mini-proj4

Morgan Ryan

Description: I wanted to do text analysis on an artist I frequently listen to. I love the emotion in fleetwood mac songs and wanted to see what I could uncover. As soon as I began digging into the data, I noticed words like ah, ooh, and more were sneaking their way into my graphs, tables, and even word clouds. To fix that, I immediately filtered them out, then I also chose to filter out the apostrophes.

What did I find? One of my favorite finds was that the most commonly used word in her songs was love. This feels fitting knowing Stevie Nicks history with Lindsey Buckingham. Moving on to a side by side comparision of the top twenty words divided up into positive and negative categories, I wasn't surprised to find more negatives then positives. While Fleetwood Mac's music is catchy, I wouldn't define it as overwhelmingly positive or happy. I thought there was a creative spread of negative words like hard, blow, bad, fool, wrong, lonely. They tell a story, we know there was a breakup between Nicks and Buckingham and I feel as though their relationship was represented in here by the sheer number of angry words used. In the first word cloud, we can see how love is the largest, surrounded by lonely, wrong, free, fool, win, etc. This is an emotional word cloud and I would expect nothing less. Curious about diving deeper into the negative feelings, I made a word cloud based on words that correlated with anger. I was not surprised that Bad was the first word but it was surrounded by lonely, feeling, lose, crazy, shatter, etc. Seeing the data laid out demonstrated a clear picture of how Stevie Nicks incorporated her anger into song. Being inspired by my anger word cloud, I created another graph that would show me which songs specifically were the most "angry". I hope to listen to them with this in mind and see if I agree. Overall, this text analysis allowed me to look further into one of my favorite bands and supported my digging into the band relationships and seeing which songs correlated with specific anger sentiments. When I do more research into them, I hope to keep these findings in mind. If I wanted to continue investigating, I might try to add data with a release date to each song and see how the anger sentiments change over time.

First, finding the data I want to work with.

```
fleet_mac <- spot_mill |>
  filter(str_detect(artist, "^(f|F).*c$")) |> #Str function, regex #1
  filter(artist == "Fleetwood Mac") |>
```

```
  select(-link, -artist)

fleet_mac
```

```
# A tibble: 180 x 2
   song                        text
   <chr>                       <chr>
 1 A Fool No More              "Yes, I've packed up my clothes  \r\nI'm moving ~
 2 Affairs Of The Heart        "One set of doors was the color of honey  \r\nOn~
 3 All Over Again              "Well it's time to say goodnight  \r\nAnd finall~
 4 Allow Me One More Show      "Written by jeremy spencer.  \r\n  \r\nWhoa, tre~
 5 Although The Sun Is Shining "Although the sun is shining  \r\nHigh above  \r~
 6 Angel                       "Oh how I wish I was back in Georgia  \r\nMaybe ~
 7 As Long As You Follow       "I've been searching  \r\nFor a pot of gold  \r\~
 8 Bad Loser                   "Well you thought you had a hold on me  \r\nBut ~
 9 Bare Trees                  "Bare trees, gray light  \r\nOh yeah it was a co~
10 Beautiful Child             "Beautiful child  \r\nBeautiful child  \r\nYou a~
# i 170 more rows
```

Next, tokenize and and tidy the data

```
tidy_fleet_mac <- fleet_mac |>
  mutate(line = row_number()) |>
  unnest_tokens(word, text, token = "words")

tidy_fleet_mac
```

```
# A tibble: 33,031 x 3
   song            line word
   <chr>          <int> <chr>
 1 A Fool No More     1 yes
 2 A Fool No More     1 i've
 3 A Fool No More     1 packed
 4 A Fool No More     1 up
 5 A Fool No More     1 my
 6 A Fool No More     1 clothes
 7 A Fool No More     1 i'm
 8 A Fool No More     1 moving
 9 A Fool No More     1 away
10 A Fool No More     1 from
# i 33,021 more rows
```

```
smart_stopwords <- get_stopwords(source = "smart")
```

Table of expressive words used in songs - expressive = oohs, ahs, etc

```
tidy_fleet_mac |>
  select(-line) |>
  mutate(expression = str_count(word, "([^d]((oo|a+|yea)h)|who+a)")) |> #str_function, regex
  filter(expression > 0) |>
  count(word, expression, sort = TRUE)
```

```
# A tibble: 14 x 3
   word                          expression     n
   <chr>                              <int> <int>
 1 yeah                                   1    55
 2 whoa                                   1    30
 3 ooooh                                  1    17
 4 oooooh                                 1    10
 5 ooooooh                                1    10
 6 bah                                    1     8
 7 ooooooohhhhhhhh                        1     7
 8 oooh                                   1     5
 9 iiiiiieeeeeeiiiiiiaaaaahhhh            1     4
10 oooohh                                 1     1
11 ooooohhh                               1     1
12 ooooohhhh                              1     1
13 oooooohh                               1     1
14 ooooooohhhhhhh                         1     1
```

Graph of most common 20 words

```
tidy_fleet_mac |>
  anti_join(smart_stopwords) |>
  count(word, sort = TRUE) |>
  filter(word != "NA", str_detect(word, "[^((oo|a+|yea)h)|who+a]")) |> #Filter out expressive
  mutate(word = str_replace_all(word, "'", ""))  |> #str_function, regex #3, removing ' in wo
  slice_max(n, n = 20) |>
  ggplot(aes(fct_reorder(word, n), n, fill = n)) +
  geom_col() +
  coord_flip() +
  theme_clean() +
  labs(title = "Most Common Words in Fleetwood Mac Songs",
```

```
        x = "Word",
        y = "Count")
```

```
Joining with `by = join_by(word)`
```

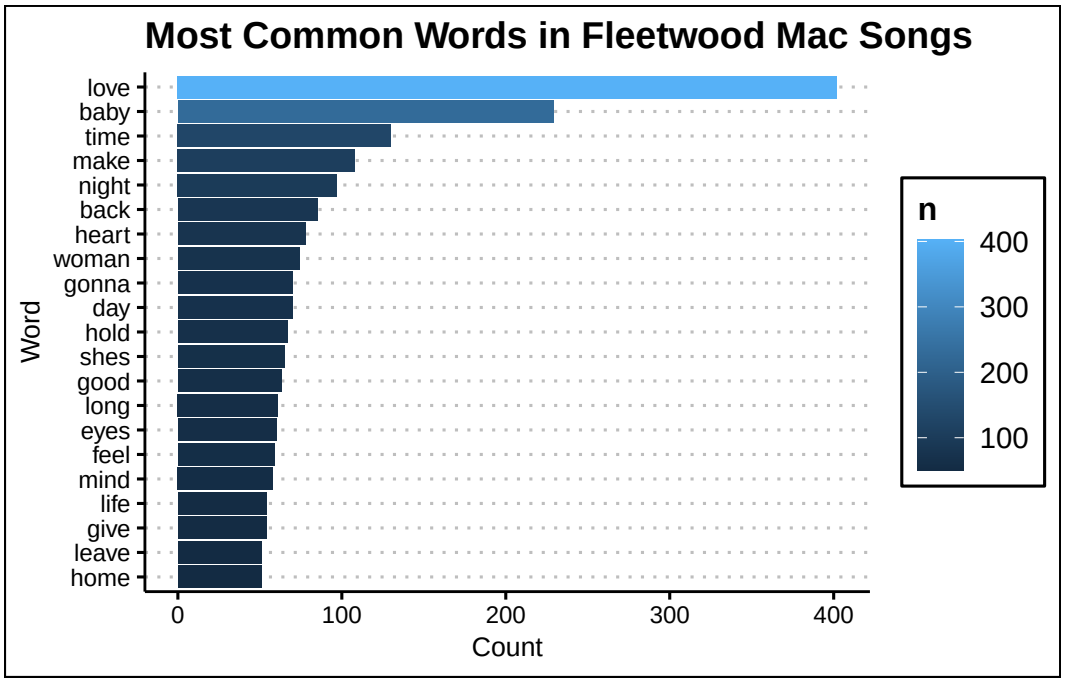


Table and Graph of top 10 positive and negative words

```
fleet_mac_sentiments <- tidy_fleet_mac |>
  inner_join(bing_sentiments) |>
  count(sentiment, word, sort = TRUE) |>
  group_by(sentiment) |>
  slice_max(n, n = 10) |>
  ungroup()
```

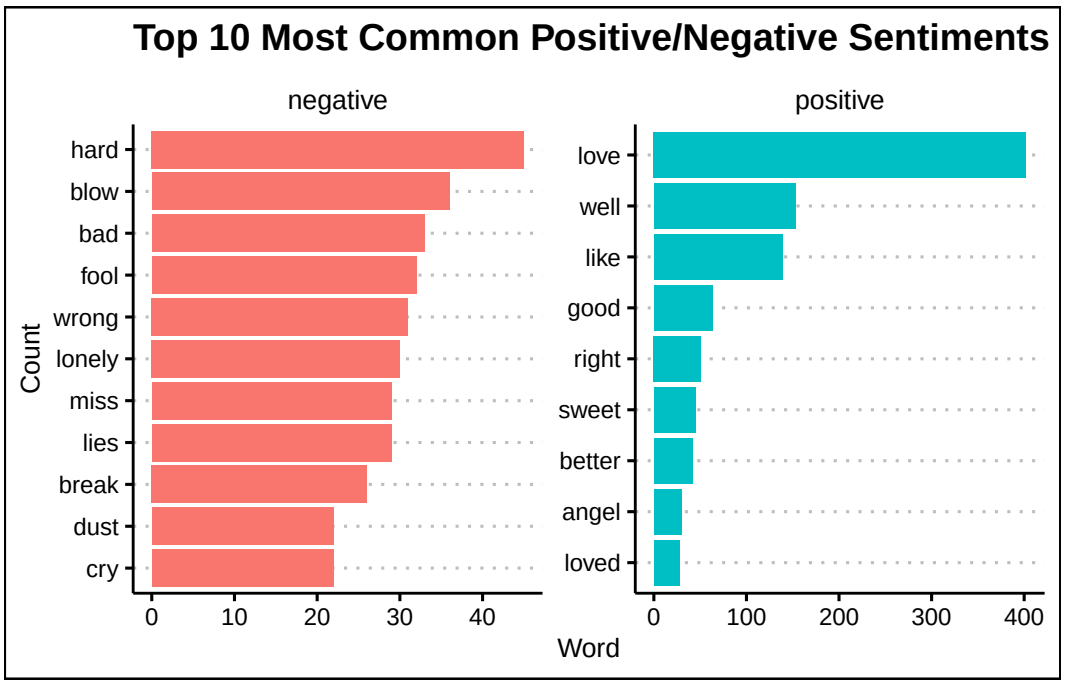
```
Joining with `by = join_by(word)`
```

```
fleet_mac_sentiments
```

```
# A tibble: 21 x 3
   sentiment word       n
```

```
   <chr>     <chr>  <int>
 1 negative  hard      45
 2 negative  blow      36
 3 negative  bad       33
 4 negative  fool      32
 5 negative  wrong     31
 6 negative  lonely    30
 7 negative  lies      29
 8 negative  miss      29
 9 negative  break     26
10 negative  cry       22
# i 11 more rows
```

```
fleet_mac_sentiments |>
  #filtering to get rid of ooh's/yeah/etc
  filter(str_detect(word, "[^((oo|a+|yea)h)|who+a]")) |>
  ggplot(aes(x = fct_reorder(word, n), y = n, fill = sentiment)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    facet_wrap(~ sentiment, scales = "free") +
  theme_clean() +
  labs(title = "Top 10 Most Common Positive/Negative Sentiments in Fleetwood Mac Songs",
       x = "Count",
       y = "Word")
```



Basic Word Cloud Using Bing Sentiments

```
words_fleet_mac <- tidy_fleet_mac |>
  anti_join(stop_words, by = join_by(word)) |>
  inner_join(bing_sentiments) |>
  count(sentiment, word, sort = TRUE) |>
  group_by(sentiment) |>
  filter(str_detect(word, "[^((oo|a+|yea)h)|who+a]")) |>
  arrange(desc(n)) |>
  mutate(colors = ifelse(sentiment == "positive", "lightgreen", "tomato"))

wordcloud(
  words = words_fleet_mac$word,
  freq = words_fleet_mac$n,
  max.words = 100,
  random.order = FALSE,
  color = words_fleet_mac$colors
)
```

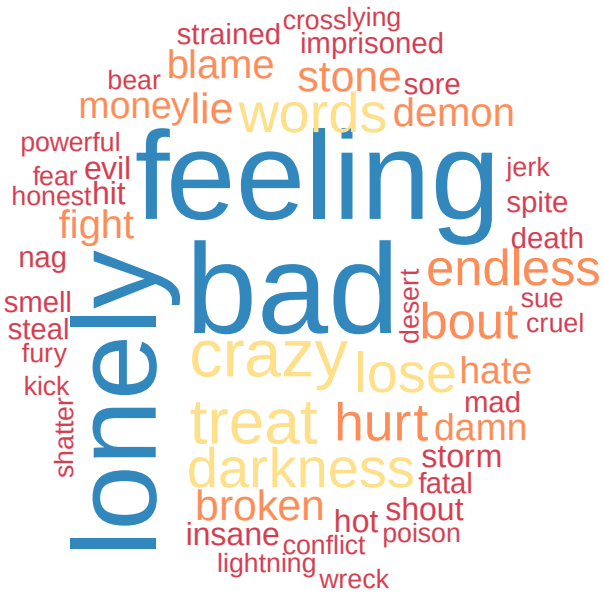
Word Cloud Using NCR Sentiments - Anger

```
ncr_sentiments <- get_sentiments(lexicon = "nrc")

angry_fleet_mac <- ncr_sentiments |>
  filter(sentiment == "anger") |>
  inner_join(tidy_fleet_mac) |>
  anti_join(stop_words, by = join_by(word)) |>
  count(word) |>
  filter(word != "NA") |>
  arrange(desc(n))
```

```
Joining with `by = join_by(word)`
```

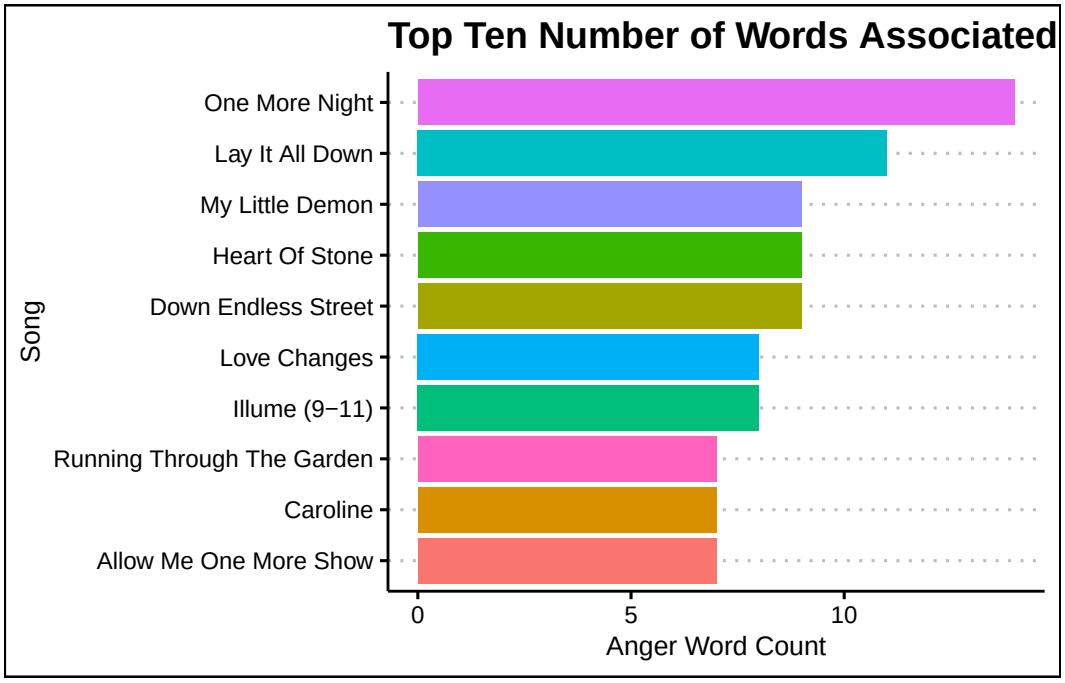
```
wordcloud(
  words = angry_fleet_mac$word,
  freq = angry_fleet_mac$n,
  max.words = 100,
  random.order = FALSE,
  color = brewer.pal(6, "Spectral")
)
```



Graph as another way to visualize anger word count using song

```
ncr_sentiments |>
  filter(sentiment == "anger") |>
  inner_join(tidy_fleet_mac) |>
  anti_join(stop_words, by = join_by(word)) |>
  count(song) |>
  slice_max(n, n = 10) |>
  arrange(desc(n)) |>
  ggplot(aes(x = fct_reorder(song, n), y = n, fill = song)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    theme_clean() +
    labs(
      title = "Top Ten Number of Words Associated with Anger by Fleetwood Mac Song",
      x = "Song",
      y = "Anger Word Count")
```

```
Joining with `by = join_by(word)`
```


Top Ten Number of Words Associated
One More Night
Lay It All Down
My Little Demon
Heart Of Stone
Down Endless Street
Love Changes
Illume (9−11)
Running Through The Garden
Caroline
Allow Me One More Show
Song
0
5
10
Anger Word Count