Melissa Camilo

5732230

A3: Business Insight

Text Analytics – DAT-5317

Billbord top 10 ALbums

How Similar are their lyrics? What is the secret?

Table of Contents

[Introduction 3](#_Toc31996101)

[Insights 3](#_Toc31996102)

[Lyric Differentiation 3](#_Toc31996103)

[Song Variety Within the Album 3](#_Toc31996104)

[Differentiation from Other Artists 3](#_Toc31996105)

[Album Sentiment 4](#_Toc31996106)

[Conclusion 4](#_Toc31996107)

[References 4](#_Toc31996108)

[Appendices 5](#_Toc31996109)

[Appendix 1: Libraries 5](#_Toc31996110)

[Appendix 2: Billie Eilish 5](#_Toc31996111)

[Output 2.1: TF IDF Scores 8](#_Toc31996112)

[Output 2.2: TF IDF by Song 9](#_Toc31996113)

[Output 2.3: Bing Sentiment World Cloud 10](#_Toc31996114)

[Output 2.4: NRC Word Cloud 11](#_Toc31996115)

[Output 2.5: Contribution to Sentiment 12](#_Toc31996116)

[Output 2.6: Afinn Mean Sentiment 12](#_Toc31996117)

[Output 2.7: DTM 12](#_Toc31996118)

[Appendix 3: Halsey 13](#_Toc31996119)

[Output 3.1: TFIDF 15](#_Toc31996120)

[Output 3.2: TD IDF by Song 16](#_Toc31996121)

[Output 3.3: Bing Sentiment World Cloud 17](#_Toc31996122)

[Output 3.4: NRC Sentiment Word Cloud 18](#_Toc31996123)

[Outpur 3.5: Contribution to sentiment 19](#_Toc31996124)

[Output 3.6: Affin Mean Score 19](#_Toc31996125)

[Output 3.7: DTM 20](#_Toc31996126)

[Appendix 4: Selena Gomez 20](#_Toc31996127)

[Output 4.1: TFIDF 23](#_Toc31996128)

[Output 4.2: TF IDF by Song 24](#_Toc31996129)

[Output 4.3: Bing Sentiment Cloud 25](#_Toc31996130)

[Output 4.4: NRC Sentiment Cloud 26](#_Toc31996131)

[Output 4.5: Contibution to sentiment 26](#_Toc31996132)

[Output 4.6: Afinn Mean Score 27](#_Toc31996133)

[Output 4.7: DTM 27](#_Toc31996134)

[Appendix 5: Harry Styles 27](#_Toc31996135)

[Output 5.1: TF IDF 30](#_Toc31996136)

[Output 5.2: TF IDF by Song 31](#_Toc31996137)

[Output 5.3: Bing Sentiment Word Cloud 32](#_Toc31996138)

[Output 5.4: NRC Sentiment Word Cloud 33](#_Toc31996139)

[Output 5.5: Contibution to Sentiment 34](#_Toc31996140)

[Output 5.6: Afinn Mean Score 34](#_Toc31996141)

[Output 5.7: DTM 34](#_Toc31996142)

[Appendix 6: Camila Cabello 35](#_Toc31996143)

[Output 6.1: TFIDF 38](#_Toc31996144)

[Output 6.2: TF IDF by Song 39](#_Toc31996145)

[Output 6.3: Bing Sentiment Word Cloud 40](#_Toc31996146)

[Output 6.4: NRC Sentiment Word Cloud 41](#_Toc31996147)

[Output 6.5: Contribution to Sentiment 42](#_Toc31996148)

[Output 6.6: Afinn Mean Score 42](#_Toc31996149)

[Output 6.7: DTM 42](#_Toc31996150)

[Appendix 7: Artist Comparison 43](#_Toc31996151)

[Output 7.1: TF IDF 43](#_Toc31996152)

[Output 7.2: Correlograms 43](#_Toc31996153)

[Output 7.2: DTM 47](#_Toc31996154)

[Output 7.3: TF IDF by Artist 48](#_Toc31996155)

[Output 7.4: Bing Sentiment Word Cloud 49](#_Toc31996156)

[Output 7.5: NRC Sentiment Word Cloud 50](#_Toc31996157)

[Output 7.6: Contribution to Sentiment 51](#_Toc31996158)

[Output 7.7: Afinn Mean Score 51](#_Toc31996159)

# Introduction

It has been long known that every creative outlet is about love, or the absence of it. This is no different for music and the lyrics that are written in the song we listen to everyday. Songs are written all the time and all surround this one gigantic topic, but how different can all these song really be? How similar are all the best-selling song of the moments? We would like to think we have a varied taste in music and that every artist has a unique voice, but many believe that there is a recipe to getting a top-ranking album in today’s music industry. A recipe to writing the perfect lyrics that will help the album drive sales and stick to someone’s mind while still speaking to the heart and communicating some kind of emotion.

# Insights

While looking at 5 of the top ranking billboard album in the last two months it was not hard to see that there is not much variety in the genre. It took very little time to see that the word used in all of the songs were vastly different but there were similarities in all the songs, across the album and across the different artists as well.

## Lyric Differentiation

### Song Variety Within the Album

One would assume that songs in the same album which are written by the same artists would all be extremely similar and hard to differentiate. That the words used on these songs would be similar because the style of writing is the same. While there is a slight similarity within the songs in each of the albums analyzed. Surprisingly when looking at the term matrix it seems that each of the artists maintained a sparsity of approximately 90% and maintained each song independent enough so that they could be identified but also slightly similar as to indicate the style of each artist.

### Differentiation from Other Artists

“Great debut albums make their own rules and bring something truly unique or singular to the table that can speak to a mass audience.” (Dimura, 2017)This is not necessarily true for every artist analyzed here and the difference between one artist and the next is not as clear. While each song within the album was carefully curated to be unique and differentiated when looked at each artists album as a whole in comparison to other artists the songs are too similar to differentiate in many cases, which can be observed by the 62% sparsity of the Document Term Matrix. So maybe it is true that there is a recipe to have the perfect song lyrics and it is hard to differentiate one artist from the next.

Where one can see a bigger differentiator is between those artists who have been around the music scene for many years and the new ‘Breakthrough’ artists which have become popular in the last one or two years. (*see* Output 7.2: Correlograms) Looking at the Correlograms from Halsey and Billie Eilish you can see some similarities and difference to the other artists, which is okay since there will always be common words that all artists share, but when looking at the ‘older’ artists such as Harry Styles and Camila Cabello there seems to be little to no differences between the words they use. This might mean that in order to break through the industry and make it big at the beginning an artist has to be different in every aspect in order to stand out and make it to the top while the artists who have built a following through the years are simply there because of the loyalty of their fans and their sound not because they have stood out lyrically.

## Album Sentiment

“Music has the power to stimulate strong emotions within us, to the extent that it is probably rare not to be somehow emotionally affected by music.” This is why you would assume an album would have strong feeling expressed. These albums chosen have proven to have very different sentiment going from more negative ones like Billie Eilish (-0.58) or Halsey (-0.24) to some that are a little more positive like Harry Styles (0.3) or Selena Gomez (0.28) which although it might seem like the numbers are more negative than positive when looking at more albums it seems to balance out to approximately -0.04. With this in mind, you could say that the albums overall feeling really depend on the artist and the massage this want to convey to others but in general the public tends to enjoy a mix of both.

When looking individually into the words it seems like there is a balance between those with positive and negative sentiment though out most of the albums with a big emphasis on the word love for every single album. This further supports the theory that everything that involves creativity revolves around love of the lack there of, in many cases expressed in different ways but at the end all very similar.

# Conclusion

In conclusion, making a top 10 Billboard album is about either having a loyal following and writing what you already know they like; keeping the status quo and relying on your sound more than your lyrics and your message. Or you can break through the industry by being unique and different to everyone else in every aspect including sentiment and lyrical choices. Overall, no matter if you are a new or old artist, the most important thing is to have lyrics which evoke emotions and songs that have strong sentiment. They express a variety of emotions but always revolve around love. While lyrics are still important to music making sound has a much bigger impact because most of the music being released right now has very similar words or messages. New artists should definitely try to steer away from the most common words and messages rather than trying to fit in so that they can make it into the billboard top 10 one day.

# References

Cabello, C. (2019). Romance. Epic Records & Syco Music.

Dimura, D. (2017, April 13). *What Makes a Great Debut Album? These 5 Classics Can Teach Us Everything*. Retrieved from Fly Paper: https://flypaper.soundfly.com/discover/what-makes-a-great-debut-album-these-5-classics-can-teach-us-everything/

Eilish, B. (2019). When We All Fall Asleep, Where Do We Go? Darkroom & Interscope.

Fangipane, A. (2020). Manic. Capitol.

Gomez, S. (2020). Rare. Interscope.

Invisible Oranges Staff. (2011, February 15). *Why Lyrics Matter*. Retrieved from Invisible Oranges: http://www.invisibleoranges.com/why-lyrics-matter/

Kopcow, C. (27, July 27). *The 18 Ways of Beginning and Ending an Album*. Retrieved from popmatters: https://www.popmatters.com/183929-the-18-stock-ways-of-beginning-and-ending-a-song-2495636646.html?rebelltitem=1#rebelltitem1

Styles, H. (2019). Fine Line. Columbia and Erskine Records.

Tirrell, N. (n.d.). *Songwriting Advice from 10 Grammy-Nominated Songwriters*. Retrieved from Berklee take note: https://online.berklee.edu/takenote/songwriting-advice-10-grammy-nominated-songwriters/

# Appendices

## Appendix 1: Libraries

library(textreadr)

library(dplyr)

library(stringr)

library(wordcloud)

library(tidytext)

library(ggplot2)

library(reshape2)

library(tidyr)

library(textdata)

library(scales)

## Appendix 2: Billie Eilish

#Loading file

NLP <- read\_document(file="Billie Eilish.docx")

#Setting into appropiate formats

a <- 1 #how many observations to you have (text)

b <- 13 #how many variables do you have (songs)

my\_df <- as.data.frame(matrix(nrow=a, ncol=b))

for(z in 1:b){

for(i in 1:a){

my\_df[i,z]<- NLP[i\*b+z-b]

}#closing z loop

}#closing i loop

#Separating songs for tokenizing

my\_txt1 <- my\_df$V1

my\_txt2 <- my\_df$V2

my\_txt3 <- my\_df$V3

my\_txt4 <- my\_df$V4

my\_txt5 <- my\_df$V5

my\_txt6 <- my\_df$V6

my\_txt7 <- my\_df$V7

my\_txt8 <- my\_df$V8

my\_txt9 <- my\_df$V9

my\_txt10 <- my\_df$V10

my\_txt11<- my\_df$V11

my\_txt12 <- my\_df$V12

my\_txt13 <- my\_df$V13

mydf1 <- data\_frame(line=1, text=my\_txt1)

mydf2 <- data\_frame(line=1, text=my\_txt2)

mydf3 <- data\_frame(line=1, text=my\_txt3)

mydf4 <- data\_frame(line=1, text=my\_txt4)

mydf5 <- data\_frame(line=1, text=my\_txt5)

mydf6 <- data\_frame(line=1, text=my\_txt6)

mydf7 <- data\_frame(line=1, text=my\_txt7)

mydf8 <- data\_frame(line=1, text=my\_txt8)

mydf9 <- data\_frame(line=1, text=my\_txt9)

mydf10 <- data\_frame(line=1, text=my\_txt10)

mydf11 <- data\_frame(line=1, text=my\_txt11)

mydf12 <- data\_frame(line=1, text=my\_txt12)

mydf13 <- data\_frame(line=1, text=my\_txt13)

#Tokenization

data(stop\_words)

frequencies\_tokens\_nostop1 <- mydf1 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop2 <- mydf2 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop3 <- mydf3 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop4 <- mydf4 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop5 <- mydf5 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop6 <- mydf6 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop7 <- mydf7 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop8 <- mydf8 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop9 <- mydf9 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop10 <- mydf10 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop11 <- mydf11 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop12 <- mydf12 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop13 <- mydf13 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

#TFIDF

#Combining tokenized songs

combined\_b <-bind\_rows(

mutate(frequencies\_tokens\_nostop1, song = "Bad Guy"),

mutate(frequencies\_tokens\_nostop2, song = "xanny"),

mutate(frequencies\_tokens\_nostop3, song = "You Should See Me in a Crown"),

mutate(frequencies\_tokens\_nostop4, song = "all the good girls go to hell"),

mutate(frequencies\_tokens\_nostop5, song = "Wish You Were Gay"),

mutate(frequencies\_tokens\_nostop6, song = "when the party's over"),

mutate(frequencies\_tokens\_nostop7, song = "8"),

mutate(frequencies\_tokens\_nostop8, song = "My Strange Addiction"),

mutate(frequencies\_tokens\_nostop9, song = "Bury a Freind"),

mutate(frequencies\_tokens\_nostop10, song = "ilomilo"),

mutate(frequencies\_tokens\_nostop11, song = "listen before i go"),

mutate(frequencies\_tokens\_nostop12, song = "I Love You"),

mutate(frequencies\_tokens\_nostop13, song = "Good Bye"),

)

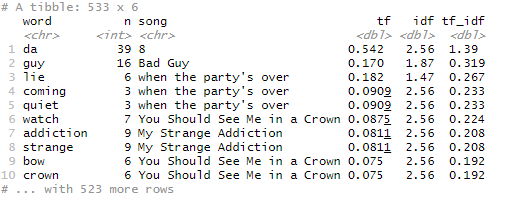
Billie\_combined <- combined\_b%>%

bind\_tf\_idf(word,song,n)

Billie\_combined %>%

arrange(desc(tf\_idf))

### Output 2.1: TF IDF Scores



Billie\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

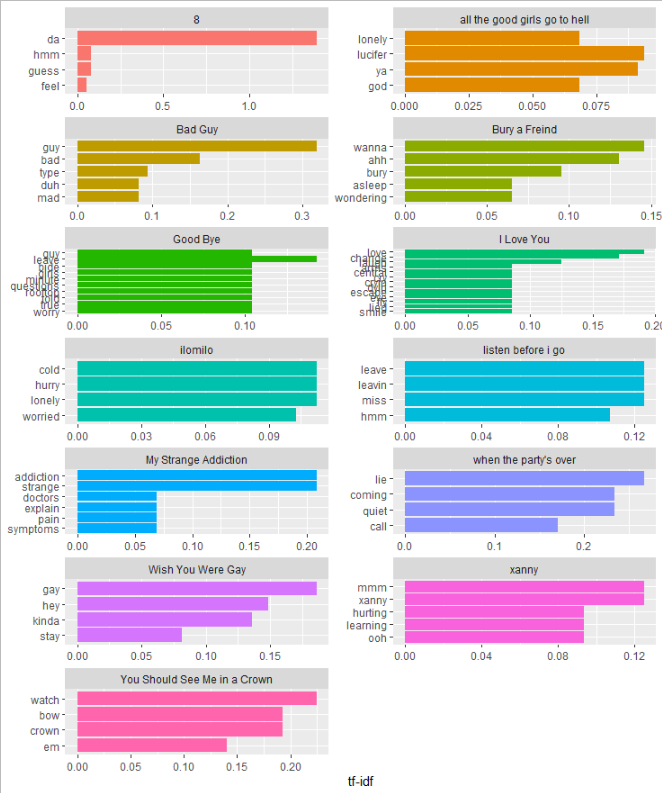
geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 2.2: TF IDF by Song



Billie\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

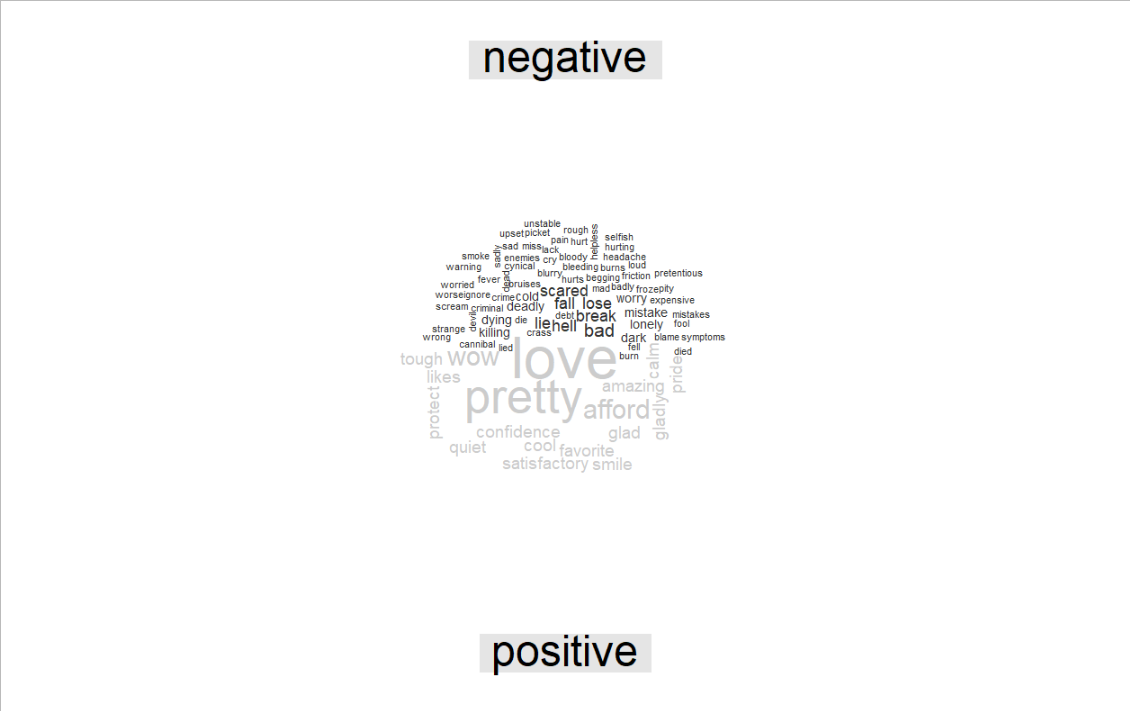
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 2.3: Bing Sentiment World Cloud



Billie\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

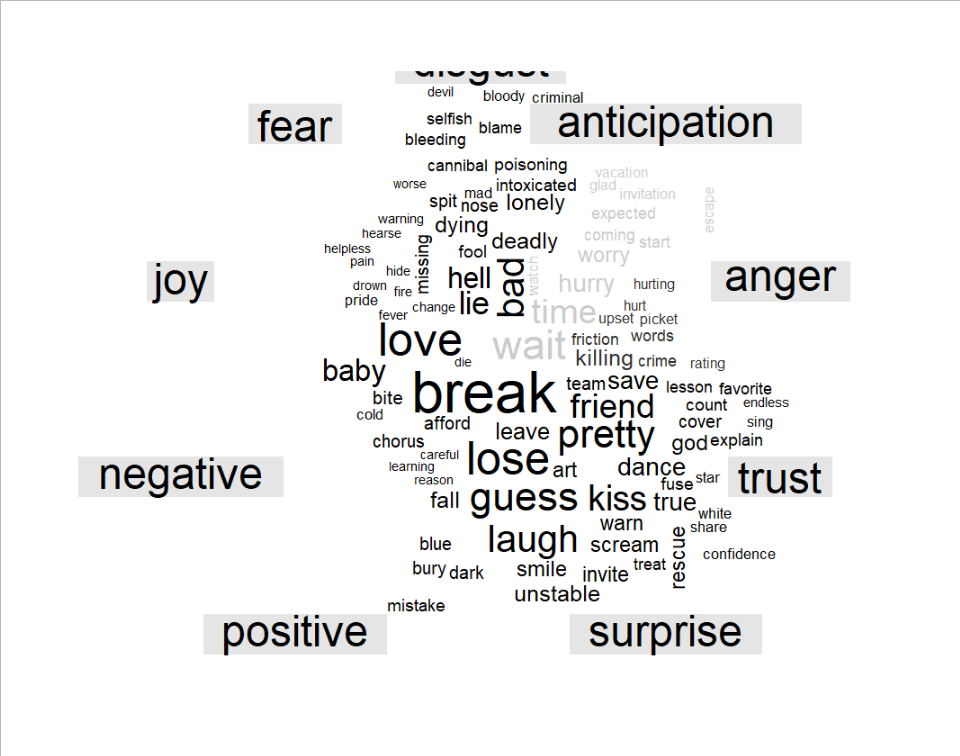
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 2.4: NRC Word Cloud



bing\_counts <- Billie\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

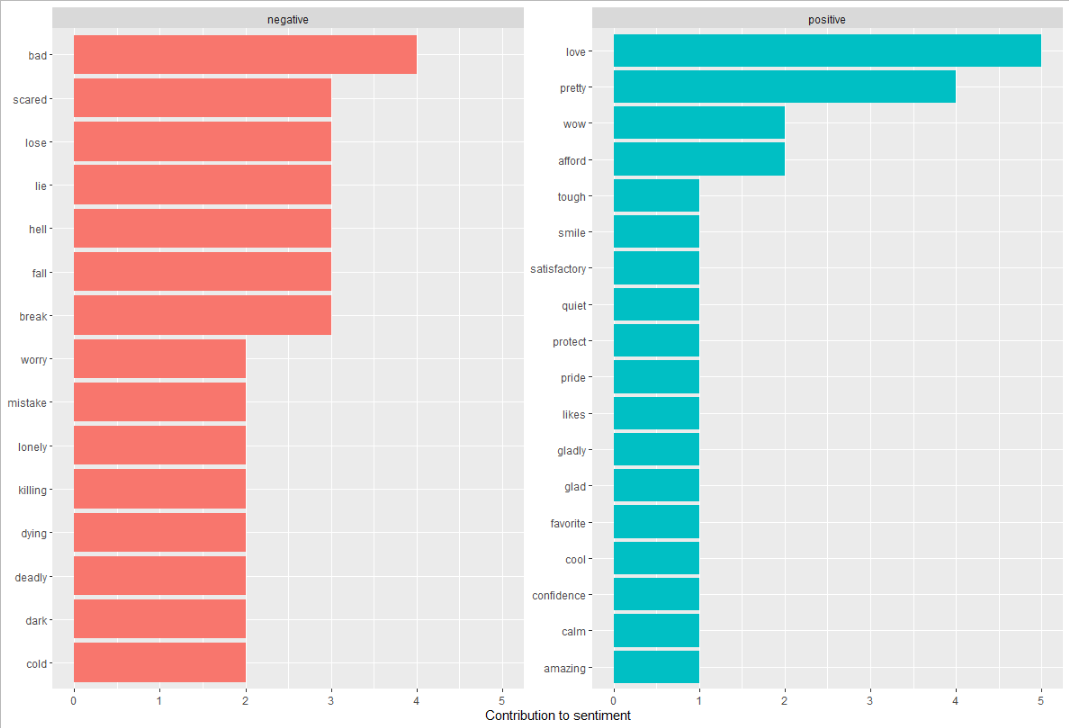
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Output 2.5: Contribution to Sentiment



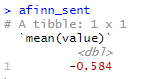
afinn\_sent <- Billie\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

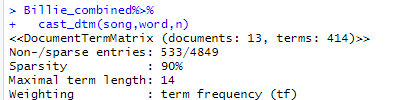
### Output 2.6: Afinn Mean Sentiment



Billie\_combined%>%

cast\_dtm(song,word,n)

### Output 2.7: DTM



## Appendix 3: Halsey

NLP <- read\_document(file="Halsey.docx")

#Setting into appropiate formats

a <- 1 #how many observations to you have (text)

b <- 13 #how many variables do you have (songs)

my\_df <- as.data.frame(matrix(nrow=a, ncol=b))

for(z in 1:b){

for(i in 1:a){

my\_df[i,z]<- NLP[i\*b+z-b]

}#closing z loop

}#closing i loop

my\_txt1 <- my\_df$V1

my\_txt2 <- my\_df$V2

my\_txt3 <- my\_df$V3

my\_txt4 <- my\_df$V4

my\_txt5 <- my\_df$V5

my\_txt6 <- my\_df$V6

my\_txt7 <- my\_df$V7

my\_txt8 <- my\_df$V8

my\_txt9 <- my\_df$V9

my\_txt10 <- my\_df$V10

my\_txt11<- my\_df$V11

my\_txt12 <- my\_df$V12

my\_txt13 <- my\_df$V13

mydf1 <- data\_frame(line=1, text=my\_txt1)

mydf2 <- data\_frame(line=1, text=my\_txt2)

mydf3 <- data\_frame(line=1, text=my\_txt3)

mydf4 <- data\_frame(line=1, text=my\_txt4)

mydf5 <- data\_frame(line=1, text=my\_txt5)

mydf6 <- data\_frame(line=1, text=my\_txt6)

mydf7 <- data\_frame(line=1, text=my\_txt7)

mydf8 <- data\_frame(line=1, text=my\_txt8)

mydf9 <- data\_frame(line=1, text=my\_txt9)

mydf10 <- data\_frame(line=1, text=my\_txt10)

mydf11 <- data\_frame(line=1, text=my\_txt11)

mydf12 <- data\_frame(line=1, text=my\_txt12)

mydf13 <- data\_frame(line=1, text=my\_txt13)

#Tokenization

data(stop\_words)

frequencies\_tokens\_nostop1 <- mydf1 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop2 <- mydf2 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop3 <- mydf3 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop4 <- mydf4 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop5 <- mydf5 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop6 <- mydf6 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop7 <- mydf7 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop8 <- mydf8 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop9 <- mydf9 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop10 <- mydf10 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop11 <- mydf11 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop12 <- mydf12 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop13 <- mydf13 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

#TFIDF

combined\_hal <-bind\_rows(

mutate(frequencies\_tokens\_nostop1, song = "Ashley"),

mutate(frequencies\_tokens\_nostop2, song = "You should be sad"),

mutate(frequencies\_tokens\_nostop3, song = "Forever"),

mutate(frequencies\_tokens\_nostop4, song = "Dominic's Interlude"),

mutate(frequencies\_tokens\_nostop5, song = "I Hate Everybody"),

mutate(frequencies\_tokens\_nostop6, song = "3am"),

mutate(frequencies\_tokens\_nostop7, song = "Finally//Beautiful Strangers"),

mutate(frequencies\_tokens\_nostop8, song = "Alanis' Interlude"),

mutate(frequencies\_tokens\_nostop9, song = "killing boys"),

mutate(frequencies\_tokens\_nostop10, song = "SUGA's Interlude"),

mutate(frequencies\_tokens\_nostop11, song = "MORE"),

mutate(frequencies\_tokens\_nostop12, song = "Still Learning"),

mutate(frequencies\_tokens\_nostop13, song = "929"),

)

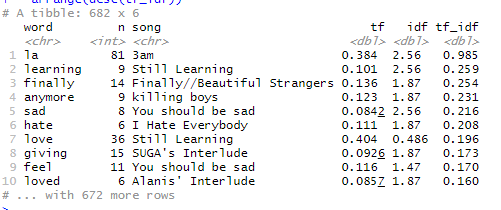
halsey\_combined <- combined\_hal%>%

bind\_tf\_idf(word,song,n)

halsey\_combined %>%

arrange(desc(tf\_idf))

### Output 3.1: TFIDF



halsey\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 3.2: TD IDF by Song



halsey\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

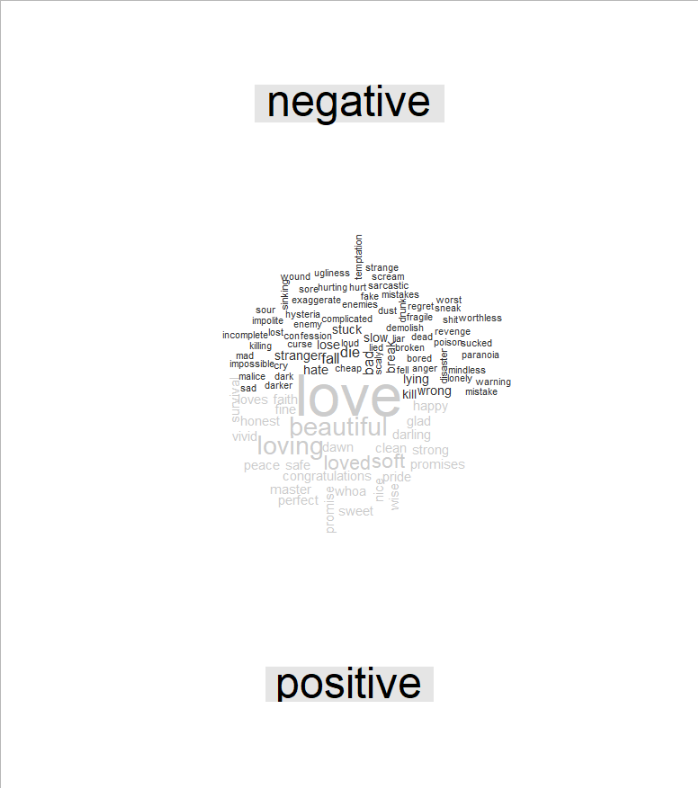
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 3.3: Bing Sentiment World Cloud



halsey\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 3.4: NRC Sentiment Word Cloud



bing\_counts <- halsey\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

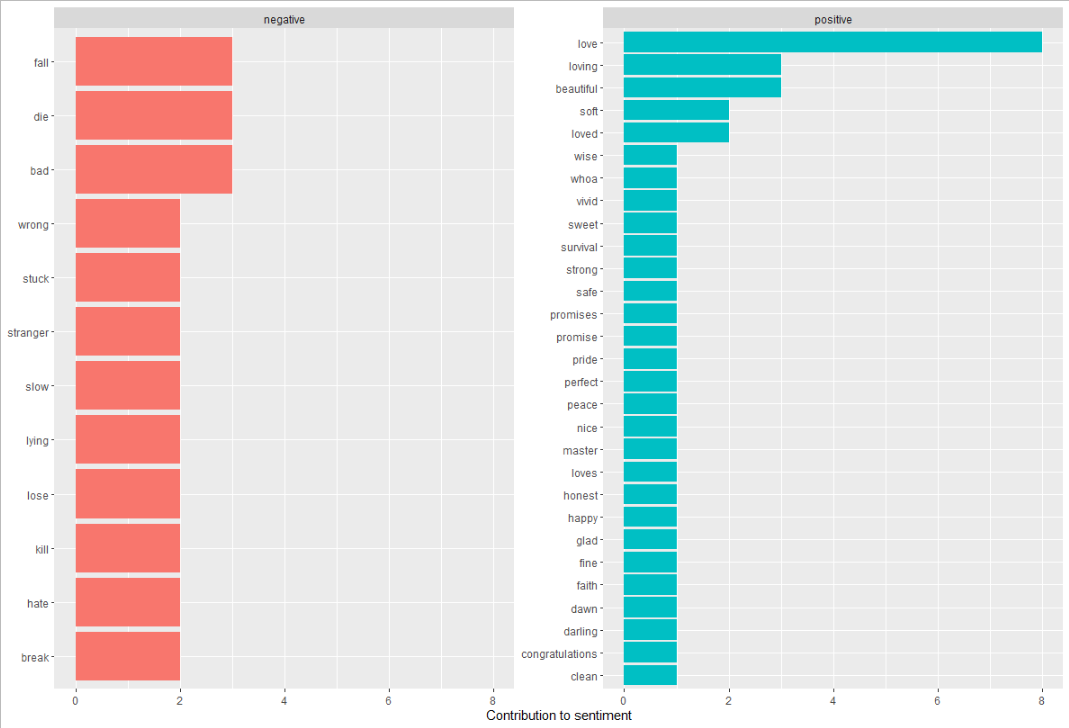
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Outpur 3.5: Contribution to sentiment



afinn\_sent <- halsey\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

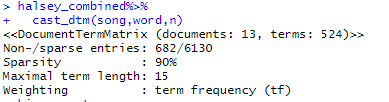
### Output 3.6: Affin Mean Score



halsey\_combined%>%

cast\_dtm(song,word,n)

### Output 3.7: DTM



## Appendix 4: Selena Gomez

NLP <- read\_document(file="Selena Gomez.docx")

#Setting into appropiate formats

a <- 1 #how many observations to you have (text)

b <- 13 #how many variables do you have (songs)

my\_df <- as.data.frame(matrix(nrow=a, ncol=b))

for(z in 1:b){

for(i in 1:a){

my\_df[i,z]<- NLP[i\*b+z-b]

}#closing z loop

}#closing i loop

my\_txt1 <- my\_df$V1

my\_txt2 <- my\_df$V2

my\_txt3 <- my\_df$V3

my\_txt4 <- my\_df$V4

my\_txt5 <- my\_df$V5

my\_txt6 <- my\_df$V6

my\_txt7 <- my\_df$V7

my\_txt8 <- my\_df$V8

my\_txt9 <- my\_df$V9

my\_txt10 <- my\_df$V10

my\_txt11<- my\_df$V11

my\_txt12 <- my\_df$V12

my\_txt13 <- my\_df$V13

mydf1 <- data\_frame(line=1, text=my\_txt1)

mydf2 <- data\_frame(line=1, text=my\_txt2)

mydf3 <- data\_frame(line=1, text=my\_txt3)

mydf4 <- data\_frame(line=1, text=my\_txt4)

mydf5 <- data\_frame(line=1, text=my\_txt5)

mydf6 <- data\_frame(line=1, text=my\_txt6)

mydf7 <- data\_frame(line=1, text=my\_txt7)

mydf8 <- data\_frame(line=1, text=my\_txt8)

mydf9 <- data\_frame(line=1, text=my\_txt9)

mydf10 <- data\_frame(line=1, text=my\_txt10)

mydf11 <- data\_frame(line=1, text=my\_txt11)

mydf12 <- data\_frame(line=1, text=my\_txt12)

mydf13 <- data\_frame(line=1, text=my\_txt13)

#Tokenization

data(stop\_words)

frequencies\_tokens\_nostop1 <- mydf1 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop2 <- mydf2 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop3 <- mydf3 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop4 <- mydf4 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop5 <- mydf5 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop6 <- mydf6 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop7 <- mydf7 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop8 <- mydf8 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop9 <- mydf9 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop10 <- mydf10 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop11 <- mydf11 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop12 <- mydf12 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop13 <- mydf13 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

#TFIDF

combined\_s <-bind\_rows(

mutate(frequencies\_tokens\_nostop1, song = "Rare"),

mutate(frequencies\_tokens\_nostop2, song = "Dance Again"),

mutate(frequencies\_tokens\_nostop3, song = "Look At Her Now"),

mutate(frequencies\_tokens\_nostop4, song = "Lose You to Love Me"),

mutate(frequencies\_tokens\_nostop5, song = "Ring"),

mutate(frequencies\_tokens\_nostop6, song = "Vulnerable"),

mutate(frequencies\_tokens\_nostop7, song = "People You Know"),

mutate(frequencies\_tokens\_nostop8, song = "Let Me Get Me"),

mutate(frequencies\_tokens\_nostop9, song = "Crowded Room"),

mutate(frequencies\_tokens\_nostop10, song = "Kinda Crazy"),

mutate(frequencies\_tokens\_nostop11, song = "Fun"),

mutate(frequencies\_tokens\_nostop12, song = "Cut You Off"),

mutate(frequencies\_tokens\_nostop13, song = "A Sweeter Place"),

)

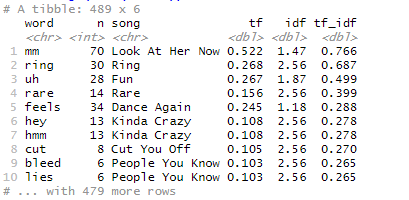
selena\_combined <- combined\_s%>%

bind\_tf\_idf(word,song,n)

selena\_combined %>%

arrange(desc(tf\_idf))

### Output 4.1: TFIDF



selena\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

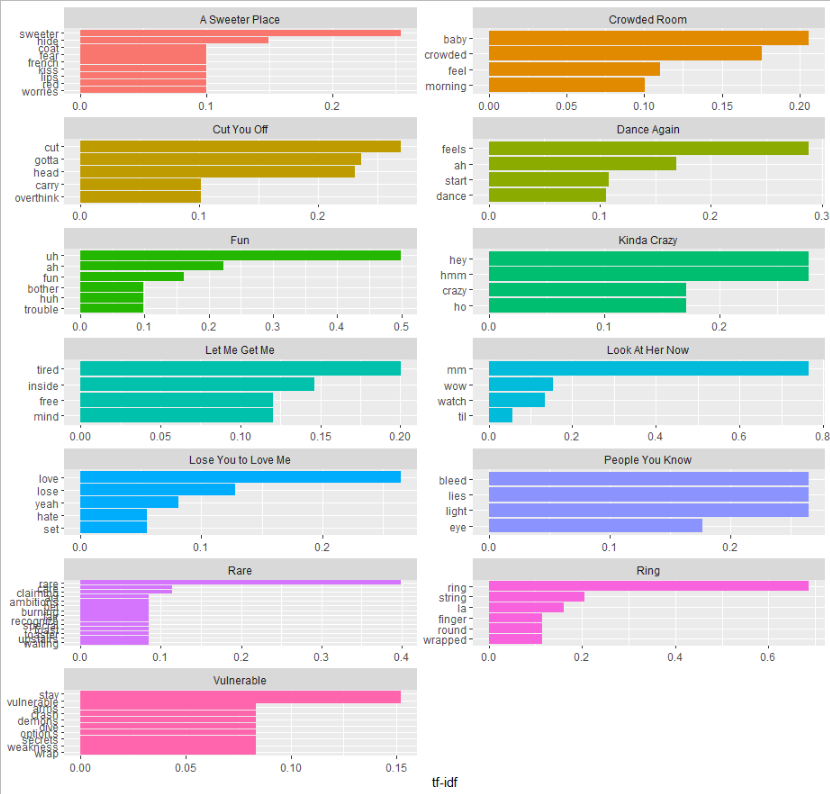
geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 4.2: TF IDF by Song



selena\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

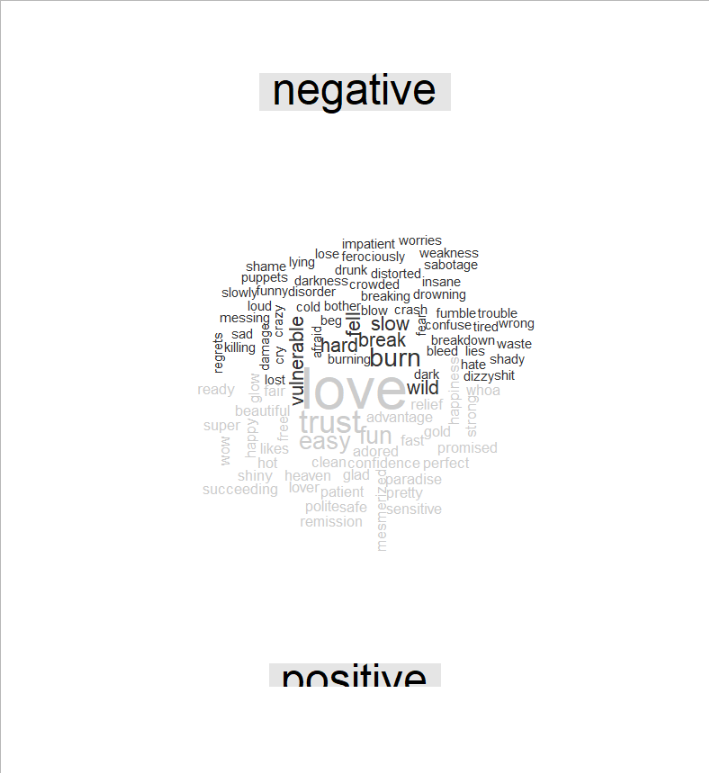
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 4.3: Bing Sentiment Cloud



selena\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

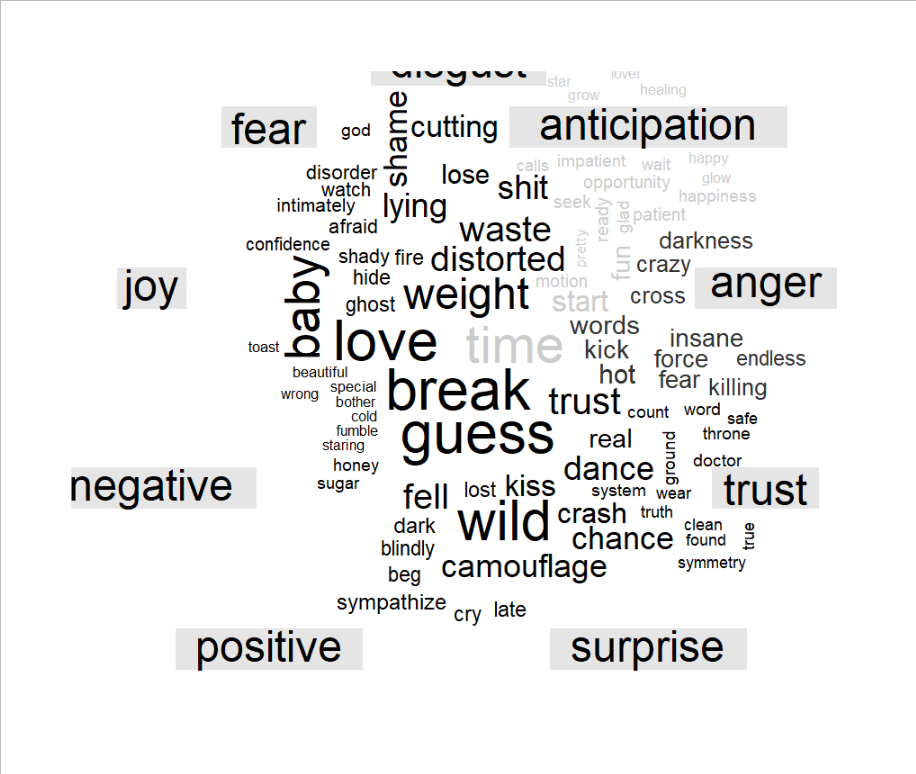
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 4.4: NRC Sentiment Cloud



bing\_counts <- selena\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

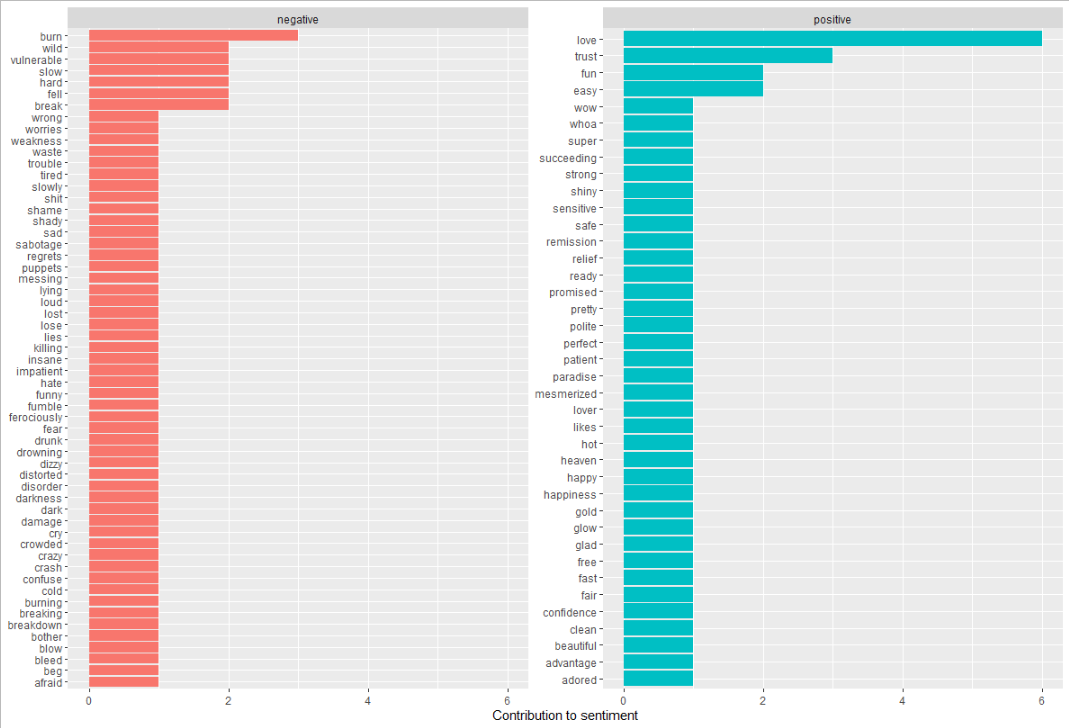
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Output 4.5: Contibution to sentiment



afinn\_sent <- selena\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

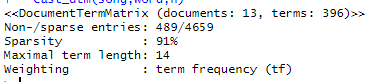
### Output 4.6: Afinn Mean Score



selena\_combined%>%

cast\_dtm(song,word,n)

### Output 4.7: DTM



## Appendix 5: Harry Styles

NLP <- read\_document(file="Selena Gomez.docx")

#Setting into appropiate formats

a <- 1 #how many observations to you have (text)

b <- 10 #how many variables do you have (songs)

my\_df <- as.data.frame(matrix(nrow=a, ncol=b))

for(z in 1:b){

for(i in 1:a){

my\_df[i,z]<- NLP[i\*b+z-b]

}#closing z loop

}#closing i loop

my\_txt1 <- my\_df$V1

my\_txt2 <- my\_df$V2

my\_txt3 <- my\_df$V3

my\_txt4 <- my\_df$V4

my\_txt5 <- my\_df$V5

my\_txt6 <- my\_df$V6

my\_txt7 <- my\_df$V7

my\_txt8 <- my\_df$V8

my\_txt9 <- my\_df$V9

my\_txt10 <- my\_df$V10

mydf1 <- data\_frame(line=1, text=my\_txt1)

mydf2 <- data\_frame(line=1, text=my\_txt2)

mydf3 <- data\_frame(line=1, text=my\_txt3)

mydf4 <- data\_frame(line=1, text=my\_txt4)

mydf5 <- data\_frame(line=1, text=my\_txt5)

mydf6 <- data\_frame(line=1, text=my\_txt6)

mydf7 <- data\_frame(line=1, text=my\_txt7)

mydf8 <- data\_frame(line=1, text=my\_txt8)

mydf9 <- data\_frame(line=1, text=my\_txt9)

mydf10 <- data\_frame(line=1, text=my\_txt10)

#Tokenization

data(stop\_words)

frequencies\_tokens\_nostop1 <- mydf1 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop2 <- mydf2 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop3 <- mydf3 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop4 <- mydf4 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop5 <- mydf5 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop6 <- mydf6 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop7 <- mydf7 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop8 <- mydf8 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop9 <- mydf9 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop10 <- mydf10 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

#TFIDF

combined\_har <-bind\_rows(

mutate(frequencies\_tokens\_nostop1, song = "Watermelon Sugar"),

mutate(frequencies\_tokens\_nostop2, song = "Adore You"),

mutate(frequencies\_tokens\_nostop3, song = "Lights Up"),

mutate(frequencies\_tokens\_nostop4, song = "Cherry"),

mutate(frequencies\_tokens\_nostop5, song = "Falling"),

mutate(frequencies\_tokens\_nostop6, song = "She"),

mutate(frequencies\_tokens\_nostop7, song = "Sunflower, Vol. 6"),

mutate(frequencies\_tokens\_nostop8, song = "Canyon Moon"),

mutate(frequencies\_tokens\_nostop9, song = "Treat People With Kindness"),

mutate(frequencies\_tokens\_nostop10, song = "Fine Line"),

)

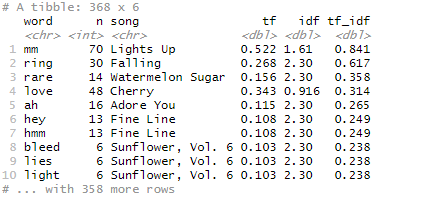
harry\_combined <- combined\_har%>%

bind\_tf\_idf(word,song,n)

harry\_combined %>%

arrange(desc(tf\_idf))

### Output 5.1: TF IDF



harry\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 5.2: TF IDF by Song



harry\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

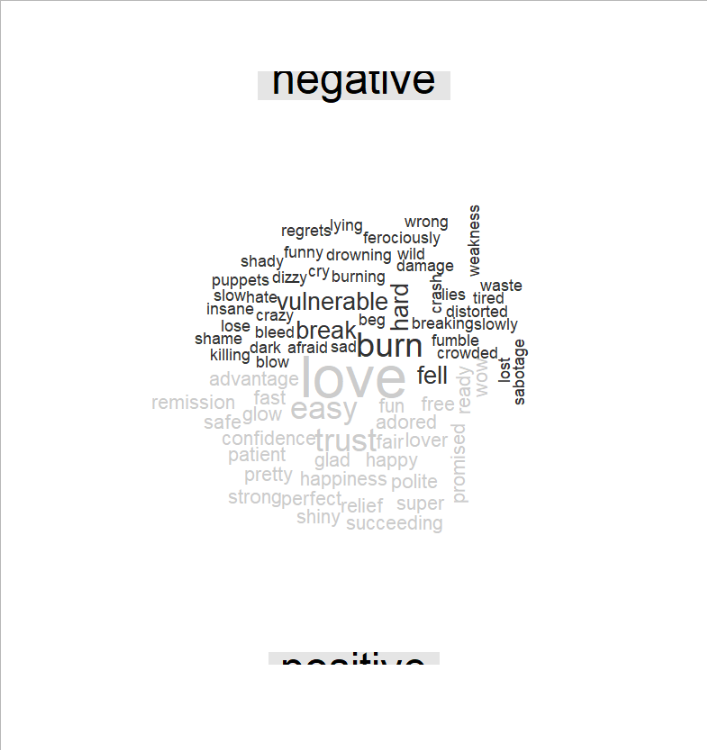
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 5.3: Bing Sentiment Word Cloud



harry\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

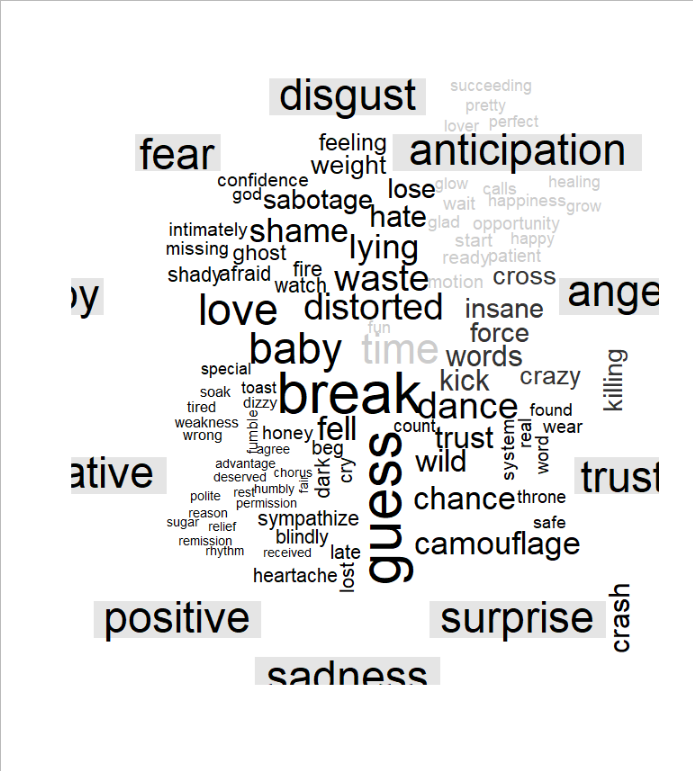
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 5.4: NRC Sentiment Word Cloud



bing\_counts <- harry\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

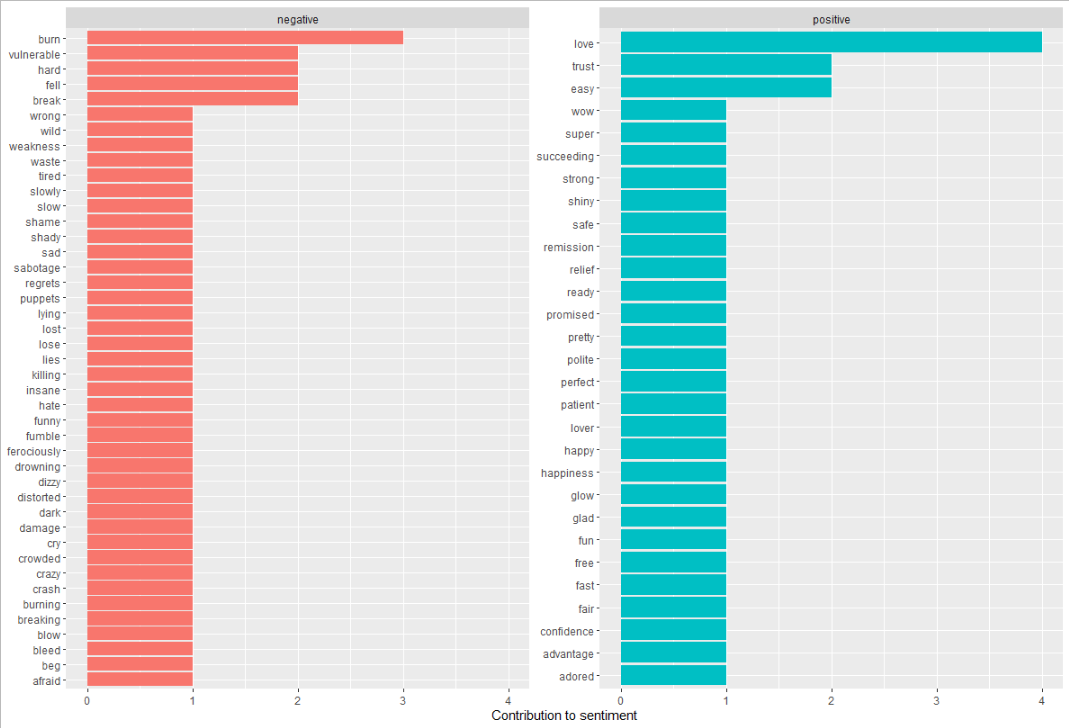
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Output 5.5: Contibution to Sentiment



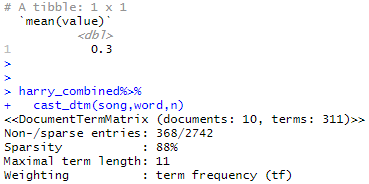
afinn\_sent <- harry\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

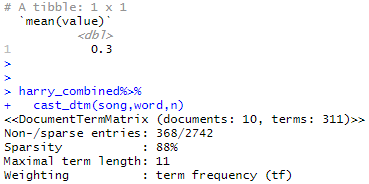
### Output 5.6: Afinn Mean Score



harry\_combined%>%

cast\_dtm(song,word,n)

### Output 5.7: DTM



## Appendix 6: Camila Cabello

NLP <- read\_document(file="Selena Gomez.docx")

#Setting into appropiate formats

a <- 1 #how many observations to you have (text)

b <- 14 #how many variables do you have (songs)

my\_df <- as.data.frame(matrix(nrow=a, ncol=b))

for(z in 1:b){

for(i in 1:a){

my\_df[i,z]<- NLP[i\*b+z-b]

}#closing z loop

}#closing i loop

my\_txt1 <- my\_df$V1

my\_txt2 <- my\_df$V2

my\_txt3 <- my\_df$V3

my\_txt4 <- my\_df$V4

my\_txt5 <- my\_df$V5

my\_txt6 <- my\_df$V6

my\_txt7 <- my\_df$V7

my\_txt8 <- my\_df$V8

my\_txt9 <- my\_df$V9

my\_txt10 <- my\_df$V10

my\_txt11<- my\_df$V11

my\_txt12 <- my\_df$V12

my\_txt13 <- my\_df$V13

my\_txt14 <- my\_df$V14

mydf1 <- data\_frame(line=1, text=my\_txt1)

mydf2 <- data\_frame(line=1, text=my\_txt2)

mydf3 <- data\_frame(line=1, text=my\_txt3)

mydf4 <- data\_frame(line=1, text=my\_txt4)

mydf5 <- data\_frame(line=1, text=my\_txt5)

mydf6 <- data\_frame(line=1, text=my\_txt6)

mydf7 <- data\_frame(line=1, text=my\_txt7)

mydf8 <- data\_frame(line=1, text=my\_txt8)

mydf9 <- data\_frame(line=1, text=my\_txt9)

mydf10 <- data\_frame(line=1, text=my\_txt10)

mydf11 <- data\_frame(line=1, text=my\_txt11)

mydf12 <- data\_frame(line=1, text=my\_txt12)

mydf13 <- data\_frame(line=1, text=my\_txt13)

mydf14 <- data\_frame(line=1, text=my\_txt14)

#Tokenization

data(stop\_words)

frequencies\_tokens\_nostop1 <- mydf1 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop2 <- mydf2 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop3 <- mydf3 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop4 <- mydf4 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop5 <- mydf5 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop6 <- mydf6 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop7 <- mydf7 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop8 <- mydf8 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop9 <- mydf9 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop10 <- mydf10 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop11 <- mydf11 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop12 <- mydf12 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop13 <- mydf13 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

frequencies\_tokens\_nostop14 <- mydf14 %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>% #here's where we remove tokens

count(word, sort=TRUE)

#TFIDF

combined\_c <-bind\_rows(

mutate(frequencies\_tokens\_nostop1, song = "Shameless"),

mutate(frequencies\_tokens\_nostop2, song = "Living Proof"),

mutate(frequencies\_tokens\_nostop3, song = "Should've Said it"),

mutate(frequencies\_tokens\_nostop4, song = "My Oh My"),

mutate(frequencies\_tokens\_nostop5, song = "Senorita"),

mutate(frequencies\_tokens\_nostop6, song = "Liar"),

mutate(frequencies\_tokens\_nostop7, song = "Bad Kind of Butterflies"),

mutate(frequencies\_tokens\_nostop8, song = "Easy"),

mutate(frequencies\_tokens\_nostop9, song = "Feel it Twice"),

mutate(frequencies\_tokens\_nostop10, song = "Dream of You"),

mutate(frequencies\_tokens\_nostop11, song = "Cry for Me"),

mutate(frequencies\_tokens\_nostop12, song = "This Love"),

mutate(frequencies\_tokens\_nostop13, song = "Used to This"),

mutate(frequencies\_tokens\_nostop14, song = "First Man"),

)

camila\_combined <- combined\_c%>%

bind\_tf\_idf(word,song,n)

camila\_combined %>%

arrange(desc(tf\_idf))

### Output 6.1: TFIDF



camila\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

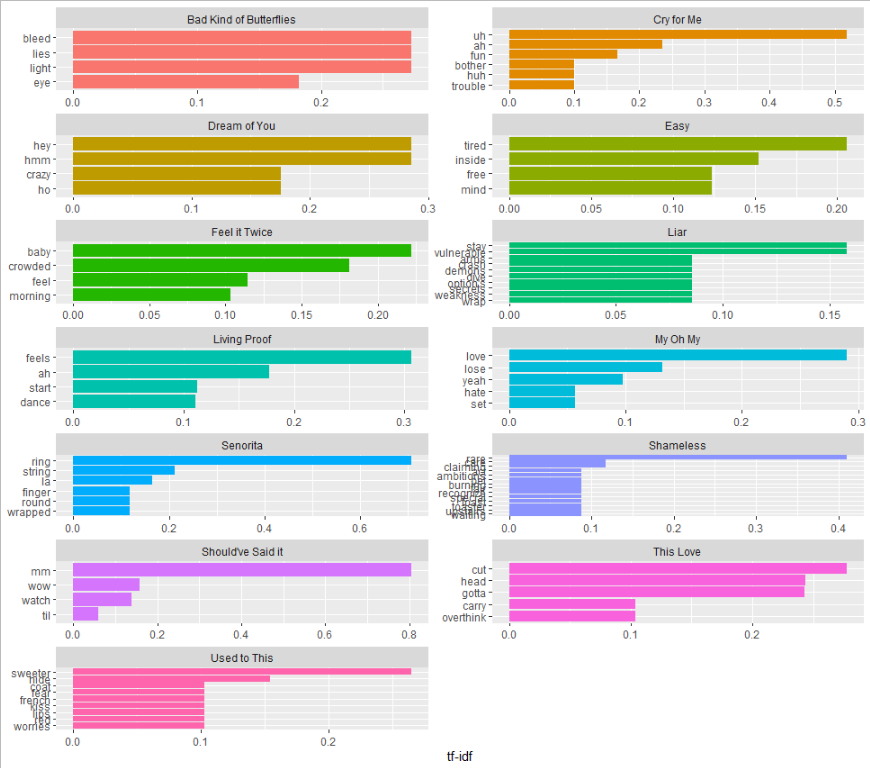
geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 6.2: TF IDF by Song



camila\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

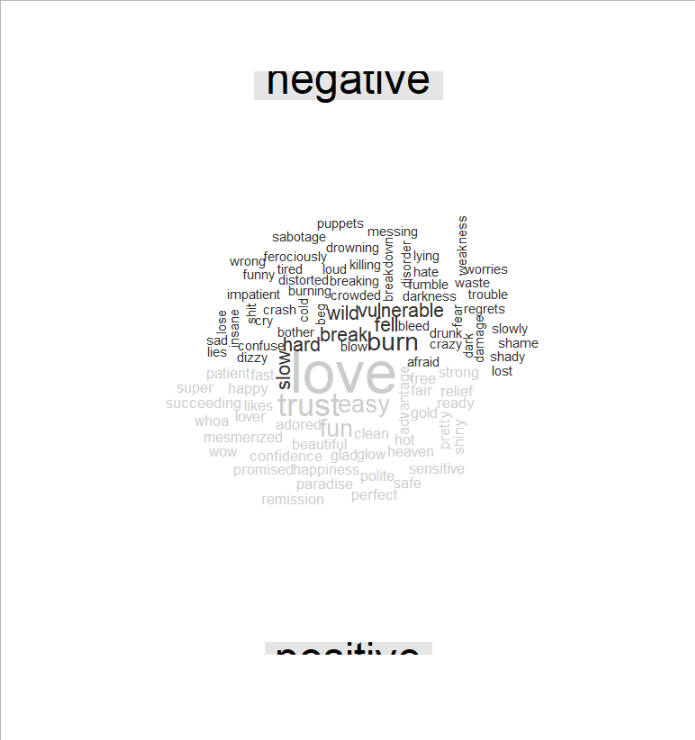
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 6.3: Bing Sentiment Word Cloud



camila\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 6.4: NRC Sentiment Word Cloud



bing\_counts <- camila\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

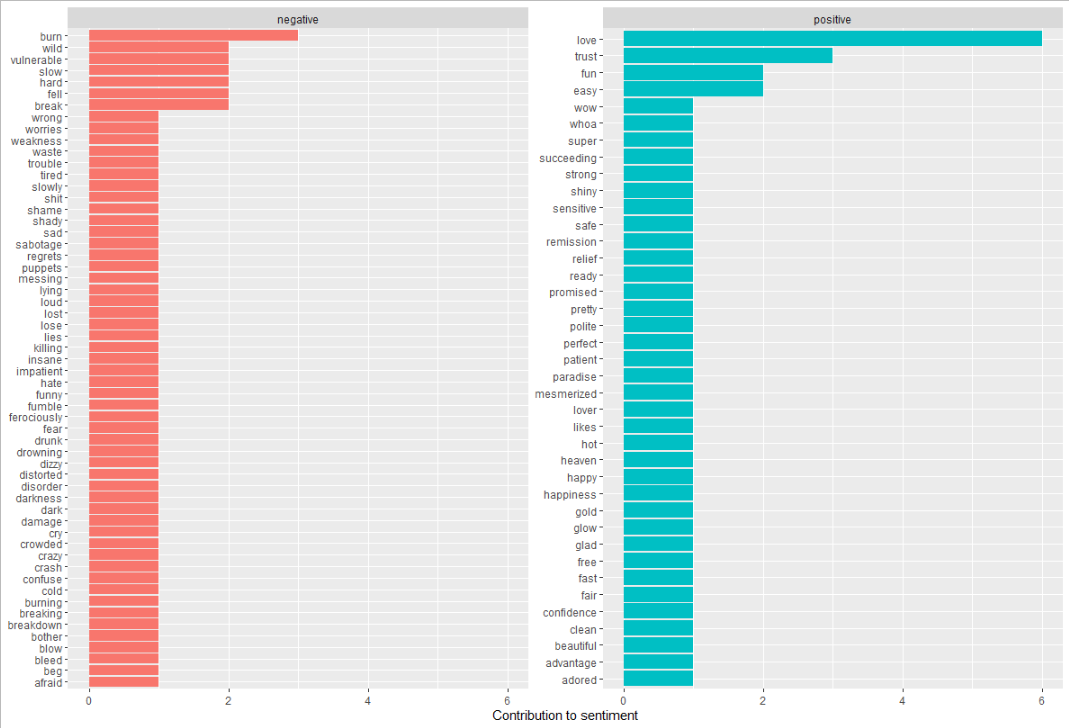
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Output 6.5: Contribution to Sentiment



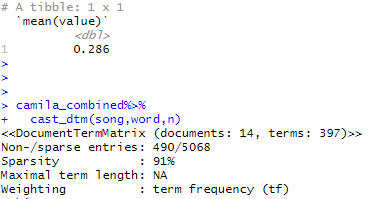
afinn\_sent <- camila\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

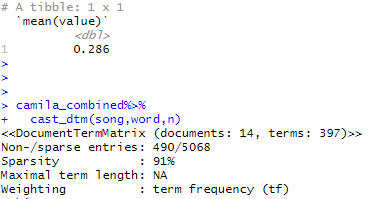
### Output 6.6: Afinn Mean Score



camila\_combined%>%

cast\_dtm(song,word,n)

### Output 6.7: DTM



## Appendix 7: Artist Comparison

combined\_us <-bind\_rows(

mutate(combined\_b, artist = "Billie Eilish"),

mutate(combined\_hal, artist = "Halsey"),

mutate(combined\_s, artist = "Selena Gomez"),

mutate(combined\_har, artist = "Harry Styles"),

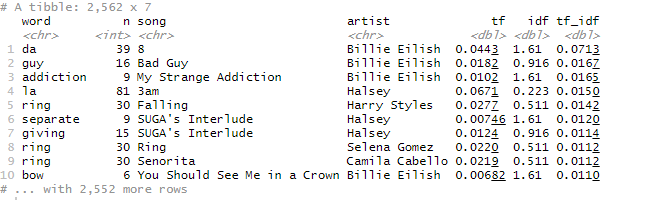
mutate(combined\_c, artist = "Camila Cabello"))

USA\_combined <- combined\_us%>%

bind\_tf\_idf(word,artist,n)%>%

arrange(desc(tf\_idf))

### Output 7.1: TF IDF



### Output 7.2: Correlograms

frequency1 <- combined\_us%>%

mutate(word=str\_extract(word, "[a-z']+")) %>%

count(artist, word) %>%

group\_by(artist) %>%

mutate(proportion = n/sum(n))%>%

select(-n) %>%

spread(artist, proportion) %>%

gather(artist, proportion, `Halsey`,`Harry Styles`,`Selena Gomez`,`Camila Cabello`)

ggplot(frequency1, 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=1.5) +

scale\_x\_log10(labels = percent\_format())+

scale\_y\_log10(labels= percent\_format())+

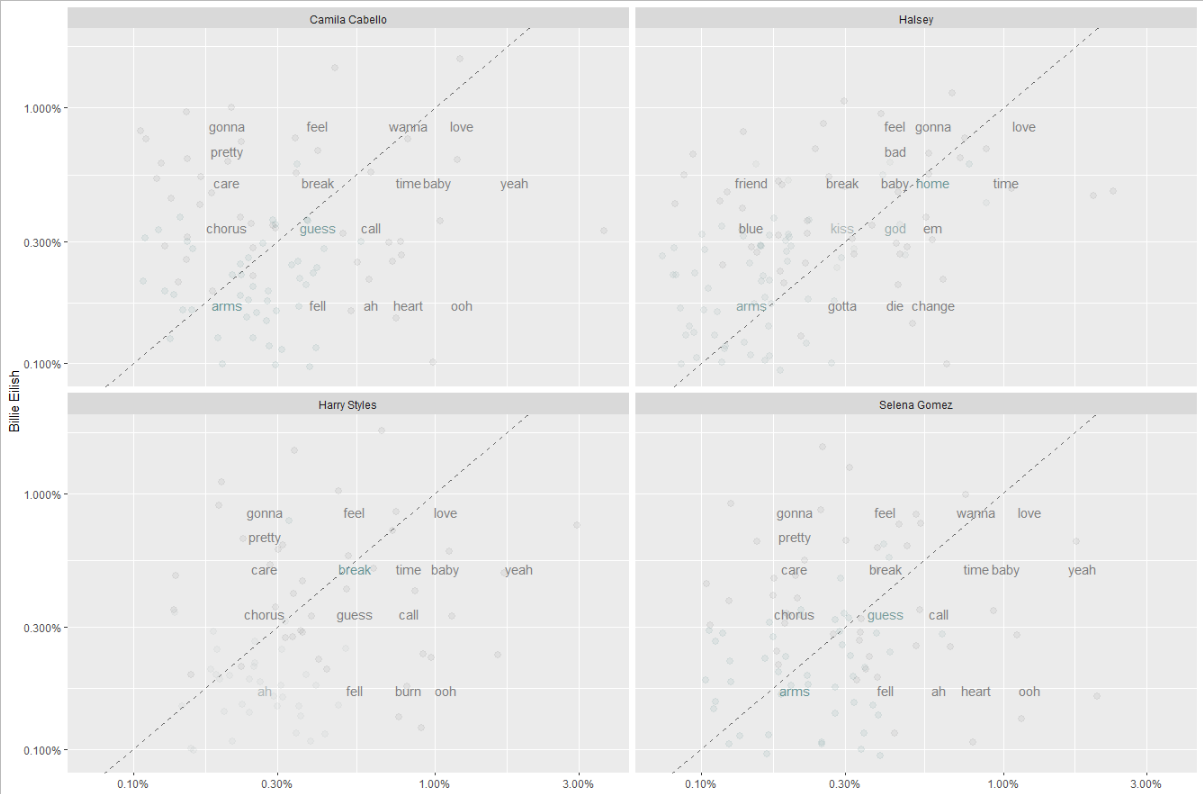
scale\_color\_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+

facet\_wrap(~artist, ncol=2)+

theme(legend.position = "none")+

labs(y= "Billie Eilish", x=NULL)

#### Output 7.2.1: Billie Eilish ~ Others



frequency2 <- combined\_us%>%

mutate(word=str\_extract(word, "[a-z']+")) %>%

count(artist, word) %>%

group\_by(artist) %>%

mutate(proportion = n/sum(n))%>%

select(-n) %>%

spread(artist, proportion) %>%

gather(artist, proportion, `Billie Eilish`,`Harry Styles`,`Selena Gomez`,`Camila Cabello`)

ggplot(frequency2, aes(x=proportion, y=`Halsey`,

color = abs(`Halsey`- 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=1.5) +

scale\_x\_log10(labels = percent\_format())+

scale\_y\_log10(labels= percent\_format())+

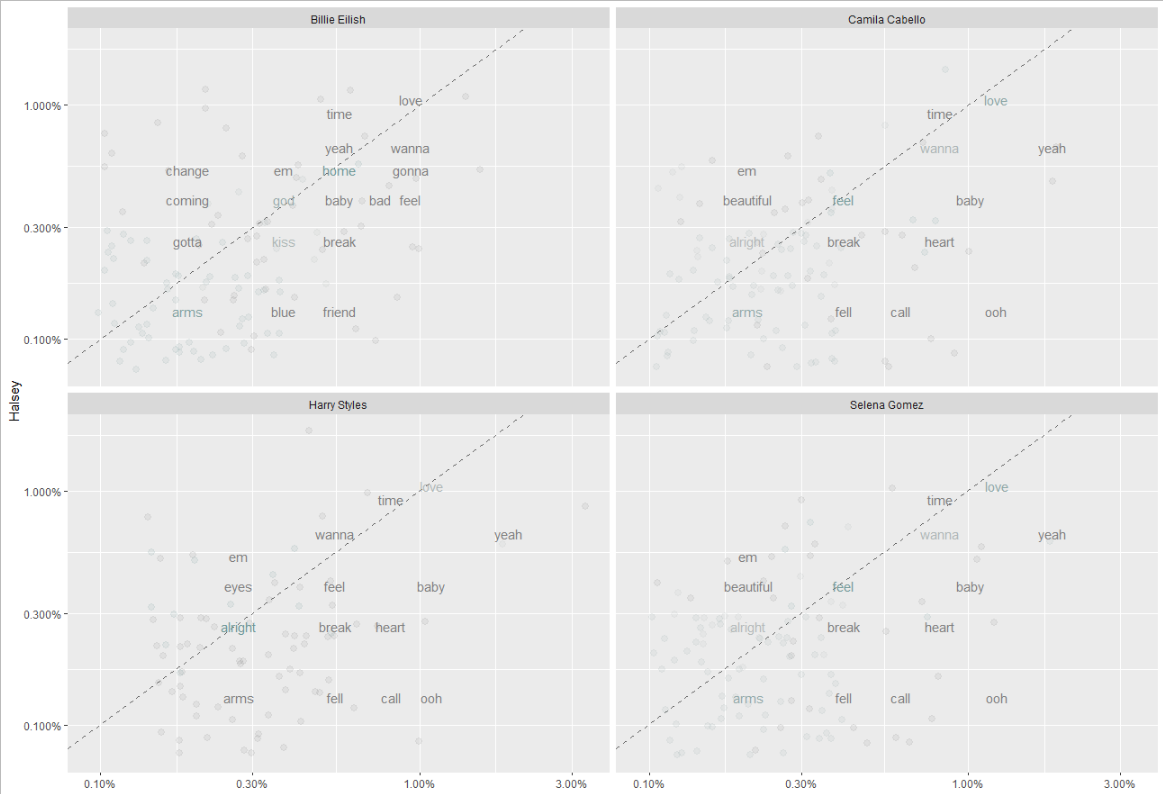
scale\_color\_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+

facet\_wrap(~artist, ncol=2)+

theme(legend.position = "none")+

labs(y= "Halsey", x=NULL)

#### Output 7.2.2: Halsey ~ Others



frequency3 <- combined\_us%>%

mutate(word=str\_extract(word, "[a-z']+")) %>%

count(artist, word) %>%

group\_by(artist) %>%

mutate(proportion = n/sum(n))%>%

select(-n) %>%

spread(artist, proportion) %>%

gather(artist, proportion, `Billie Eilish`,`Halsey`,`Selena Gomez`,`Camila Cabello`)

ggplot(frequency3, aes(x=proportion, y=`Harry Styles`,

color = abs(`Harry Styles`- 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=1.5) +

scale\_x\_log10(labels = percent\_format())+

scale\_y\_log10(labels= percent\_format())+

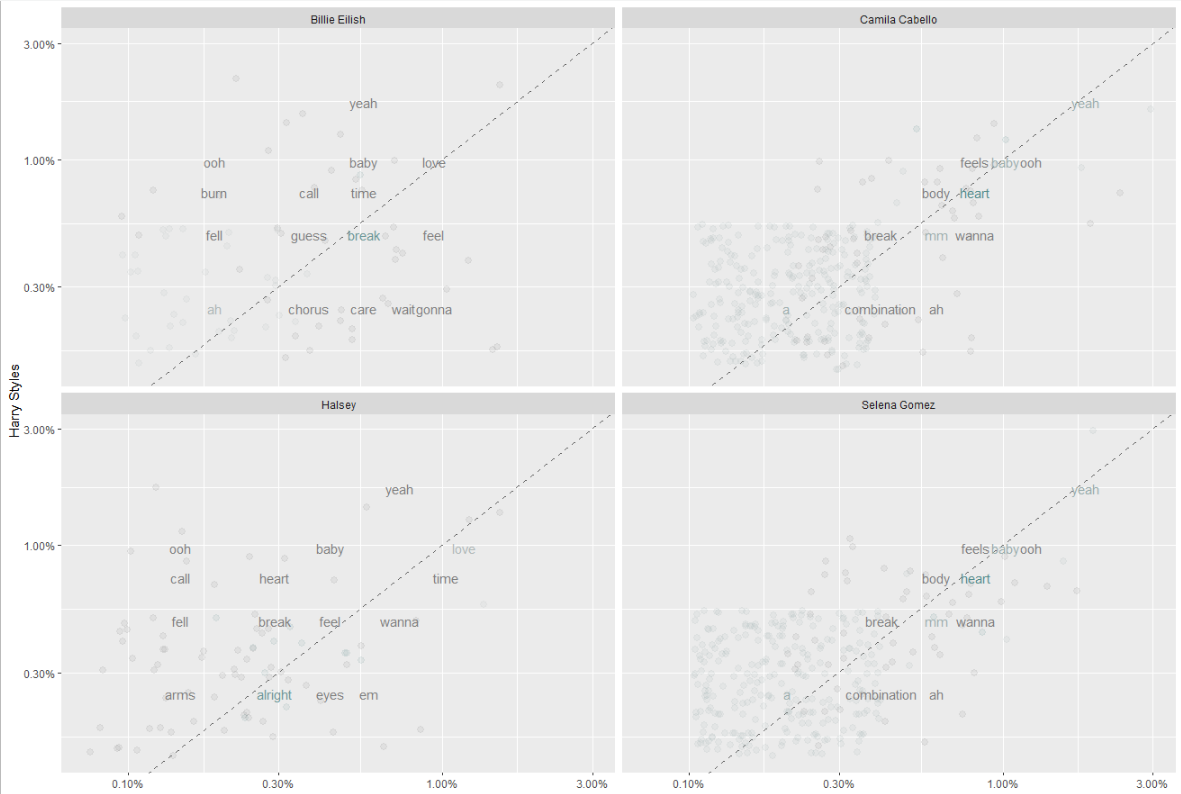
scale\_color\_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+

facet\_wrap(~artist, ncol=2)+

theme(legend.position = "none")+

labs(y= "Harry Styles", x=NULL)

#### Output 7.2.3: Harry Styles ~ Others



frequency4 <- combined\_us%>%

mutate(word=str\_extract(word, "[a-z']+")) %>%

count(artist, word) %>%

group\_by(artist) %>%

mutate(proportion = n/sum(n))%>%

select(-n) %>%

spread(artist, proportion) %>%

gather(artist, proportion, `Billie Eilish`,`Halsey`,`Harry Styles`,`Camila Cabello`)

ggplot(frequency4, aes(x=proportion, y=`Selena Gomez`,

color = abs(`Selena Gomez`- 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=1.5) +

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= "Selena Gomez", x=NULL)

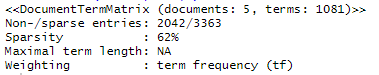
#### Output 7.2.4: Selena Gomez ~ Others



USA\_combined%>%

cast\_dtm(artist,word,n)

### Output 7.2: DTM



USA\_combined %>%

arrange(desc(tf\_idf)) %>%

mutate(word=factor(word, levels=rev(unique(word)))) %>%

group\_by(song) %>%

top\_n(4) %>%

ungroup %>%

ggplot(aes(word, tf\_idf, fill=song))+

geom\_col(show.legend=FALSE)+

labs(x=NULL, y="tf-idf")+

facet\_wrap(~song, ncol=2, scales="free")+

coord\_flip()

### Output 7.3: TF IDF by Artist



USA\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

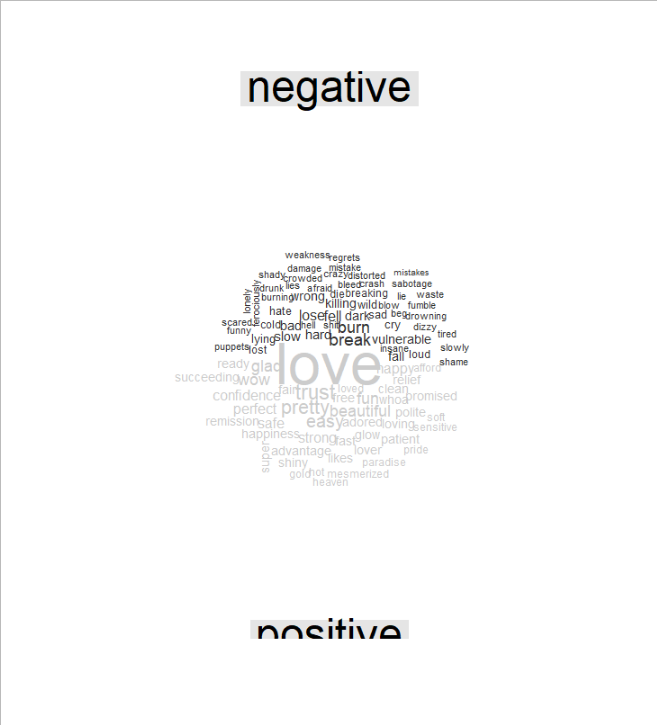
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 7.4: Bing Sentiment Word Cloud



USA\_combined %>%

inner\_join(get\_sentiments("nrc")) %>%

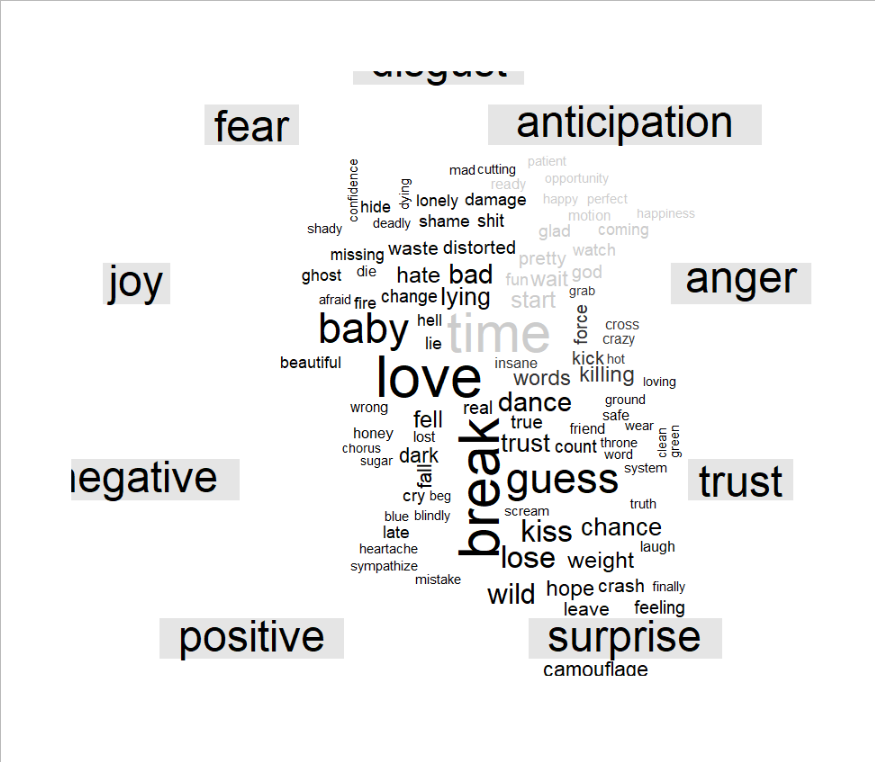
count(word, sentiment, sort=TRUE) %>%

acast(word ~sentiment, value.var="n", fill=0) %>%

comparison.cloud(colors = c("grey20", "gray80"),

max.words=100)

### Output 7.5: NRC Sentiment Word Cloud



bing\_counts <- USA\_combined %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort=T) %>%

ungroup() %>%

group\_by(sentiment) %>%

top\_n(10) %>%

ungroup() %>%

mutate(word=reorder(word, n)) %>%

ggplot(aes(word, n, fill=sentiment)) +

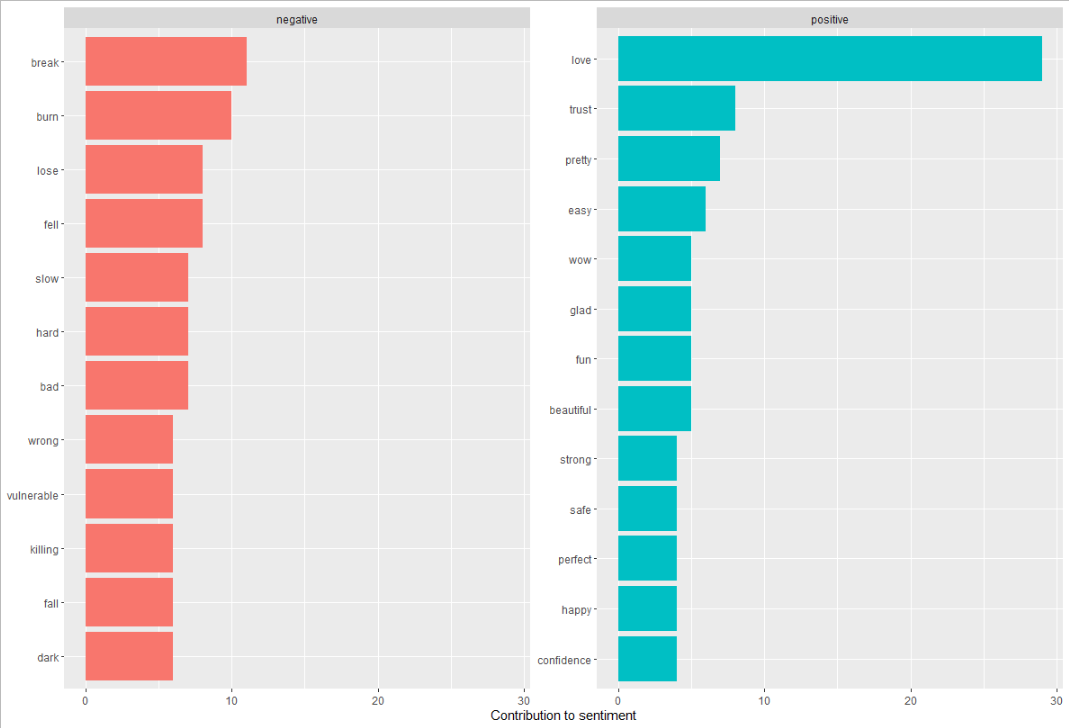
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y")+

labs(y="Contribution to sentiment", x=NULL)+

coord\_flip()

### Output 7.6: Contribution to Sentiment



afinn\_sent <- USA\_combined %>%

inner\_join(get\_sentiments("afinn"))%>%

summarise(mean(value))

afinn\_sent

### Output 7.7: Afinn Mean Score

