Ed-sheeran-you-need-me-i-dont-need-you-live-at-the-live-room-
lyrics",
+                  "https://genius.com/Ed-sheeran-give-me-love-lyrics",
+                  "https://genius.com/Ed-sheeran-lego-house-lyrics",
+                  "https://genius.com/Ed-sheeran-sing-lyrics",
+                  "https://genius.com/Ed-sheeran-bloodstream-lyrics",
+                  "https://genius.com/Ed-sheeran-dont-lyrics",
+                  "https://genius.com/Ed-sheeran-supermarket-flowers-lyrics",
+                  "https://genius.com/Ed-sheeran-south-of-the-border-lyrics",
+                  "https://genius.com/Ed-sheeran-eraser-lyrics",
+                  "https://genius.com/Ed-sheeran-beautiful-people-lyrics",
+                  "https://genius.com/Ed-sheeran-new-man-lyrics",
+                  "https://genius.com/Ed-sheeran-how-would-you-feel-paean-lyrics",
+                  "https://genius.com/Ed-sheeran-remember-the-name-lyrics")
>
> # Create a list of URL that contains lyrics from Bruno Mars
> mars_url <- c("https://genius.com/Bruno-mars-thats-what-i-like-lyrics",
```

```
+           "https://genius.com/Bruno-mars-versace-on-the-floor-lyrics",
+           "https://genius.com/Bruno-mars-24k-magic-lyrics",
+           "https://genius.com/Bruno-mars-when-i-was-your-man-lyrics",
+           "https://genius.com/Bruno-mars-finesse-lyrics",
+           "https://genius.com/Bruno-mars-just-the-way-you-are-lyrics",
+           "https://genius.com/Bruno-mars-grenade-lyrics",
+           "https://genius.com/Bruno-mars-locked-out-of-heaven-lyrics",
+           "https://genius.com/Bruno-mars-count-on-me-lyrics",
+           "https://genius.com/Bruno-mars-treasure-lyrics",
+           "https://genius.com/Bruno-mars-the-lazy-song-lyrics",
+           "https://genius.com/Bruno-mars-talking-to-the-moon-lyrics",
+           "https://genius.com/Bruno-mars-it-will-rain-lyrics",
+           "https://genius.com/Bruno-mars-marry-you-lyrics",
+           "https://genius.com/Bruno-mars-young-girls-lyrics",
+           "https://genius.com/Bruno-mars-perm-lyrics",
+           "https://genius.com/Bruno-mars-calling-all-my-lovelies-lyrics",
+           "https://genius.com/Bruno-mars-if-i-knew-lyrics",
+           "https://genius.com/Bruno-mars-straight-up-and-down-lyrics",
+           "https://genius.com/Bruno-mars-liquor-store-blues-lyrics",
+           "https://genius.com/Bruno-mars-rest-of-my-life-lyrics",
+           "https://genius.com/Bruno-mars-runaway-baby-lyrics",
+           "https://genius.com/Bruno-mars-natalie-lyrics",
+           "https://genius.com/Bruno-mars-the-other-side-lyrics")
>
> # Create a list of URL that contains lyrics from Billie Eilish
> eilish_url <- c("https://genius.com/Billie-eilish-when-the-partys-over-lyrics",
+             "https://genius.com/Billie-eilish-bad-guy-lyrics",
+             "https://genius.com/Billie-eilish-everything-i-wanted-lyrics",
+             "https://genius.com/Billie-eilish-idontwannabeyouanymore-lyrics",
+             "https://genius.com/Billie-eilish-bury-a-friend-lyrics",
+             "https://genius.com/Billie-eilish-wish-you-were-gay-lyrics",
+             "https://genius.com/Billie-eilish-ocean-eyes-lyrics",
+             "https://genius.com/Billie-eilish-i-love-you-lyrics",
+             "https://genius.com/Billie-eilish-you-should-see-me-in-a-crown-lyrics",
+             "https://genius.com/Billie-eilish-all-the-good-girls-go-to-hell-lyrics",
+             "https://genius.com/Billie-eilish-xanny-lyrics",
+             "https://genius.com/Billie-eilish-copycat-lyrics",
+             "https://genius.com/Billie-eilish-bellyache-lyrics",
+             "https://genius.com/Billie-eilish-watch-lyrics",
+             "https://genius.com/Billie-eilish-my-strange-addiction-lyrics",
+             "https://genius.com/Billie-eilish-my-boy-lyrics",
+             "https://genius.com/Billie-eilish-listen-before-i-go-lyrics",
+             "https://genius.com/Billie-eilish-hostage-lyrics",
+             "https://genius.com/Billie-eilish-six-feet-under-lyrics",
+             "https://genius.com/Billie-eilish-ilomilo-lyrics",
+             "https://genius.com/Billie-eilish-come-out-and-play-lyrics",
+             "https://genius.com/Billie-eilish-bitches-broken-hearts-lyrics",
+             "https://genius.com/Billie-eilish-8-lyrics",
+             "https://genius.com/Billie-eilish-party-favor-lyrics",
+             "https://genius.com/Billie-eilish-61818-lyrics",
+             "https://genius.com/Billie-eilish-goodbye-lyrics",
```

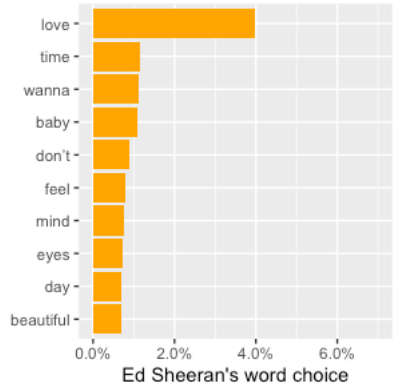
```
+              "https://genius.com/Billie-eilish--lyrics",
+              "https://genius.com/Billie-eilish-bored-lyrics",
+              "https://genius.com/Billie-eilish-when-i-was-older-lyrics",
+              "https://genius.com/Billie-eilish-the-end-of-the-world-lyrics",
+              "https://genius.com/Billie-eilish-fingers-crossed-lyrics",
+              "https://genius.com/Billie-eilish-shes-broken-lyrics",
+              "https://genius.com/Billie-eilish-see-through-lyrics")
>
> # Create a list of URL that contains lyrics from Beyonce
> beyonce_url <- c("https://genius.com/Beyonce-drunk-in-love-lyrics",
+              "https://genius.com/Beyonce-formation-lyrics",
+              "https://genius.com/Beyonce-partition-lyrics",
+              "https://genius.com/Beyonce-mine-lyrics",
+              "https://genius.com/Beyonce-hold-up-lyrics",
+              "https://genius.com/Beyonce-sorry-lyrics",
+              "https://genius.com/Beyonce-if-i-were-a-boy-lyrics",
+              "https://genius.com/Beyonce-pray-you-catch-me-lyrics",
+              "https://genius.com/Beyonce-all-night-lyrics",
+              "https://genius.com/Beyonce-flawless-lyrics",
+              "https://genius.com/Beyonce-halo-lyrics",
+              "https://genius.com/Beyonce-dont-hurt-yourself-lyrics",
+              "https://genius.com/Beyonce-listen-lyrics",
+              "https://genius.com/Beyonce-crazy-in-love-lyrics",
+              "https://genius.com/Beyonce-love-on-top-lyrics",
+              "https://genius.com/Beyonce-freedom-lyrics",
+              "https://genius.com/Beyonce-blow-lyrics",
+              "https://genius.com/Beyonce-rocket-lyrics",
+              "https://genius.com/Beyonce-sandcastles-lyrics",
+              "https://genius.com/Beyonce-7-11-lyrics",
+              "https://genius.com/Beyonce-6-inch-lyrics",
+              "https://genius.com/Beyonce-love-drought-lyrics",
+              "https://genius.com/Beyonce-pretty-hurts-lyrics")
>
> # Create a list of URL that contains lyrics from Ariana Grande
> ariana_url <- c("https://genius.com/Ariana-grande-thank-u-next-lyrics",
+             "https://genius.com/Ariana-grande-7-rings-lyrics",
+             "https://genius.com/Ariana-grande-god-is-a-woman-lyrics",
+             "https://genius.com/Ariana-grande-side-to-side-lyrics",
+             "https://genius.com/Ariana-grande-no-tears-left-to-cry-lyrics",
+             "https://genius.com/Ariana-grande-breathin-lyrics",
+             "https://genius.com/Ariana-grande-break-up-with-your-girlfriend-im-bored-lyrics",
+             "https://genius.com/Ariana-grande-imagine-lyrics",
+             "https://genius.com/Ariana-grande-needy-lyrics",
+             "https://genius.com/Ariana-grande-into-you-lyrics",
+             "https://genius.com/Ariana-grande-dangerous-woman-lyrics",
+             "https://genius.com/Ariana-grande-one-last-time-lyrics",
+             "https://genius.com/Ariana-grande-the-way-lyrics",
+             "https://genius.com/Ariana-grande-problem-lyrics",
+             "https://genius.com/Ariana-grande-focus-lyrics",
+             "https://genius.com/Ariana-grande-almost-is-never-enough-lyrics",
+             "https://genius.com/Ariana-grande-break-free-lyrics",
```

```
+            "https://genius.com/Ariana-grande-greedy-lyrics",
+            "https://genius.com/Ariana-grande-ghostin-lyrics",
+            "https://genius.com/Ariana-grande-sweetener-lyrics",
+            "https://genius.com/Ariana-grande-nasa-lyrics",
+            "https://genius.com/Ariana-grande-rem-lyrics",
+            "https://genius.com/Ariana-grande-in-my-head-lyrics",
+            "https://genius.com/Ariana-grande-the-light-is-coming-lyrics",
+            "https://genius.com/Ariana-grande-goodnight-n-go-lyrics",
+            "https://genius.com/Ariana-grande-fake-smile-lyrics",
+            "https://genius.com/Ariana-grande-bloodline-lyrics")
>
>
>
> # Scraping all lyrics from Ed Sheeran by using geniusr
> datalist = list()
> for (i in 1: length(sheeran_url)) {
+   dat <- scrape_lyrics_url(song_lyrics_url = sheeran_url[i])
+   datalist[[i]] <- dat # add it to your list
+ }
> sheeran_df = do.call(rbind, datalist)
>
> # Scraping all lyrics from Bruno Mars geniusr
> datalist = list()
> for (i in 1: length(mars_url)) {
+   dat <- scrape_lyrics_url(song_lyrics_url = mars_url[i])
+   datalist[[i]] <- dat # add it to your list
+ }
> mars_df = do.call(rbind, datalist)
>
> # Scraping all lyrics from Billie Eilish
> datalist = list()
> for (i in 1: length(eilish_url)) {
+   dat <- scrape_lyrics_url(song_lyrics_url = eilish_url[i])
+   datalist[[i]] <- dat # add it to your list
+ }
> eilish_df = do.call(rbind, datalist)
>
> # Scraping all lyrics from Beyonce geniusr
> datalist = list()
> for (i in 1: length(beyonce_url)) {
+   dat <- scrape_lyrics_url(song_lyrics_url = beyonce_url[i])
+   datalist[[i]] <- dat # add it to your list
+ }
> beyonce_df = do.call(rbind, datalist)
>
> # Scraping all lyrics from Ariana Grande
> datalist = list()
> for (i in 1: length(ariana_url)) {
+   dat <- scrape_lyrics_url(song_lyrics_url = ariana_url[i])
+   datalist[[i]] <- dat # add it to your list
+ }
```

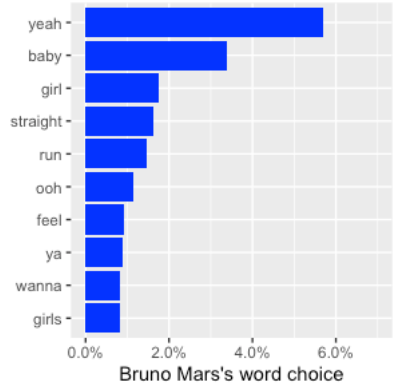
```
> ariana_df = do.call(rbind, datalist)
>
>
> # Tokenize lyrics from Ed Sheeran
> sheeran_struc <- sheeran_df %>%
+   unnest_tokens(word,line) %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
> print(sheeran_struc)
# A tibble: 1,319 x 2
   word         n
   <chr>    <int>
 1 love       143
 2 time        42
 3 wanna       40
 4 baby        39
 5 don’t       32
 6 feel        29
 7 mind        27
 8 eyes        26
 9 beautiful   25
10 day         25
# … with 1,309 more rows
>
> # Tokenize lyrics from Bruna Mars
> mars_struc <- mars_df %>%
+   unnest_tokens(word,line) %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
> print(mars_struc)
# A tibble: 726 x 2
   word        n
   <chr>   <int>
 1 yeah      143
 2 baby       85
 3 girl       44
 4 straight   41
 5 run        37
 6 ooh        29
 7 feel       23
 8 ya         22
 9 girls      21
10 wanna      21
# … with 716 more rows
>
> # Tokenize lyrics from Billie Eilish
> eilish_struc <- eilish_df %>%
+   unnest_tokens(word,line) %>%
+   anti_join(stop_words) %>%
```

```
+   count(word, sort = TRUE)
Joining, by = "word"
> print(eilish_struc)
# A tibble: 792 x 2
  word      n
  <chr> <int>
 1 da       71
 2 love     65
 3 wanna    48
 4 hmm      29
 5 boy      27
 6 feel     22
 7 bad      20
 8 eyes     20
 9 watch    20
10 mind     19
# … with 782 more rows
>
> # Tokenize lyrics from Beyonce
> beyonce_struc <- beyonce_df %>%
+   unnest_tokens(word,line) %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
> print(beyonce_struc)
# A tibble: 1,188 x 2
  word      n
  <chr> <int>
 1 love    152
 2 baby     84
 3 halo     67
 4 girl     60
 5 slay     49
 6 feel     44
 7 crazy    39
 8 wanna    37
 9 top      34
10 night    33
# … with 1,178 more rows
>
> # Tokenize lyrics from Ariana Grande
> ariana_struc <- ariana_df %>%
+   unnest_tokens(word,line) %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
> print(ariana_struc)
# A tibble: 812 x 2
  word       n
  <chr>  <int>
 1 yeah     236
```

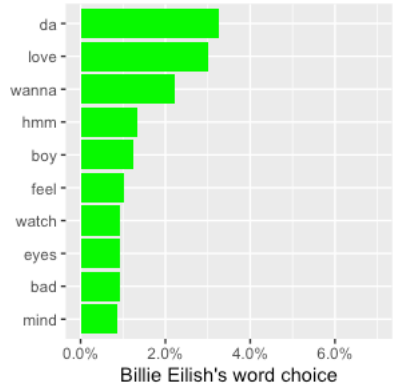
```
 2 love     124
 3 baby      83
 4 ayy       82
 5 focus     55
 6 greedy    55
 7 da        54
 8 bout      53
 9 boy       46
10 time      45
# … with 802 more rows
>
>
> # Conver to percentage
> freq_sheeran_word <- sheeran_struc %>%
+                   mutate(proportion = n/sum(n))
>
> # Conver to percentage
> freq_mars_word <- mars_struc %>%
+                   mutate(proportion = n/sum(n))
>
> # Conver to percentage
> freq_eilish_word <- eilish_struc %>%
+                   mutate(proportion = n/sum(n))
>
> # Conver to percentage
> freq_beyonce_word <- beyonce_struc %>%
+                   mutate(proportion = n/sum(n))
>
> # Conver to percentage
> freq_ariana_word <- ariana_struc %>%
+                   mutate(proportion = n/sum(n))
>
> # Visualize the word's frequency in bar chart
> freq_sheeran_word %>%
+   top_n(10) %>%
+   ggplot(aes(reorder(word, proportion) , proportion))+
+   geom_col(show.legend = FALSE, fill = 'orange', alpha = 1)+
+   scale_fill_brewer(direction = -1, palette = "Blues")+
+   scale_y_continuous(limits = c(0,0.07), labels = scales::percent) +
+   labs(y="Ed Sheeran's word choice", x=NULL)+
+   coord_flip()
Selecting by proportion
```



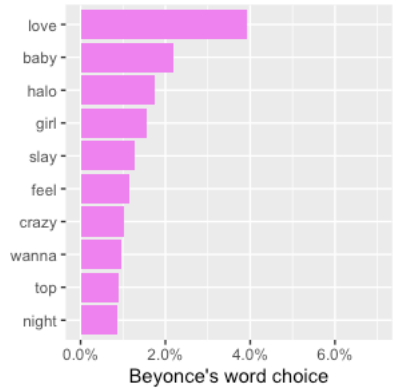

```
>
> # Visualize the word's frequency in bar chart
> freq_mars_word %>%
+   top_n(10) %>%
+   ggplot(aes(reorder(word, proportion) , proportion))+
+   geom_col(show.legend = FALSE, fill = 'blue', alpha = 1)+
+   scale_y_continuous(limit = c(0,0.07), labels = scales::percent) +
+   labs(y="Bruno Mars's word choice", x=NULL)+
+   coord_flip()
Selecting by proportion
```



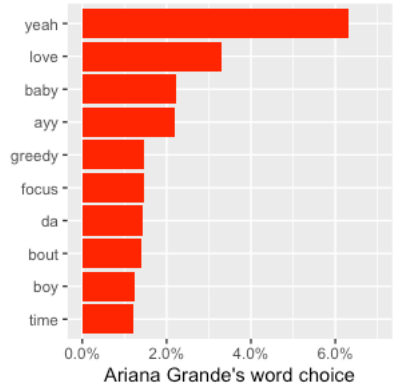

```
>
> # Visualize the word's frequency in bar chart
> freq_eilish_word %>%
+   top_n(10) %>%
+   ggplot(aes(reorder(word, proportion) , proportion))+
+   geom_col(show.legend = FALSE, fill = 'green', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
+   labs(y="Billie Eilish's word choice", x=NULL)+
+   coord_flip()
Selecting by proportion
```




```
>
> # Visualize the word's frequency in bar chart
> freq_beyonce_word %>%
+   top_n(10) %>%
+   ggplot(aes(reorder(word, proportion) , proportion))+
+   geom_col(show.legend = FALSE, fill = 'violet', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
+   labs(y="Beyonce's word choice", x=NULL)+
+   coord_flip()
Selecting by proportion
```



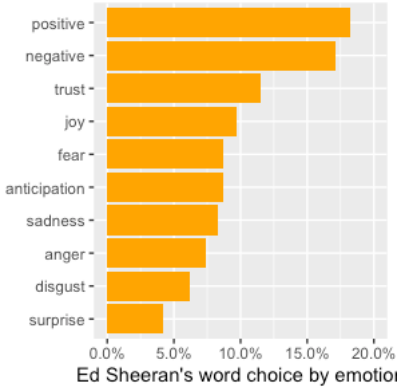

```
>
> # Visualize the word's frequency in bar chart
> freq_ariana_word %>%
+   top_n(10) %>%
+   ggplot(aes(reorder(word, proportion) , proportion))+
+   geom_col(show.legend = FALSE, fill = 'red', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.07),labels = scales::percent) +
+   labs(y="Ariana Grande's word choice", x=NULL)+
+   coord_flip()
Selecting by proportion
```




```
>
>
>
> # Inner join with nrc dictionary and create a proportion column
> sheeran_feeling <- sheeran_struc %>%
+   inner_join(get_sentiments('nrc')) %>%
+   count(sentiment, sort = TRUE) %>%
+   mutate(proportion = n/sum(n))
Joining, by = "word"
> print(sheeran_feeling)
# A tibble: 10 x 3
   sentiment       n proportion
   <chr>       <int>      <dbl>
 1 positive      130     0.182
 2 negative      122     0.171
 3 trust          82     0.115
 4 joy            69     0.0968
 5 anticipation   62     0.0870
 6 fear           62     0.0870
 7 sadness        59     0.0827
 8 anger          53     0.0743
 9 disgust        44     0.0617
10 surprise       30     0.0421
>
> # Inner join with nrc dictionary and create a proportion column
> mars_feeling <- mars_struc %>%
+   inner_join(get_sentiments('nrc')) %>%
+   count(sentiment, sort = TRUE) %>%
+   mutate(proportion = n/sum(n))
Joining, by = "word"
> print(mars_feeling)
# A tibble: 10 x 3
   sentiment       n proportion
   <chr>       <int>      <dbl>
 1 positive      103     0.188
 2 negative       84     0.153
 3 trust          62     0.113
 4 joy            61     0.111
 5 anticipation   50     0.0912
 6 fear           46     0.0839
```

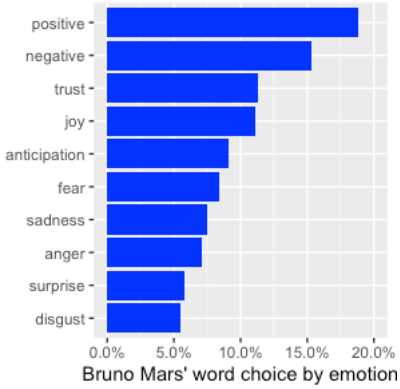
```
 7 sadness        41     0.0748
 8 anger          39     0.0712
 9 surprise       32     0.0584
10 disgust        30     0.0547
>
> # Inner join with nrc dictionary and create a proportion column
> eilish_feeling <- eilish_struc %>%
+   inner_join(get_sentiments('nrc')) %>%
+   count(sentiment, sort = TRUE) %>%
+   mutate(proportion = n/sum(n))
Joining, by = "word"
> print(eilish_feeling)
# A tibble: 10 x 3
   sentiment       n proportion
   <chr>       <int>      <dbl>
 1 negative      120     0.199
 2 positive       83     0.138
 3 fear           68     0.113
 4 sadness        67     0.111
 5 trust          52     0.0864
 6 anger          50     0.0831
 7 anticipation   48     0.0797
 8 joy            48     0.0797
 9 disgust        39     0.0648
10 surprise       27     0.0449
>
> # Inner join with nrc dictionary and create a proportion column
> beyonce_feeling <- beyonce_struc %>%
+   inner_join(get_sentiments('nrc')) %>%
+   count(sentiment, sort = TRUE) %>%
+   mutate(proportion = n/sum(n))
Joining, by = "word"
> print(beyonce_feeling)
# A tibble: 10 x 3
   sentiment       n proportion
   <chr>       <int>      <dbl>
 1 negative      138     0.183
 2 positive      124     0.165
 3 trust          82     0.109
 4 joy            71     0.0943
 5 fear           69     0.0916
 6 anticipation   64     0.0850
 7 sadness        64     0.0850
 8 anger          58     0.0770
 9 disgust        46     0.0611
10 surprise       37     0.0491
>
> # Inner join with nrc dictionary and create a proportion column
> ariana_feeling <- ariana_struc %>%
+   inner_join(get_sentiments('nrc')) %>%
+   count(sentiment, sort = TRUE) %>%
```

```
+   mutate(proportion = n/sum(n))
Joining, by = "word"
> print(ariana_feeling)
# A tibble: 10 x 3
   sentiment       n proportion
   <chr>       <int>      <dbl>
 1 negative      101     0.179
 2 positive      100     0.177
 3 trust          55     0.0975
 4 fear           54     0.0957
 5 anticipation   53     0.0940
 6 anger          47     0.0833
 7 sadness        47     0.0833
 8 joy            45     0.0798
 9 disgust        34     0.0603
10 surprise       28     0.0496
>
>
> # Plotting bar chart
> sheeran_feeling %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(sentiment=reorder(sentiment, proportion)) %>%
+   ggplot(aes(sentiment , proportion))+
+   geom_col(show.legend = FALSE, fill = 'orange', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
+   labs(y="Ed Sheeran's word choice by emotion", x=NULL)+
+   coord_flip()
Selecting by proportion
```

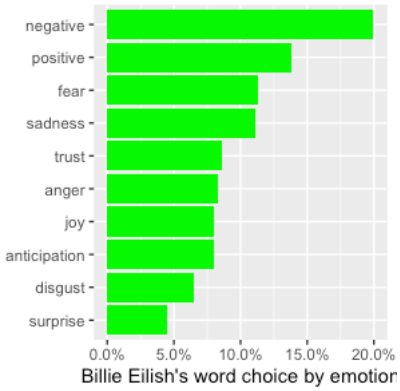



```
>
> # Plotting bar chart
> mars_feeling %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(sentiment=reorder(sentiment, proportion)) %>%
+   ggplot(aes(sentiment , proportion))+
+   geom_col(show.legend = FALSE, fill = 'blue', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
+   labs(y="Bruno Mars' word choice by emotion", x=NULL)+
```

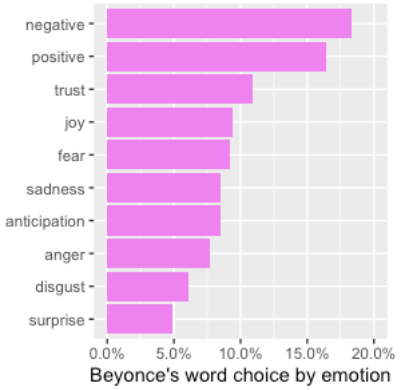
```
+   coord_flip()
Selecting by proportion
```



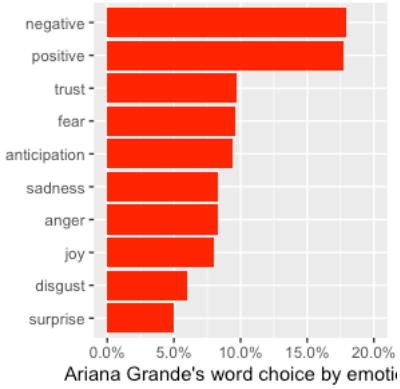

```
>
> # Plotting bar chart
> eilish_feeling %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(sentiment=reorder(sentiment, proportion)) %>%
+   ggplot(aes(sentiment , proportion))+
+   geom_col(show.legend = FALSE, fill = 'green', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
+   labs(y="Billie Eilish's word choice by emotion", x=NULL)+
+   coord_flip()
Selecting by proportion
```




```
>
> # Plotting bar chart
> beyonce_feeling %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(sentiment=reorder(sentiment, proportion)) %>%
+   ggplot(aes(sentiment , proportion))+
+   geom_col(show.legend = FALSE, fill = 'violet', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
+   labs(y="Beyonce's word choice by emotion", x=NULL)+
+   coord_flip()
Selecting by proportion
```




```
>
> # Plotting bar chart
> ariana_feeling %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(sentiment=reorder(sentiment, proportion)) %>%
+   ggplot(aes(sentiment , proportion))+
+   geom_col(show.legend = FALSE, fill = 'red', alpha = 1)+
+   scale_y_continuous(limits = c(0,0.2), labels = scales::percent) +
+   labs(y="Ariana Grande's word choice by emotion", x=NULL)+
+   coord_flip()
Selecting by proportion
```




```
>
>
> # Binding all artists by rows and calculate the probability that each words appear
> frequency <- bind_rows(mutate(sheeran_struc, Artist="Ed Sheeran"),
+                        mutate(mars_struc, Artist= "Bruno Mars"),
+                        mutate(eilish_struc, Artist="Billie Eilish"),
+                        mutate(beyonce_struc, Artist = 'Beyonce'),
+                        mutate(ariana_struc, Artist='Ariana Grande')) %>%
+   group_by(Artist) %>%
+   mutate(proportion = n/sum(n))%>%
+   select(-n) %>%
+   spread(Artist, proportion) %>%
+   gather(Artist, proportion, `Ed Sheeran`, `Bruno Mars`, `Beyonce`, `Ariana Grande`)
>
>
> # Create a correlogram
> ggplot(frequency, aes(x=proportion, y=`Billie Eilish`,
```

```
+                 color = abs(`Billie Eilish`- proportion)))+
+   geom_abline(color="grey40", lty=2)+
+   geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
+   geom_text(aes(label=word),check_overlap = TRUE, vjust=0.15) +
+   scale_x_log10(labels = percent_format())+
+   scale_y_log10(labels= percent_format())+
+   scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
+   facet_wrap(~Artist, ncol=2)+
+   theme(legend.position = "none")+
+   labs(y= "Billie Eilish", x=NULL)
Warning messages:
1: Removed 11315 rows containing missing values (geom_point).
2: Removed 11315 rows containing missing values (geom_text).
```