CP191 - Round 1 Report and Reflection

Li-Lian Ang

Minerva Schools at KGI

Abstract

This project exploration examines how textual analysis on Taylor Swift lyrics could be applied to a Capstone project on textual analysis on a corpus in the field of humanities. Given that the use of data analysis techniques in humanities is a fairly niche, the successful completion of this project exploration has shown that it would be a viable product for my Capstone project.

Introduction

At the intersection of data science and humanities lies a rich, largely unexplored field of possibilities. After completing the AH162 (Uses and Misuses of History) unit on digital public humanities last semester, I have become exposed to the untapped potential of digital tools to provide insight to the humanities. Word clouds, sentiment analysis and geospatial analysis presents an opportunity to reveal patterns in large swaths of data in a way that descriptive statistics, graphs and literature reviews cannot.

Humanities scholars have begun learning how these digital tools could power scholarly research in the humanities, but it is certainly not the norm, in fact few even possess the technical skills required to begin such analyses much less innovate with them (*Graham, et al., n.d.*). Whereas popular humanities making by journalists and other content creators have been utilising these tools more creatively to educate the public on culture, sociology and history. Some prime examples are The Pudding and the Huffington Post Highline (digital publications centred around visual essays), and Vox (a news website with beautifully presented data visualisations in their videos). These works provide insight into publicly available data that is useful in understanding

humanity. Scholarly works do the same, but the form it takes is often inaccessible to most who do not have access to academic journals or the educational background to get through the jargon that often overflows from the text.

My primary interest in the humanities is in history, particularly the large swathes of data we have encoded in census data, letters, treatises, visual art and music that paints a picture of our multi-coloured past. One idea I hoped to pursue for my Capstone project was on how I might glean insights from this data using digital tools to explore this niche intersection between technology and humanities.

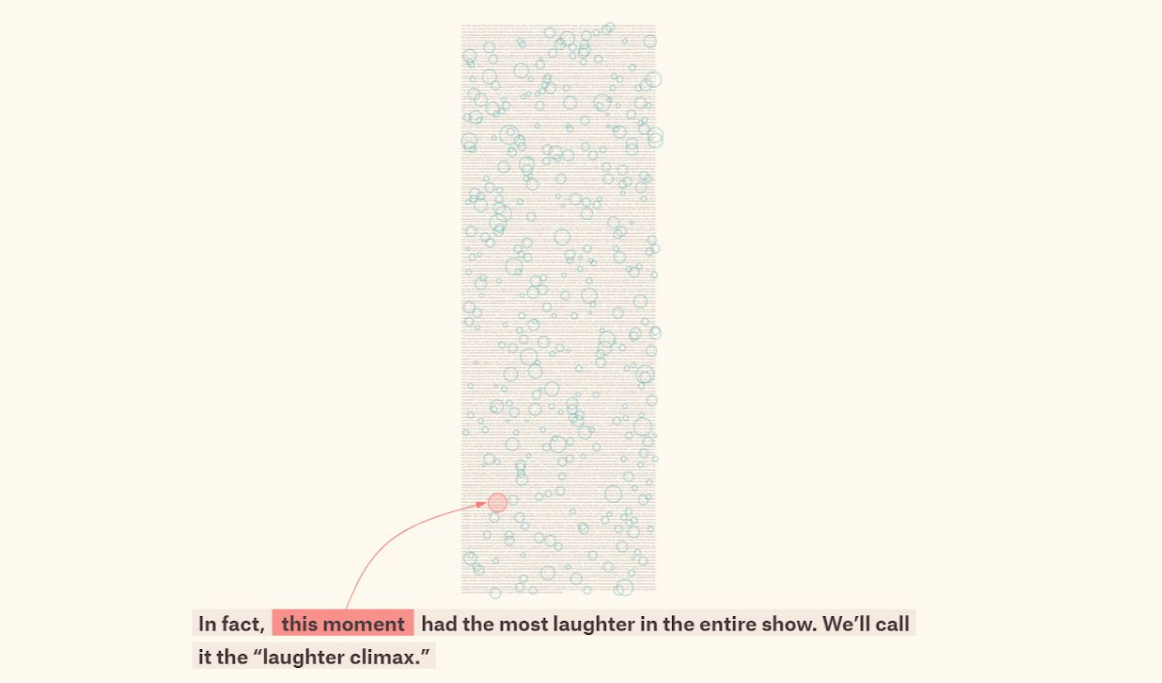

Figure 1. The Pudding article on Ali Wong Structure of Stand Up Comedy. This section analyses a stand up performance by Ali Wong and identifies pauses in the script that indicate laughter and uses these patterns to determine how the overall structure creates humour (*Goldenberg & Matt, n.d.*).

Learning JavaScript

In pursuing this project idea, I was concerned about my technical ability to perform these analyses given the array of coding languages I had to be acquainted with to replicate a Pudding article in both data analysis and visualization. At the beginning of my exploration, I devoted time to learning JavaScript, a programming language for web page interactivity which is a main component I hoped to include in my Capstone project, from a recommended online textbook. However, I found it difficult to grasp the concepts provided and felt little motivation in completing the book without an avenue to apply the skills the book sought to impart on me. It seemed unfeasible for me to become proficient in JavaScript by the summer let alone with the next few coding languages I had to learn as well before I could create a product on par with the Pudding.

Taylor Swift Lyric Analysis

I began talking to my peers about the underlying purpose behind my Capstone project which, at least to me, was for it to serve as a springboard for my future career. It led me to conversations about how I hoped to develop myself towards that end. These conversations yielded no specific conclusions, but provided me with inspiration on forms my Capstone project could take. One of the most inspirational examples came from Vick Liu, a fellow Arts & Humanities major, who introduced me to Grady Smith's analysis of country music stereotypes. It involved textual analysis, data mining and generation of interesting visualisations which were core components of my intended Capstone project, but on a smaller scale.

So came to life an analysis of a lyric analysis of Taylor Swift, my all-time favorite artist, to see if I could debunk the popular opinion that she only sings about breakups. This project

would simulate how I could perform textual analysis on a larger corpus of works in the humanities and give me a sense of the timescale required to complete a scaled version.

Data Collection and Cleaning

The first step of any data analysis begins with having data to analyse. Even though I could populate a spreadsheet of all 128 of Taylor Swift's songs complete with all relevant data on it, it would be a long, tedious task. Instead, I used an existing dataset from Kaggle that had all but her latest two albums which I populated myself (*PromptCloud, 2018*). I also added columns for songwriters, genre, nominations, awards and featured artists for each song to increase the possible permutations for analysis, but didn't end up using them during this round of exploration.

The data from Kaggle was formatted so that each line of her song was a row of data which I would have used as it were, but to include the missing 34 songs in the same way would be too time consuming. I learned a few tricks using Excel to merge all the songs lyrics from the same song into a single cell, convert the function used previously into values and replace all the line breaks with spaces (*Beer, M.D., 2017; Replace a formula with its result, n.d.; Barbara, 2013*). The properly formatted data could then be used in R, a programming language for data analysis, to further clean the data. All code for the following sections can be found in the Appendix.

artist	album	track_title	track_n	lyric	year	Song Write	Genre	Nominatio	Awards	Featured Artist
Taylor Swi	Taylor Swi	Tim McGra	1	"He said th	2006	Liz Rose	Teen Pop, Country			
Taylor Swi	Taylor Swi	Picture To	2	State the c	2006	Liz Rose	Country			
Taylor Swi	Taylor Swi	Teardrops	3	Drew looks	2006	Liz Rose	Pop	MTV Video Music Award for Best New Artist		
Taylor Swi	Taylor Swi	A Place In	4	I don't kno	2006	Robert Elli	Teen Pop, Country			
Taylor Swi	Taylor Swi	Cold as Yo	5	You have a	2006	Liz Rose	Teen Pop, Country			
Taylor Swi	Taylor Swi	The Outsid	6	I didn't knc	2006		Teen Pop			
Taylor Swi	Taylor Swi	Tied Toget	7	Seems the	2006	Liz Rose	Teen Pop, Country			
Taylor Swi	Taylor Swi	Stay Beaut	8	Cory's eyes	2006	Liz Rose	Teen Pop, Country			
Taylor Swi	Taylor Swi	Should've	9	It's strange	2006		Teen Pop, Country			
Taylor Swi	Taylor Swi	Mary's Son	10	She said, I	2006	Liz Rose, M	Teen Pop, Country			
Taylor Swi	Taylor Swi	Our Song	11	I was riding	2006		Teen Pop, Country			
Taylor Swi	Taylor Swi	I'm Only M	12	Friday nigh	2006	Angelo Thc	Teen Pop, Country			
Taylor Swi	Taylor Swi	Invisible	13	She can't s	2006	Robert Elli	Country			
Taylor Swi	Taylor Swi	A Perfectly	14	Why woulc	2006	Cornelius E	Teen Pop			
Taylor Swi	Fearless	Fearless	1	There's sor	2008					
Taylor Swi	Fearless	Fifteen	2	You take a	2008					
Taylor Swi	Fearless	Love Story	3	We were b	2008					
Taylor Swi	Fearless	Hey Steph	4	Hey Steph	2008					
Taylor Swi	Fearless	White Hor	5	Say you're	2008					
Taylor Swi	Fearless	You Belong	6	You're on t	2008					
Taylor Swi	Fearless	Breathe (F	7	I see your	2008					
Taylor Swi	Fearless	Tell Me Wl	8	I took a cha	2008					
Taylor Swi	Fearless	You're Not	9	All this tim	2008					
Taylor Swi	Fearless	The Way I	10	He is sensil	2008					
Taylor Swi	Fearless	Forever &	11	Once upon	2008					
Taylor Swi	Fearless	The Best D	12	I'm five yea	2008					
Taylor Swi	Fearless	Change	13	And it's a s	2008					
Taylor Swi	Speak Now	Mine	1	You were i	2010					
Taylor Swi	Speak Now	Sparks Fly	2	The way yc	2010					
Taylor Swi	Speak Now	Back To De	3	I'm so glad	2010					
Taylor Swi	Speak Now	Speak Now	4	I am not th	2010					
Taylor Swi	Speak Now	Dear John	5	Long were	2010					
Taylor Swi	Speak Now	Mean	6	You, with y	2010					

Figure 2. Snippet of spreadsheet with all relevant data about Taylor Swift's songs completed using Microsoft Excel.

After importing the data, I need to ensure that the data was in the right format, remove all the unnecessary punctuation and stop words. The over one thousand stop words from the tidytext package are taken from a database with prepositions, pronouns, transition words and other commonly used words while I have also added some of my own based on the initial results from analysing Taylor Swift's lyrics such as 'ah' and 'ooh'. However, it is rather problematic to simply rely on the stop words database because it may remove words that would not actually

have been irrelevant. In future analyses, I would need to go through which words are in the database and if it would be problematic based on the topic of analyses. Finally, I separated each word into a row of data.

Frequency Diagrams

A bit of Googling led to the discovery of an online textbook for text mining using R (*Silge & Robinson, 2020*). Their book centred on analysing books, but ultimately the analysis of all text is the same — the search for patterns between words that the human brain could not discern on its own. The methods I will employ largely follow from those provided in this textbook.

One of the simplest forms of textual analysis is to analyse the frequency of words which I have plotted simply across all her songs, and sorted by album. The plot in Figure 4 suffers from bad visualisation where it is difficult to glean much information from such a dense plot. Both plots do not reveal much about the lyrics except for an interesting bit of trivia.

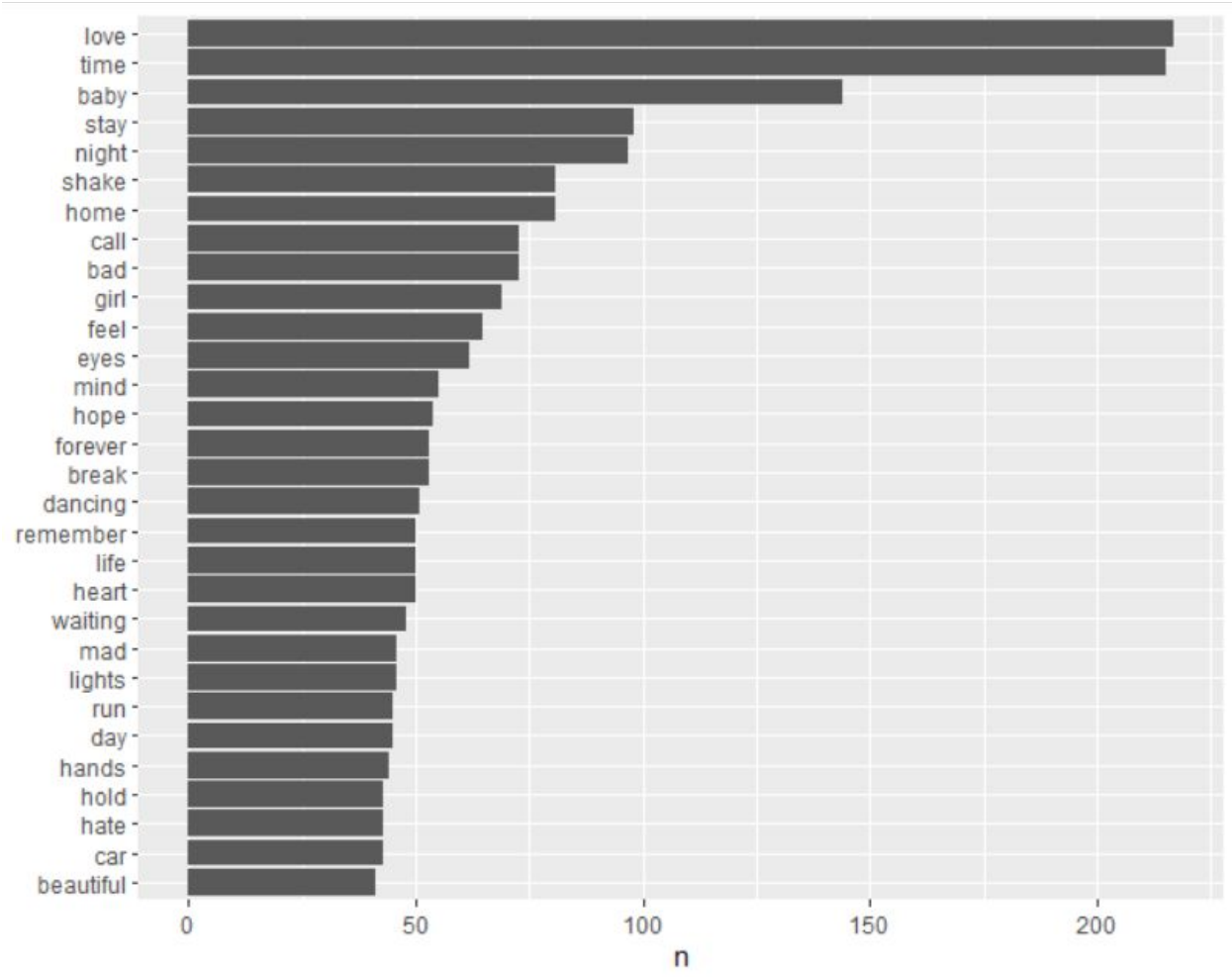

Figure 3. Frequency diagram of the most common words across all of Taylor Swift's songs for all words that occur more than 40 times.

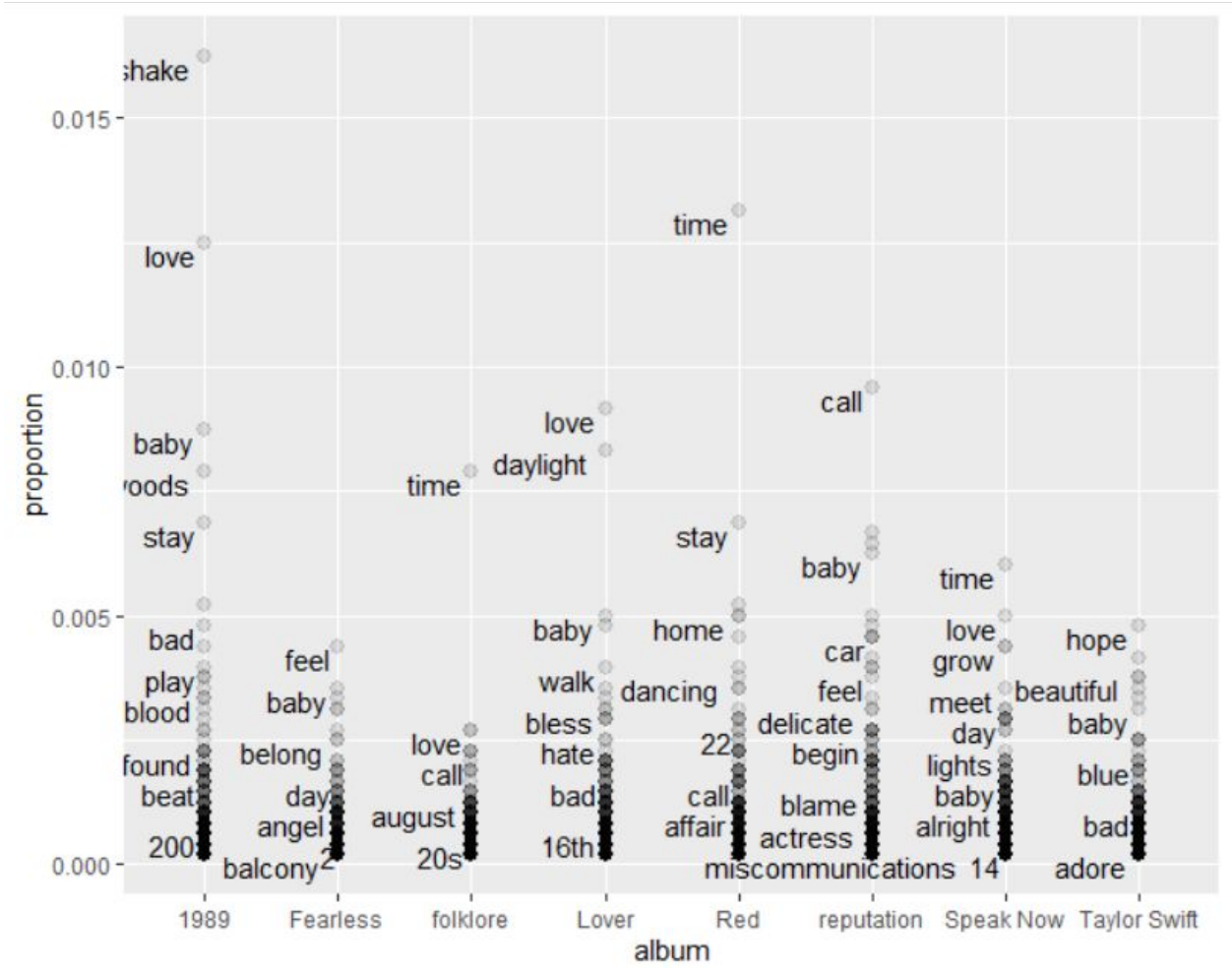

Figure 4. Plot of the frequency distribution of the most common words in Taylor Swift's songs grouped by album.

An extension of this would be to group these words by topic and determine the relative frequency of each grouping of words and each word within those groupings. The former would help to answer whether she does sing mostly about breakups and if not, what topics she does sing about. The latter would give insight into the language she uses to describe these topics and by extension her thoughts about them. I decided not to pursue the challenge of topic modelling the lyrics as the current databases that have already grouped words into topics have been created around more academic topics like economics and politics. These will be useful for analysing

topics within the humanities, but to create buckets of topics for this prototype was not a task I wanted to undertake.

Pairwise Correlation (ngram)

Instead of looking at singular words, pairwise correlation allows analysis of each word in relation to the words that accompany it. The method can be expanded to triplets, quadruplets or any number of words which is why the method is also called ngram. Figure 5 shows how pairwise correlation can be used to understand the variety of connotations a word is used in. With love, there are more positive associations with it in the words ‘beautiful’, ‘mine’ and ‘true’ than there are in the negative associations like ‘fight’, ‘miss’ and ‘leave’. These could provide evidence against the hypothesis that Taylor Swift only sings about breakups which is a lot more insightful than simply knowing that the word ‘love’ appears most frequently. The network diagram in Figure 6 strives to serve the same purpose where you can zoom in and see the relationship between each word to better understand their connotations.

Performed on an analysis of humanities texts, it could provide insights into how different phrases or words were used in the past that might change our readings of different texts since language changes over time. With the network map, it could also show how clusters of words are used together.

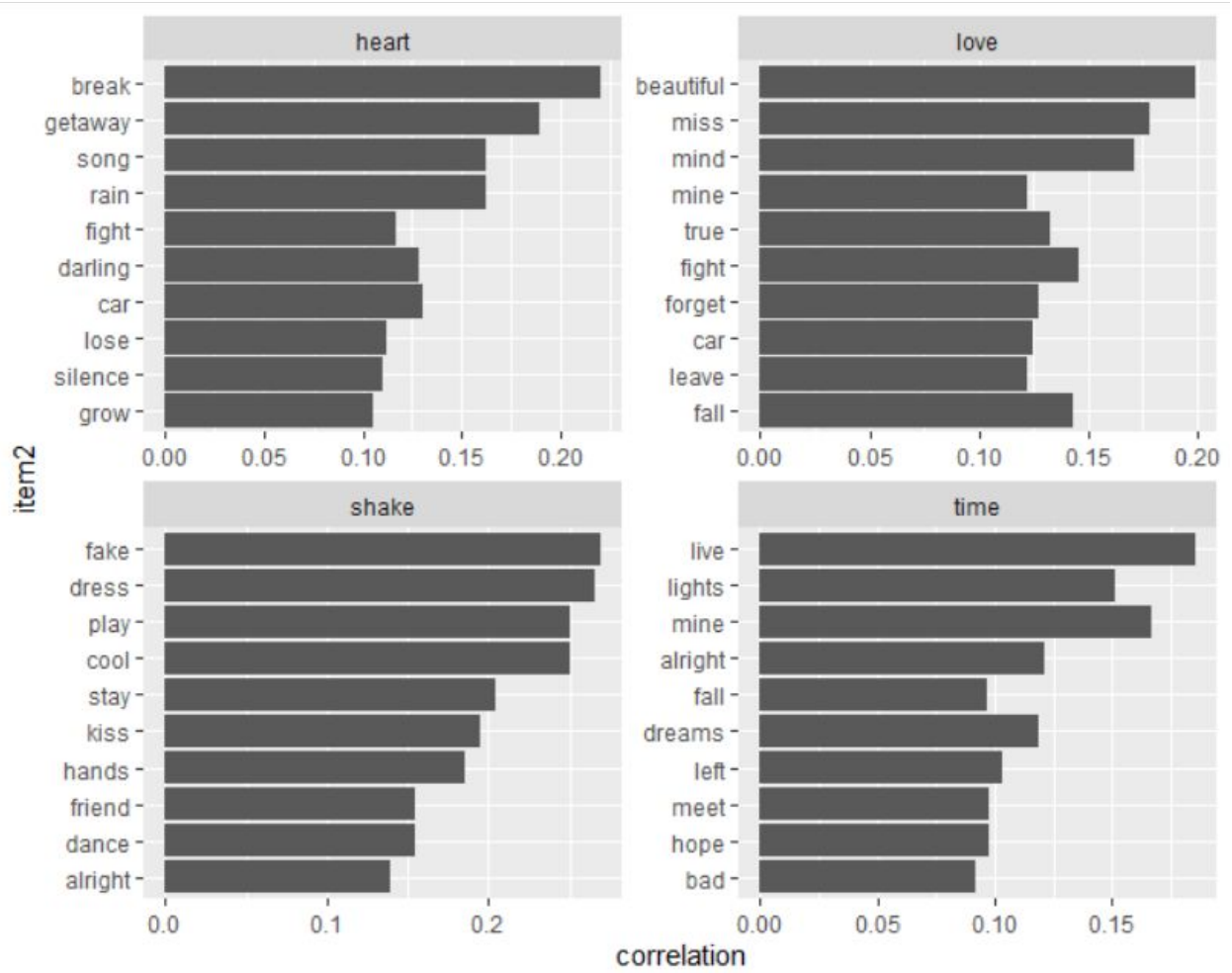

Figure 5. Top 10 words most associated with each of the words in the headers.

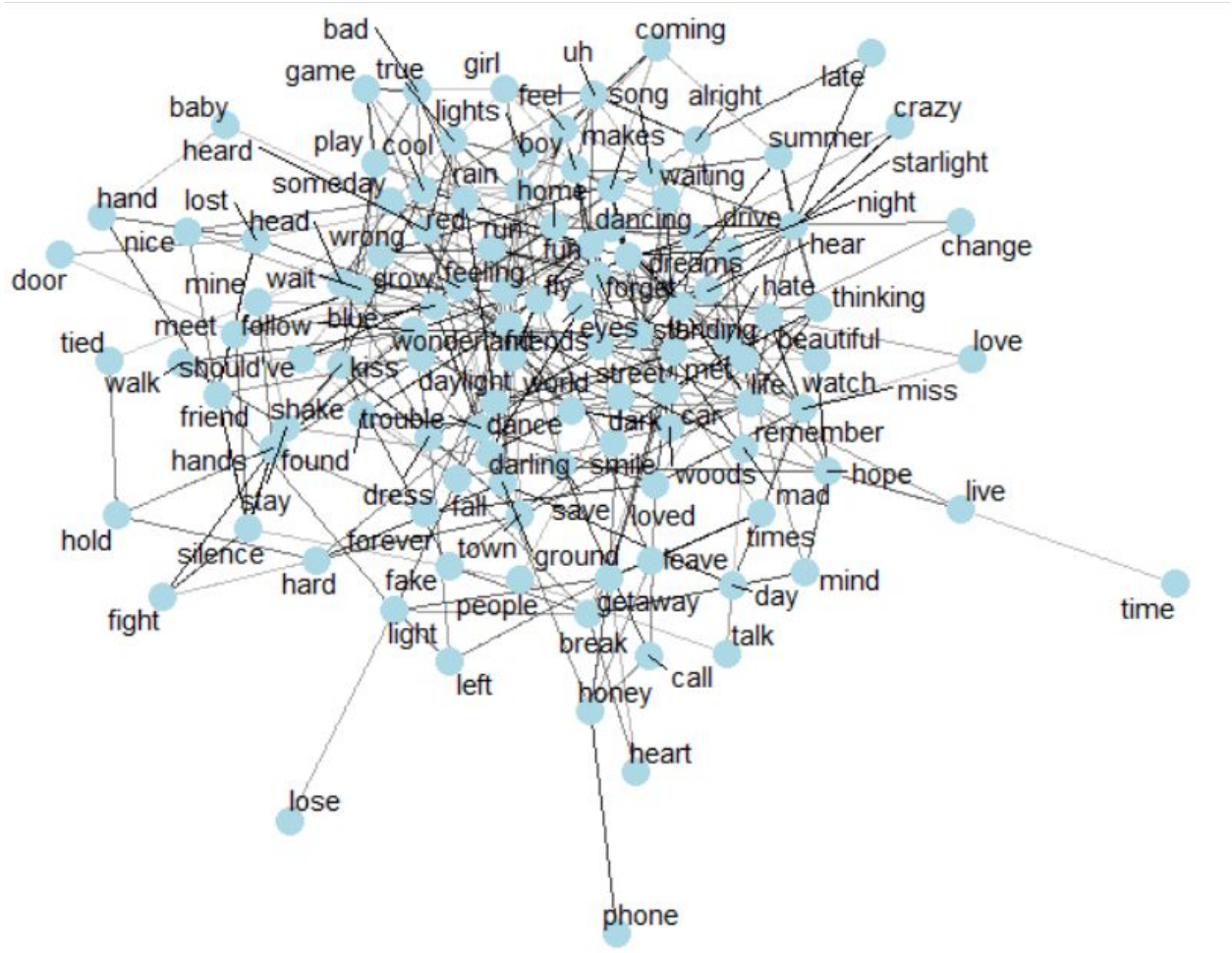

Figure 6. Network diagram of the relationship between words in Taylor Swift's lyrics.

Sentiment Analysis

To get an overall feeling of the text, the above method would be fairly tedious. There is a method called sentiment analysis that uses three different methods to categorise the feelings given by each text. The AFINN method scores each word between -5 to 5 of negative to positive sentiment, the Bing method categorises words binarily into positive or negative and the NRC method categorises words binarily within categories like 'joy', 'sadness' and 'anger'. The scores

attributed to each word are stored encoded within each method through crowdsourcing or the manual categorization.

Figure 7. Sentiment analysis of each of Taylor Swift's albums using three different methods.

Sentiment analysis is an interesting way to understand the sentiment behind a text from an objective standpoint (removed from the biases of individuals), but it is problematic. In Figure 8, the most common negative word is 'shake' which regularly is associated with a violent action, but in the context of Taylor Swift's music where one of her songs 'Shake It Off' involves heavy

repetition of the word, but used to convey a positive feeling. The same can be seen in Figure 9, where ‘Shake It Off’ is considered to be a largely negative song even though if you’ve ever heard the song, you would most likely disagree. These results point to the inability of sentiment analysis to take satire into account. The method also ignores how certain words interact with each other to create an emergent property. For example, in her song ‘All Too Well’ the line ‘We're dancing 'round the kitchen in the refrigerator light’, taken at face value appears to be an overall positive sentence, but it is *because* it paints such a happy imagery that ultimately becomes shattered at the end of the song that makes the song so much sadder. These effects are difficult to encode, but are important in understanding how language is understood by people.

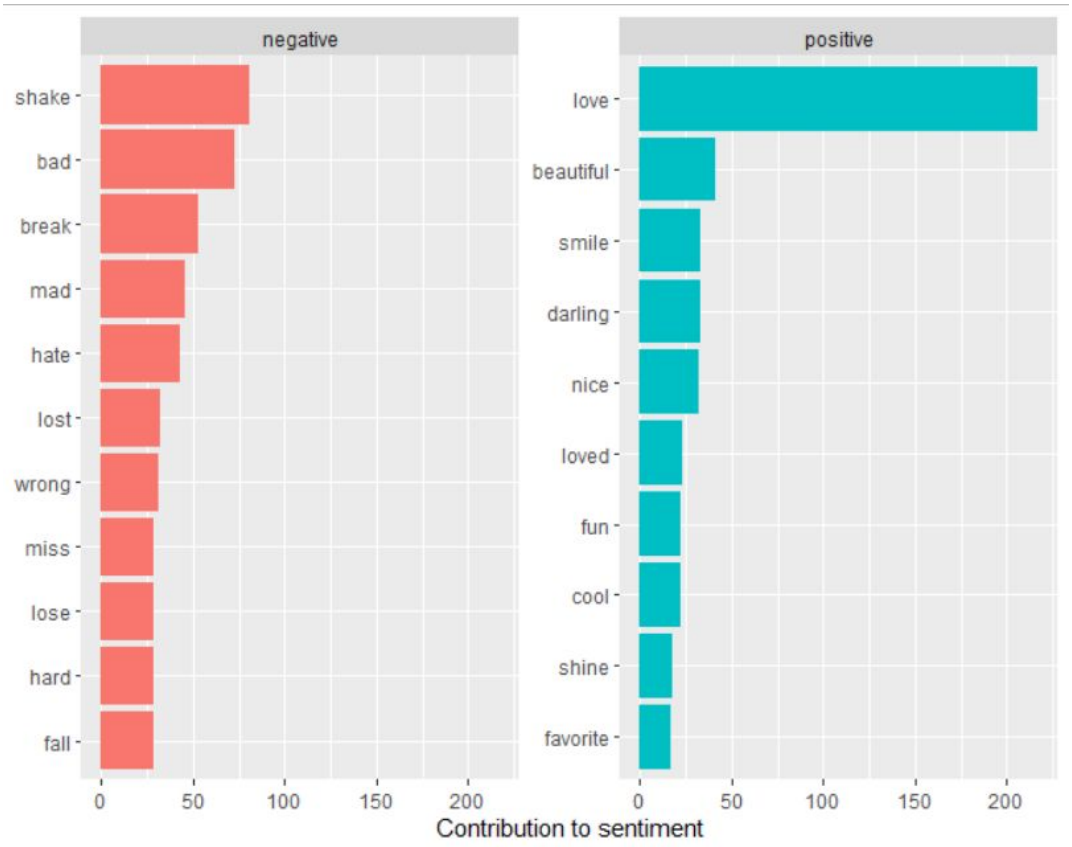

Figure 8. Most common positive and negative words across all Taylor Swift’s songs following the Bing method.

	album	track_title	negativewords	words	ratio
	<fct>	<fct>	<int>	<int>	<dbl>
1	1989	Shake It Off	137	208	0.659
2	Fearless	You're Not Sorry	14	53	0.264
3	folklore	mad woman	28	96	0.292
4	Lover	Afterglow	23	72	0.319
5	Red	I Knew You Were Trouble	50	92	0.543
6	reputation	Gorgeous	26	84	0.310
7	Speak Now	The Story of Us	34	96	0.354
8	Taylor Swift	Picture To Burn	43	100	0.43

Figure 9. Table of the ratio of negative words following the BING method in each of Taylor Swift's albums and the track with the highest proportion of negative words.

Word Cloud

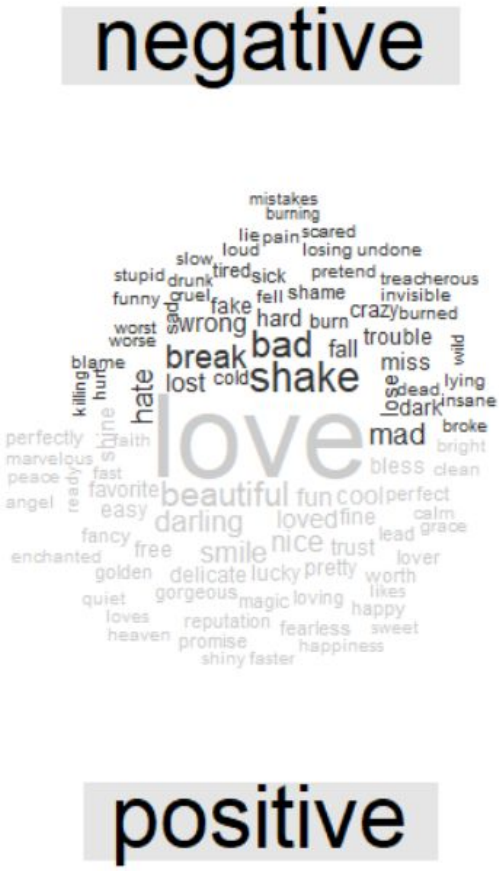

Figure 10. Word Cloud of the top 100 positive and negative words, where the size of the word relates with its frequency of occurrence.

One way of providing context is by merging sentiment analysis with ngrams. The negative sentiment of the phrase ‘never win’ or ‘not better’ is ignored with the sentiment analysis from the previous section when words are analysed individually. Figure 11 shows how we can determine the converse sentiments that are associated with a word based on having a negation word preceding it. Words we previously assumed to be strongly negatively associated like ‘miss’ and ‘leave’ in relation to our understanding of the connotation behind ‘love’ now hold even smaller weight in our understanding of how love is portrayed in Taylor Swift’s songs. This method can be extended when analysing a larger corpus of text to take into account the complex nature of language.

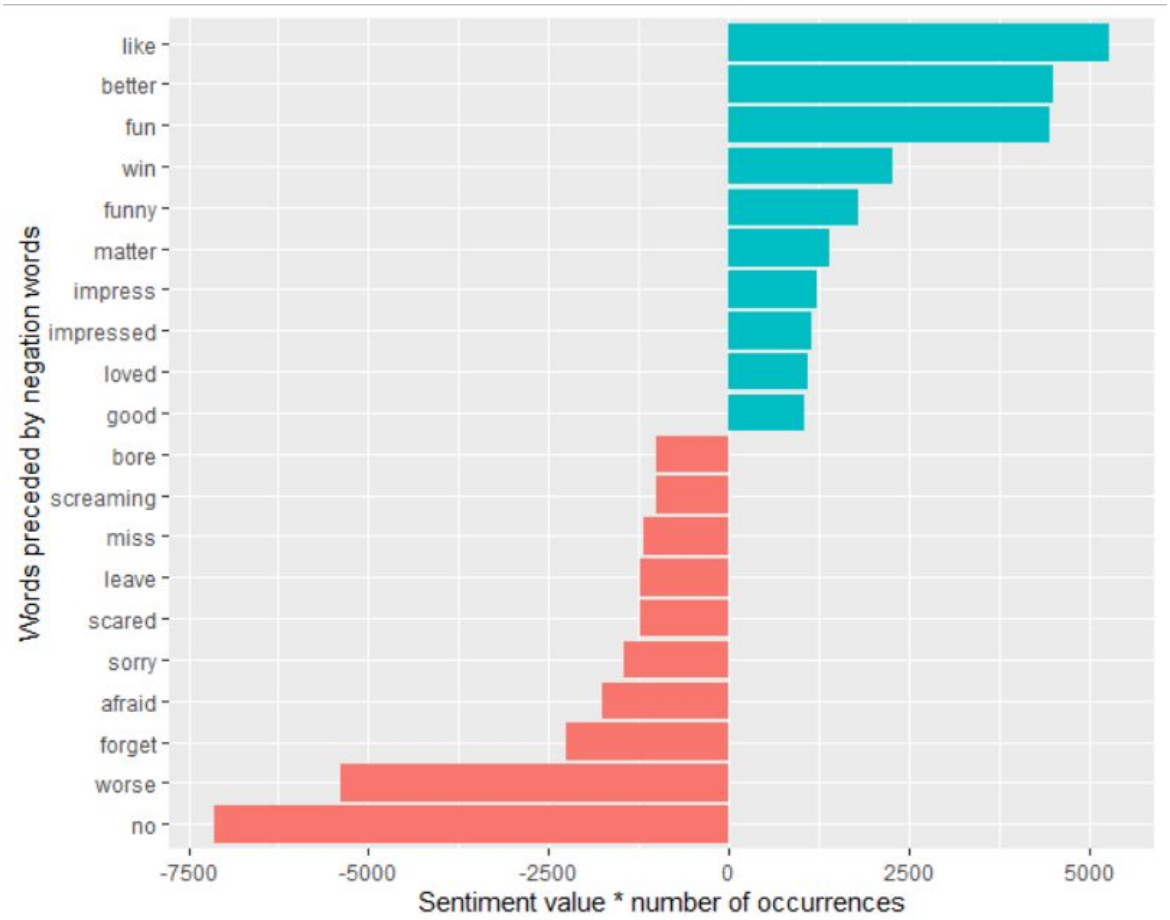

Figure 11. Sentiment analysis of the top 20 words that are preceded by a negation word.

Conclusion: Does Taylor Swift sing mostly about breakups?

From my preliminary analysis, the answer is — it's complicated. Without including an analysis of the music in itself, taking into account the influence of the co-writers of her songs and the complex relationship between words in her lyrics, it is difficult to give a concrete answer.

Analysis and Reflection

The goal of this exploration into Taylor Swift was to understand what is possible with textual analysis and if this type of analysis can be feasibly done within the time constraints. Throughout the previous section I have underscored the additional research I would have to do, namely research into more robust ways for sentiment analysis to be applied taking into consideration context and emergent properties, and ways to effectively apply topic modelling within the topic of research I choose for my Capstone project. These are challenges that are faced by those innovating within the field of textual analysis and are not unique to me, so I am fairly confident that there is a community who is actively developing solutions to the questions I posed.

On the front of feasibility, this exploration has demonstrated how achievable textual analysis is for my level of skill with programming languages. I was able to complete this simple analysis within a frame of time which when scaled by the scope of Capstone, I believe can be done. I approached each challenge throughout the process through the perspective of scalability which enabled me to learn more efficient ways to collect, clean and analyze data that I can replicate in my Capstone project. Switching to a project of desirable difficulty, where the final product required a mastery of both data analysis and data visualization as replicating the Pudding

articles would have entailed, gave me more motivation to complete the project and gain tangible new skills.

From here, I could either dive deeper into data analysis or think about how I might better present the figures. The project would in either way fulfil my two majors in Data Science & Statistics and Humanities Applications. The insights I could glean from using this method on a topic within the humanities would certainly be an appropriate Capstone project. However, as I begin to think about a second round of project exploration, I need to take into consideration what exactly it is I want to achieve with my Capstone in relation to my future career. The direction that provides will shape the focus of my Capstone project.

Description and analysis of HC and LO applications

#qualitydeliverables

The project was the appropriate depth and scope for a one month project exploration in that it could be feasibly completed during the time constraint while allowing me to learn practically how one idea for my Capstone could go. The project was completed with the appropriate rigor with justifications and in-depth explorations onto how each aspect could be scaled and applied to the context of humanities analyses.

#metrics

Identified areas in sentiment analysis, data cleaning and topic modelling to generate a more robust conclusion. I never intended to actually verify the hypothesis presented for the analysis, but instead used it as a guiding point for my project. The true aim of the project was to determine if it was feasible to complete with my level of skill and the time constraints of Capstone. These two aims were met.

#navigation

I had a growth mindset throughout the exploration where even though learning JavaScript was too time-consuming and demotivating, I did not continue to pursue a path with high effort but low rewards. Instead, I reached out to people and continued to actively think about different ways to approach my goal for the project exploration which was to test my technical skills. It yielded an in-depth exploration into textual analysis.

#modeling

Effectively developed a model to analyse Taylor Swift lyrics in terms of sentiment and its connotations. Evaluated how effective the limitations of the model in generating insights that could be used to substantiate the hypothesis and provided a critique on how it could be improved either by manually readjusting the conclusions based on your own knowledge, searching for more context specific datasets or mixing multiple methods to give the model more data to draw inferences from.

Appendix

Data collection and Cleaning

```
###IMPORT DATA###
ts <-tibble(read.csv("taylor_swift_lyrics.csv")) #import data as tibble
ts<- mutate(ts, text= as.character(ts$lyric)) #change format of lyrics from factor
into characters

###CLEANING DATA###
ts$text <-removePunctuation(ts$text, #remove punctuations
          preserve_intra_word_contractions = TRUE, #keep contractions
          preserve_intra_word_dashes = TRUE) #keep hyphenated words

#defining our stopwords
custom_stop_words <- bind_rows( tibble(
    word = c('ay',"la","ey","ah","ahh", "aah", "whoa", "ooh", "mmm","eh", "yeah",
"gonna","ha", "wanna", "hey", "york"),
    lexicon = c("custom")),
  tidytext::stop_words)

ts <- ts %>%
  tidytext::unnest_tokens(input = text, output= word, to_lower= TRUE)#separating
each word
```

Frequency Diagram

```
###FREQUENCY DIAGRAM OF WORDS###
#Creates frequency diagram of most common words in lyrics
ts %>%
  anti_join(custom_stop_words)%>%
  dplyr::count(word, sort = TRUE) %>% #counts the instances of each word
  filter(n > 40) %>% #presents words that appear more than 50 times
  mutate(word = reorder(word, n)) %>%
  #plotting
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()

###FREQUENCY DIAGRAM BY ALBUM###
albumname <- as.character(unique(ts$album)) #extracts all the album names

#creates the dataframe for the frequency of each word within each album
frequency <- ts %>%
```

```
  anti_join(custom_stop_words)%>%
  dplyr::count(album, word) %>% #counts the frequency of each word
  group_by(album) %>% #groups the count by album
  mutate(proportion = n / length(n)) %>% #calculates the proportion in relation to
the album
  select(-n) %>%

  #organise data
  tidyr::spread(album, proportion)%>%
  tidyr::gather(album, proportion, all_of(albumname))

#plot data
ggplot(frequency, aes(x = album, y=proportion)) + #data input
  geom_point(alpha = 0.1, size = 2.5) + #data point
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1, hjust = 1.2) #label
```

Pairwise Correlation (ngram)

```
###PAIRWISE CORRELATION###
#pairwise correlation
word_cors <- ts %>%
  anti_join(custom_stop_words)%>%
  group_by(word) %>%
  filter(n() >= 20) %>%
  pairwise_cor(word, track_title, sort = TRUE)

#plot to see the words associated with the filter items
word_cors %>%
  filter(item1 %in% c("love", "heart", "shake", "time")) %>% #change here
  group_by(item1) %>%
  top_n(10) %>% #top x number of associated words
  ungroup() %>%
  mutate(item2 = reorder(item2, correlation)) %>%
  ggplot(aes(item2, correlation)) +
  geom_bar(stat = "identity") +
  facet_wrap(~item1, scales = "free") +
  coord_flip()

set.seed(2016)

#visualisation in cluster
word_cors %>%
  filter(correlation > .01) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
```

```
  geom_edge_link(aes(edge_alpha = correlation), show.legend = FALSE) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), repel = TRUE) +
  theme_void()
```

Sentiment Analysis

```
###SENTIMENT ANALYSIS###
#AFINN Method - Score between -5 to 5 for sentiment
afinn <- ts %>%
  anti_join(custom_stop_words)%>%
  inner_join(get_sentiments("afinn")) %>%
  dplyr::group_by(album) %>%  #group by album
  dplyr::summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")

#Bing Method - binary categorisation into positive and negative
#NRC Method - categorises in binary between range of categories
bing_and_nrc <- bind_rows(
  ts %>%
    anti_join(custom_stop_words)%>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  ts %>%
    anti_join(custom_stop_words)%>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c(
                   "positive",
                   "negative"
                 ))) %>%
    mutate(method = "NRC")
) %>%
  dplyr::count(method, album, sentiment) %>%
  tidyr::spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

#plot sentiments
bind_rows(
  afinn,
  bing_and_nrc
) %>%
  ggplot(aes(album, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```

```
#most common positive and negative words
bing_word_counts <- ts %>%
  anti_join(custom_stop_words)%>%
  inner_join(get_sentiments("bing")) %>%
  dplyr::count(word, sentiment, sort = TRUE) %>%
  ungroup()

#plot
bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>% #number of words in each category
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(
    y = "Contribution to sentiment",
    x = NULL
  ) +
  coord_flip()

###SENTIMENT ANALYSIS BY SONG###
bingnegative <- get_sentiments("bing") %>%
  filter(sentiment == "negative") #filter by negative sentiments

wordcounts <- ts %>%
  anti_join(custom_stop_words)%>%
  group_by(album, track_title) %>%
  dplyr::summarize(words = n())

ts %>%
  semi_join(bingnegative) %>%
  group_by(album,track_title) %>%
  dplyr::summarize(negativewords = n()) %>%
  left_join(wordcounts, by = c("album", "track_title")) %>%
  mutate(ratio = negativewords / words) %>%
  top_n(1) %>%
  ungroup()

###WORDCLOUD###
#comparison word cloud
ts %>%
  anti_join(custom_stop_words)%>%
  inner_join(get_sentiments("bing")) %>%
  dplyr::count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
```

```
  comparison.cloud(
    colors = c("gray20", "gray80"),
    max.words = 100
  )

###BIGRAMS CONTEXT TO SENTIMENT ANALYSIS###
#seperate every word into pairs
ts_bigrams <- ts %>%
  tidytext::unnest_tokens(output= bigram, input = lyric, token = "ngrams", n = 2)

#split each pair of words into columns
bigrams_separated <- ts_bigrams %>%
  tidyr::separate(bigram, c("word1", "word2"), sep = " ")

#assign sentiment score to each word
AFINN <- get_sentiments("afinn")

#analyse words that are preceded by negation words
negation_words <- c("not", "no", "never", "without")

negated_words <- bigrams_separated %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  dplyr::count(word2, value, sort = TRUE)

#plot
negated_words %>%
  mutate(contribution = n * value) %>%
  arrange(desc(abs(contribution))) %>%
  head(20) %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * value, fill = n * value > 0)) +
  geom_col(show.legend = FALSE) +
  xlab("Words preceded by negation words") +
  ylab("Sentiment value * number of occurrences") +
  coord_flip()
```

References

Barbara (2013, July 31). Find And Replace Line Breaks In Excel. *How to Excel at Excel.* Retrieved from https://howtoexcelatexcel.com/excel-tips-tricks/find-and-replace-line-breaks-in-excel/

Beer, M.D. (2017). How to concatenate values in multiple cells based on a condition? Stack Overflow. Retrieved from https://stackoverflow.com/a/42698801

Goldenberg, Russel & Daniels, Matt. (n.d.). The Structure of Stand-Up Comedy. *The Pudding.* Retrieved from https://pudding.cool/2018/02/stand-up/

Graham, S., Milligan, T., & Weingart S. (n.d.) Exploring Big Historical Data: The Historian's Macroscope. Retrieved from http://www.themacroscope.org/2.0/

Haverbeke, Marjin. (2018). Eloquent Javascript 3rd Edition. *No Starch Press.* Retrieved from https://eloquentjavascript.net/index.html

PromptCloud. (2018, July 20). Taylor Swift Song Lyrics from all the albums. Retrieved from https://www.kaggle.com/PromptCloudHQ/taylor-swift-song-lyrics-from-all-the-albums

Replace a formula with its result. (n.d.). Retrieved from https://support.microsoft.com/en-us/office/replace-a-formula-with-its-result-38001951-c0e8-4fbd-9048-37ef81e2891e#:~:text=To copy the actual value,F9, and then press ENTER.

Silge, J., & Robinson, D. (2020). Text mining with R: A tidy approach. Retrieved from https://www.tidytextmining.com/index.html

Smith, Grady. (2020). Every country song has these lyrics. Right?. [Video file] *YouTube.* Retrieved from https://www.youtube.com/watch?v=48ZxNFGJTo8&ab_channel=GradySmith