# Econ 106

Lecture 14

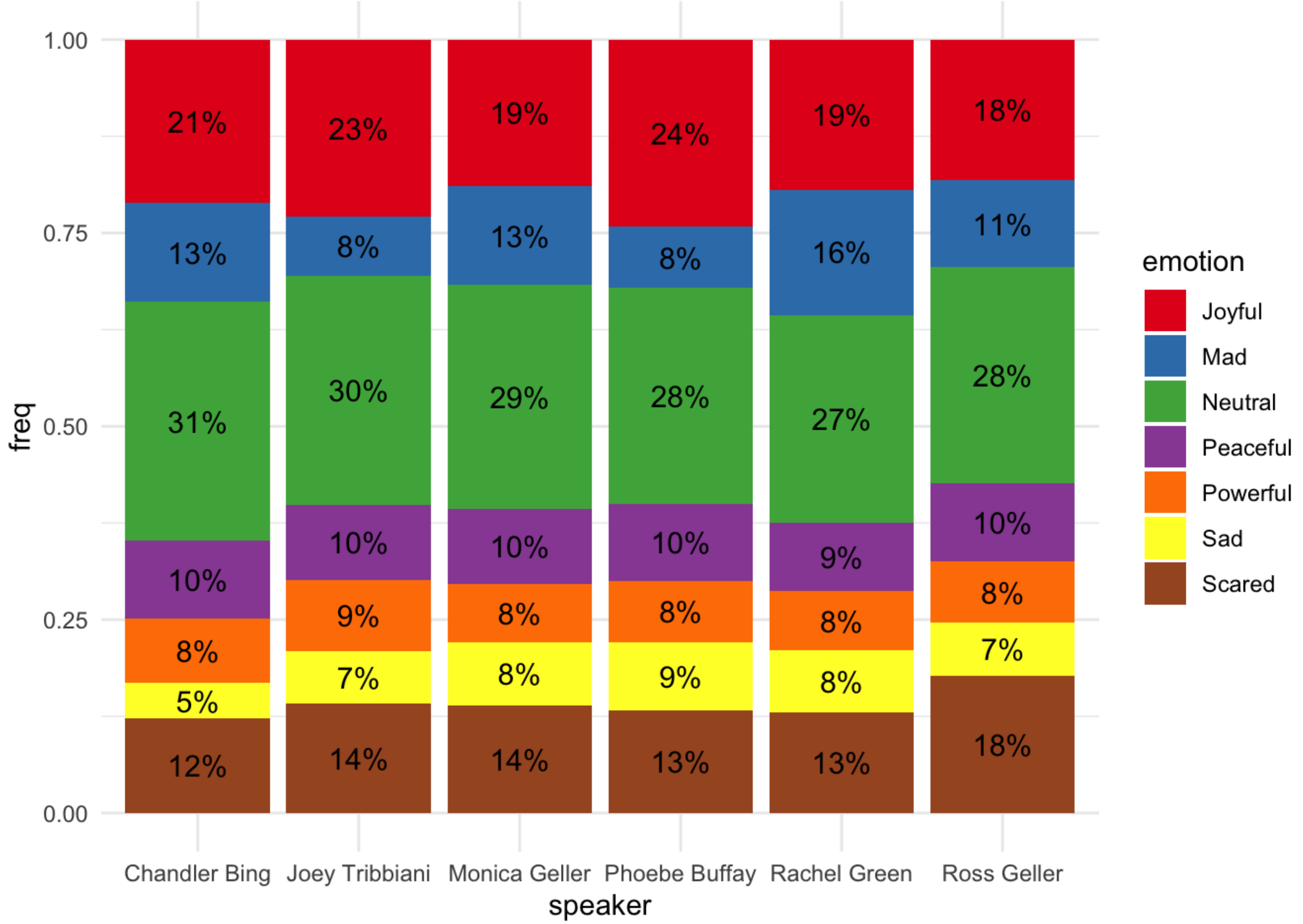
slides derived from:

https://www.tidytextmining.com/tidytext

# Reminders

- Lab #4 is due Sunday 11:59pm
- Please submit MS #2 by 11:59pm tonight if you haven’t already done so
- Next week is Thanksgiving week:
    - in person lecture on Monday
    - Zoom lecture on Wednesday (or recording, I’ll let you know what I decide)

# #tidytuesday

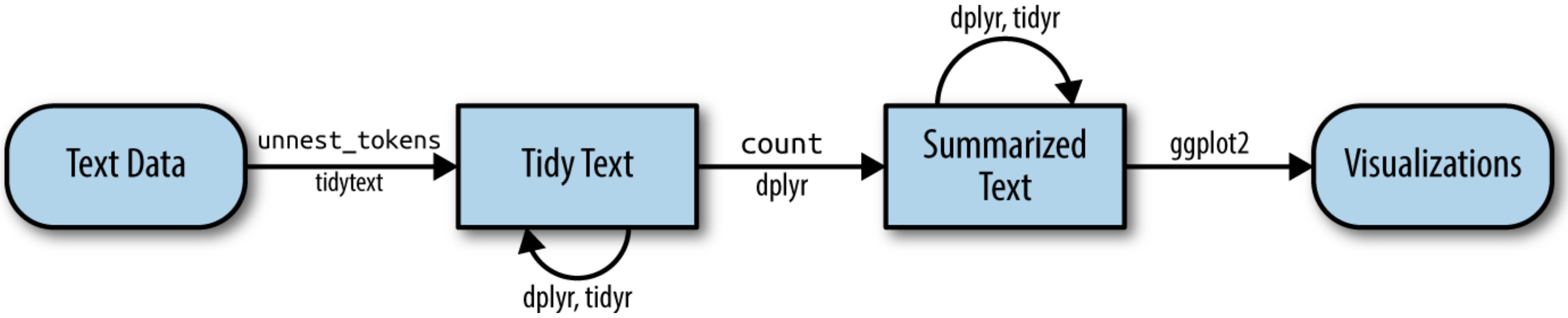


https://rpubs.com/Pyll/Friendship_is_magic

# Outline

- Text as Data:
  - sentiment analysis
  - word clouds

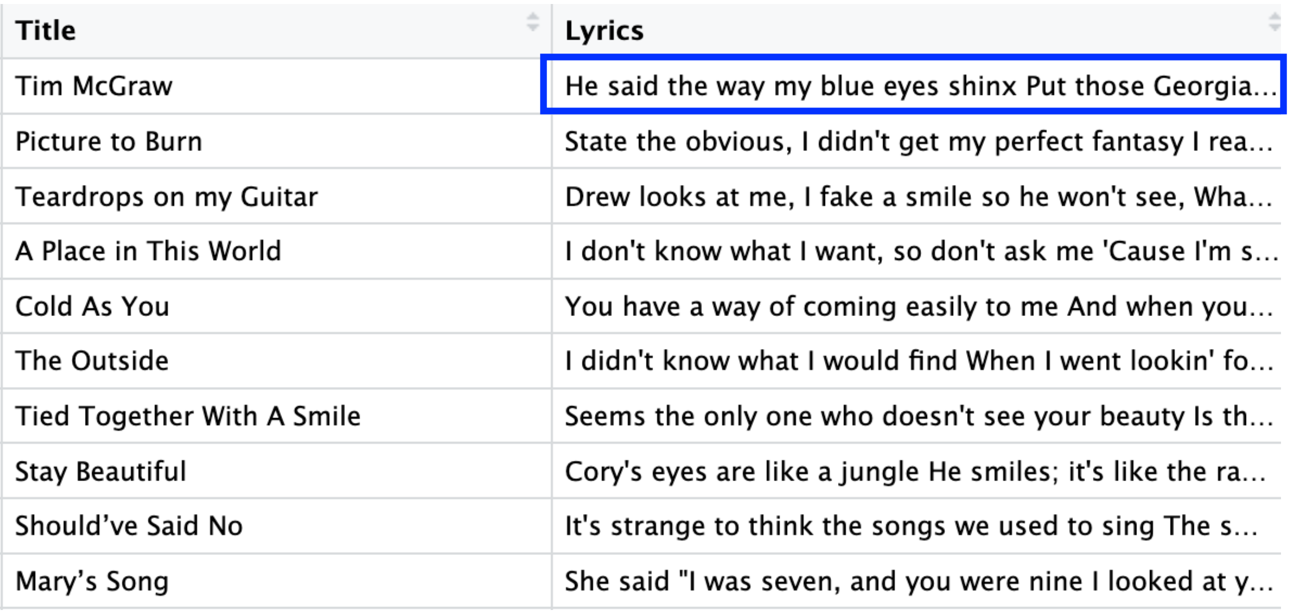
# Overview of Text Analysis in R



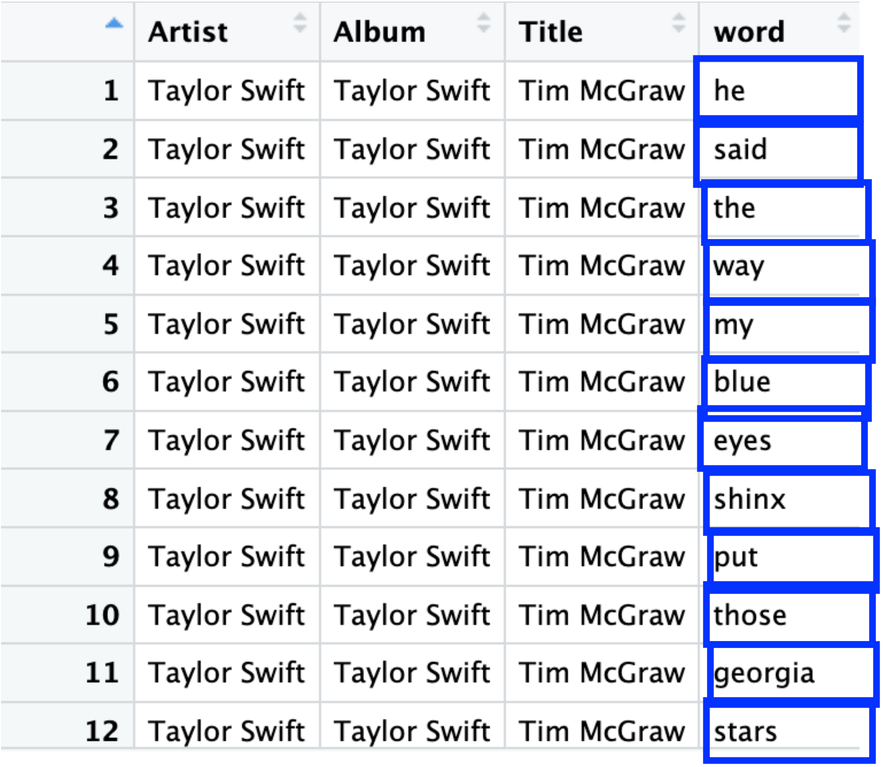
https://www.tidytextmining.com/tidytext

# Step 1: Tokenize

```
tidy_lyrics <- taylor_swift_lyrics %>%
  unnest_tokens(output=word, input=Lyrics)
```

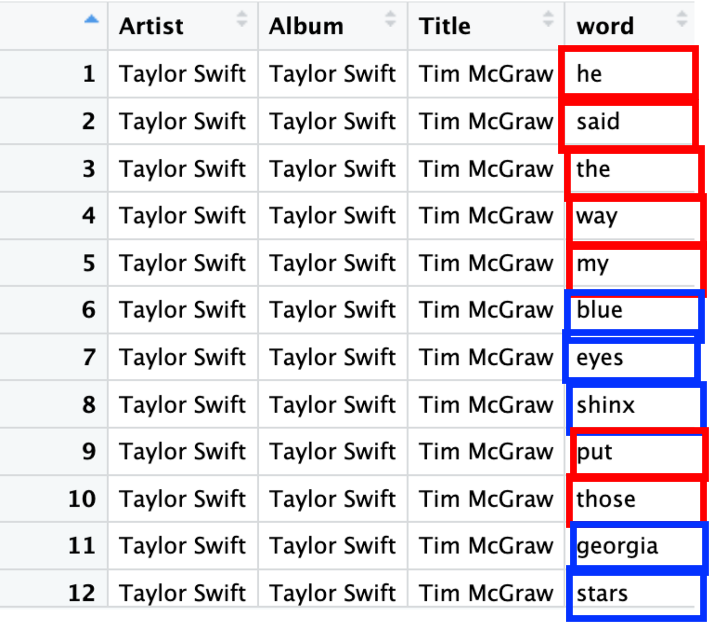


| Title | Lyrics |
|---|---|
| Tim McGraw | He said the way my blue eyes shinx Put those Georgia… |
| Picture to Burn | State the obvious, I didn't get my perfect fantasy I rea… |
| Teardrops on my Guitar | Drew looks at me, I fake a smile so he won't see, Wha… |
| A Place in This World | I don't know what I want, so don't ask me 'Cause I'm s… |
| Cold As You | You have a way of coming easily to me And when you… |
| The Outside | I didn't know what I would find When I went lookin' fo… |
| Tied Together With A Smile | Seems the only one who doesn't see your beauty Is th… |
| Stay Beautiful | Cory's eyes are like a jungle He smiles; it's like the ra… |
| Should've Said No | It's strange to think the songs we used to sing The s… |
| Mary's Song | She said "I was seven, and you were nine I looked at y… |



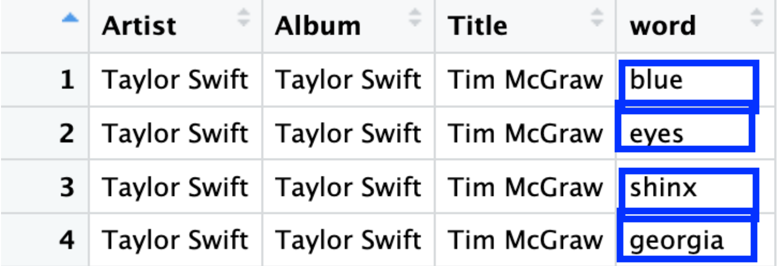
|  | Artist | Album | Title | word |
|---|---|---|---|---|
| 1 | Taylor Swift | Taylor Swift | Tim McGraw | he |
| 2 | Taylor Swift | Taylor Swift | Tim McGraw | said |
| 3 | Taylor Swift | Taylor Swift | Tim McGraw | the |
| 4 | Taylor Swift | Taylor Swift | Tim McGraw | way |
| 5 | Taylor Swift | Taylor Swift | Tim McGraw | my |
| 6 | Taylor Swift | Taylor Swift | Tim McGraw | blue |
| 7 | Taylor Swift | Taylor Swift | Tim McGraw | eyes |
| 8 | Taylor Swift | Taylor Swift | Tim McGraw | shinx |
| 9 | Taylor Swift | Taylor Swift | Tim McGraw | put |
| 10 | Taylor Swift | Taylor Swift | Tim McGraw | those |
| 11 | Taylor Swift | Taylor Swift | Tim McGraw | georgia |
| 12 | Taylor Swift | Taylor Swift | Tim McGraw | stars |

# Step 2: Remove Stop Words



| | Artist | Album | Title | word |
|---|---|---|---|---|
| 1 | Taylor Swift | Taylor Swift | Tim McGraw | he |
| 2 | Taylor Swift | Taylor Swift | Tim McGraw | said |
| 3 | Taylor Swift | Taylor Swift | Tim McGraw | the |
| 4 | Taylor Swift | Taylor Swift | Tim McGraw | way |
| 5 | Taylor Swift | Taylor Swift | Tim McGraw | my |
| 6 | Taylor Swift | Taylor Swift | Tim McGraw | blue |
| 7 | Taylor Swift | Taylor Swift | Tim McGraw | eyes |
| 8 | Taylor Swift | Taylor Swift | Tim McGraw | shinx |
| 9 | Taylor Swift | Taylor Swift | Tim McGraw | put |
| 10 | Taylor Swift | Taylor Swift | Tim McGraw | those |
| 11 | Taylor Swift | Taylor Swift | Tim McGraw | georgia |
| 12 | Taylor Swift | Taylor Swift | Tim McGraw | stars |

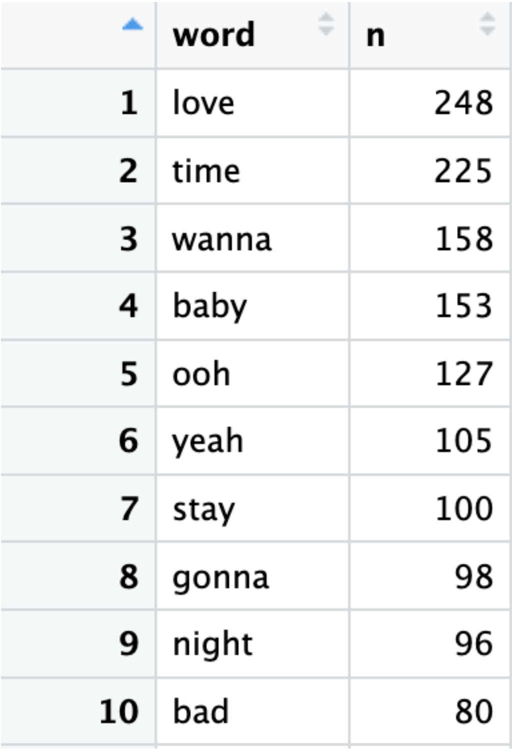
```
tidy_lyrics_no_stop <-
anti_join(x=tidy_lyrics, y=stop_words)
```



| | Artist | Album | Title | word |
|---|---|---|---|---|
| 1 | Taylor Swift | Taylor Swift | Tim McGraw | blue |
| 2 | Taylor Swift | Taylor Swift | Tim McGraw | eyes |
| 3 | Taylor Swift | Taylor Swift | Tim McGraw | shinx |
| 4 | Taylor Swift | Taylor Swift | Tim McGraw | georgia |

# Step 3: Summarize (Count)

```
tidy_lyrics_top_ten<-
tidy_lyrics_no_stop %>%
          count(word) %>%
           arrange(desc( n)) %>%
           slice_head(n=10)
```

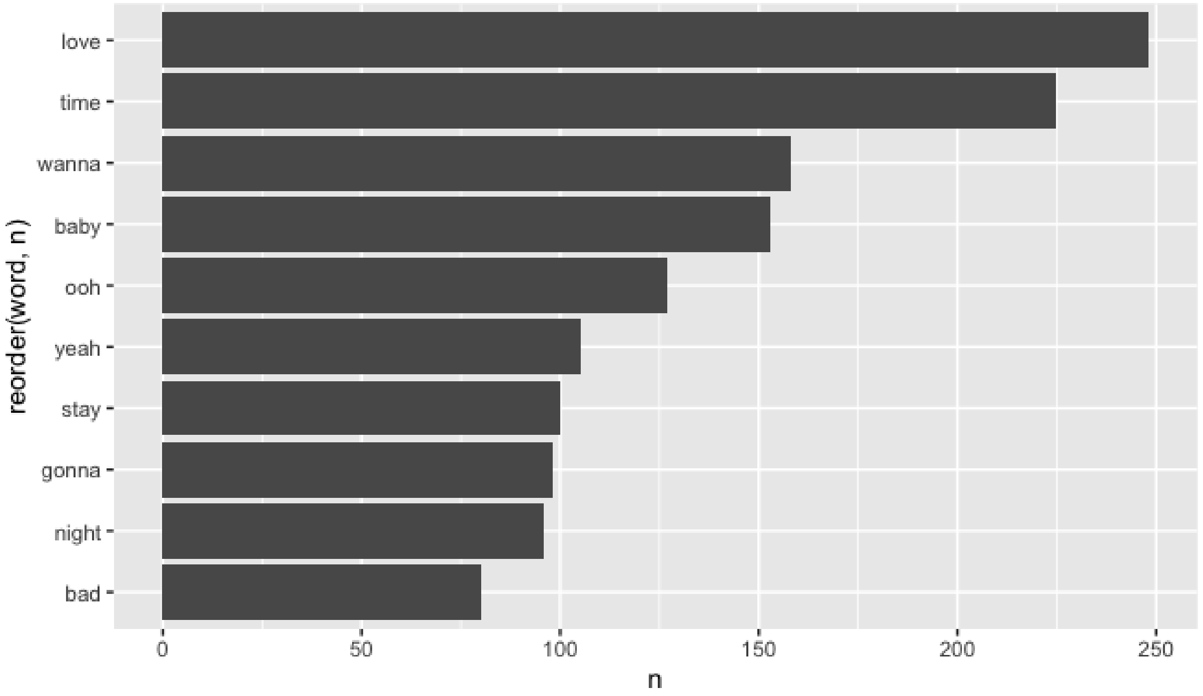


|  | word | n |
|---|---|---|
| 1 | love | 248 |
| 2 | time | 225 |
| 3 | wanna | 158 |
| 4 | baby | 153 |
| 5 | ooh | 127 |
| 6 | yeah | 105 |
| 7 | stay | 100 |
| 8 | gonna | 98 |
| 9 | night | 96 |
| 10 | bad | 80 |

# Step 4: Visualize

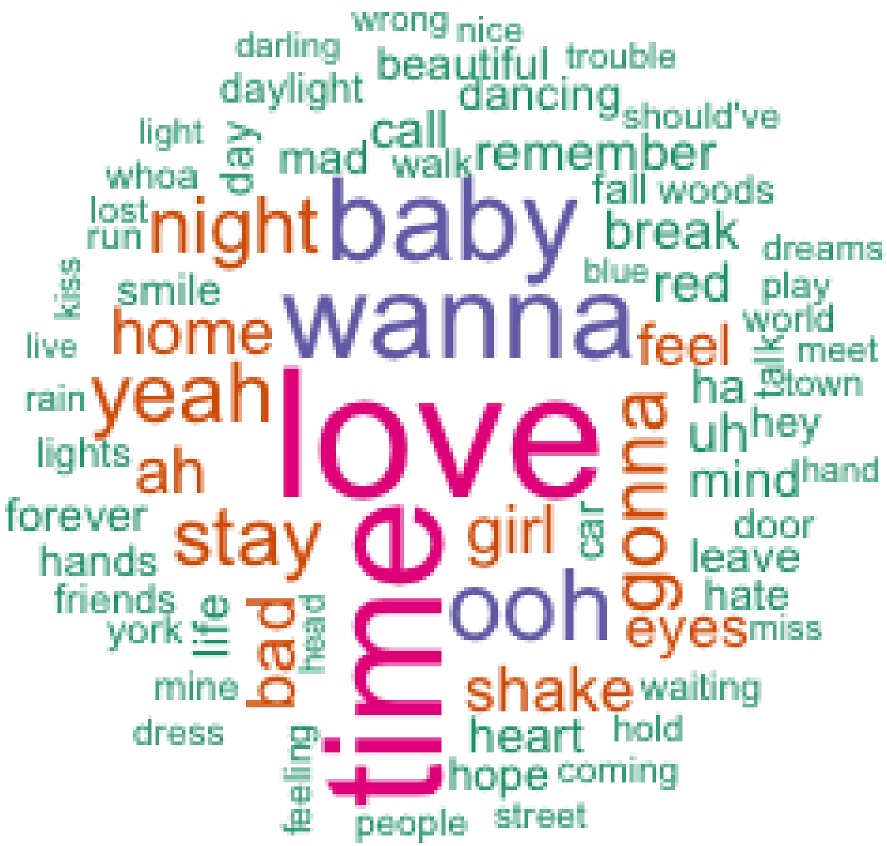
https://pollev.com/vsovero

```
ggplot(data=tidy_lyrics_top_ten,
mapping=aes(x= reorder(word, n), y=n))+
  geom_col() +
coord_flip()
```



# Word Clouds

- Word clouds are another method of visualizing word frequencies
- the size and color of the word denote relative frequency



# Setup for Word Clouds

- Create a frequency table of word counts using **count**()
- sort the counts in descending order using **arrange**(**desc**())
- keep the top 75 words for the word cloud using **slice_head**()

```
tidy_lyrics_no_stop %>%
              count(word) %>%
               arrange(desc( n)) %>%
              slice_head(n=75)
```

# Word Clouds

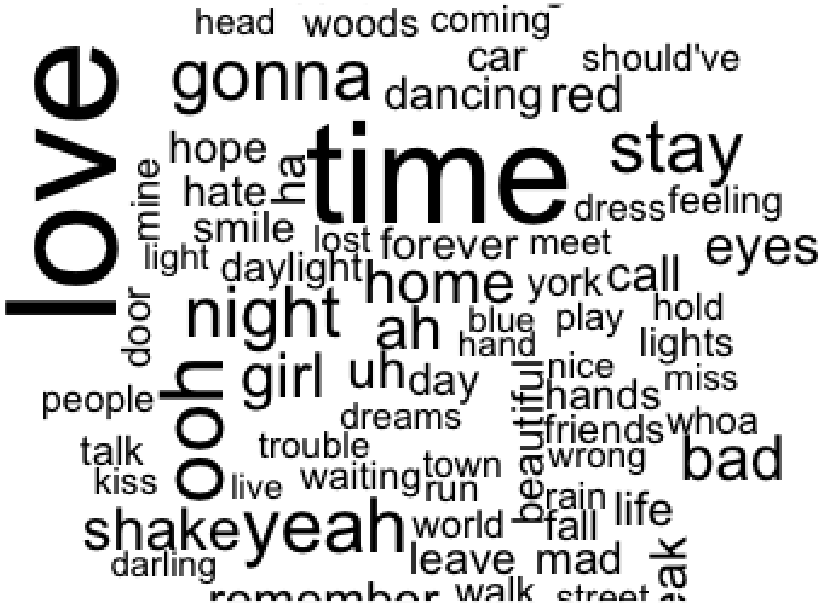
- we pass the frequency table over to the wordcloud() function using pipes and with()
- wordcloud() arguments:
  - words: name of the variable that contains the words you want to visualize
  - freq: name of the variable that records the frequency of each word

```
tidy_lyrics_no_stop %>%
              count(word) %>%
               arrange(desc( n)) %>%
              slice_head(n=75) %>%
with(wordcloud(words =word, freq=n))
```

# Word Clouds

- Output: world cloud where size represents relative frequency
- Note: default is for random subset of words to be visualized when there isn't enough space for all of the words

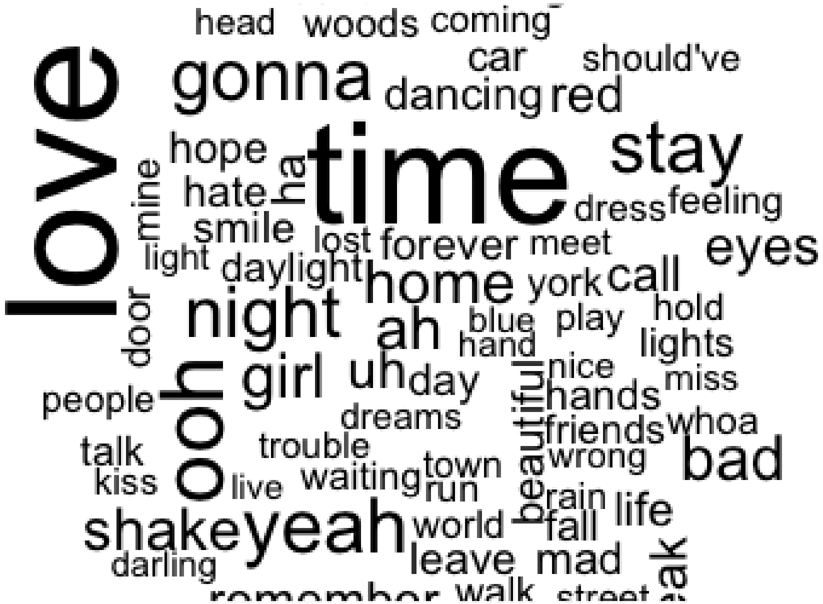
```
tidy_lyrics_no_stop %>%
                count(word) %>%
                 arrange(desc( n)) %>%
                slice_head(n=75) %>%
with(wordcloud(words =word, freq=n))
```



# Word Clouds

- we can turn off the random selection of words with the random.order argument
- layout is still random

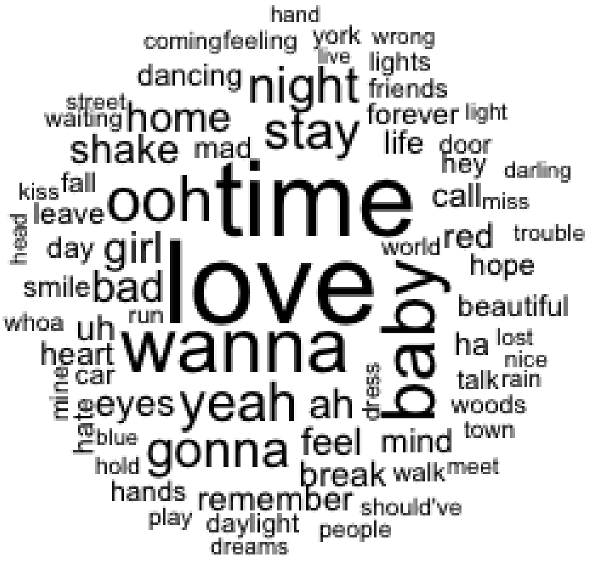
```
tidy_lyrics_no_stop %>%
                count(word) %>%
                 arrange(desc( n)) %>%
                slice_head(n=75) %>%
with(wordcloud(words =word, freq=n,
                 random.order=FALSE))
```



# Word Cloud Adjustments: word scale

- We can adjust the relative scale of the word sizes by adding the scale argument

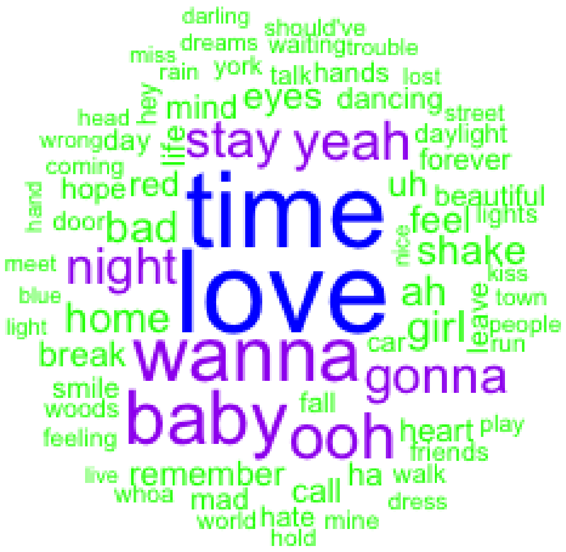
```
tidy_lyrics_no_stop %>%
              count(word) %>%
               arrange(desc( n)) %>%
              slice_head(n=75) %>%
with(wordcloud(words =word, freq=n,
 random.order=FALSE, scale= c(3, 0.25)))
```



# Word Cloud Adjustments: add color

- We can manually choose the colors using the colors argument
- color order listed will be assigned to words from lowest to highest frequency
- list of color names available here: [http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf](http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf)

```
tidy_lyrics_no_stop %>%
              count(word) %>%
               arrange(desc( n)) %>%
              slice_head(n=75) %>%
with(wordcloud(words =word, freq=n,
 random.order=FALSE, scale= c(3, 0.25),
 colors=c('green', 'purple', 'blue')))
```



# Exercise

https://pollev.com/vsovero

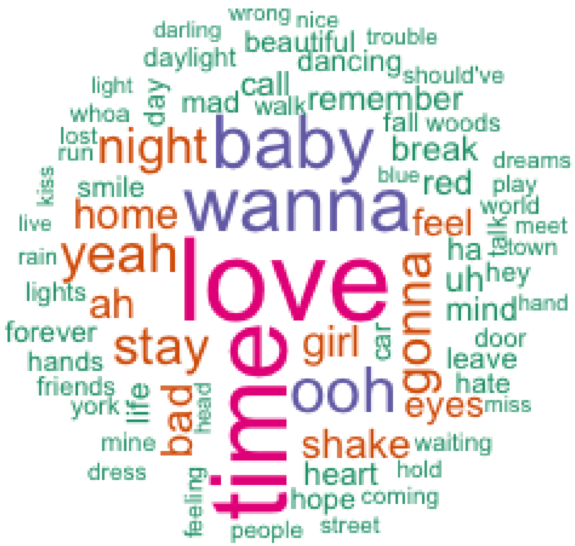
- tokenize the comments in the rate my professors data frame
- remove stop words, plus ‘professor’, ‘class’, ‘students’, and ‘teacher’
- use the following color scale: purple, green, yellow, red

# Word Cloud Adjustments: add color

- We can choose a color palette using brewer.pal() from the RColorBrewer package
- Arguments:
  - number of colors (4)
  - name of palette (Dark2)

- Palette names available here: [https://r-graph-gallery.com/38-rcolorbrewers-palettes.html](https://r-graph-gallery.com/38-rcolorbrewers-palettes.html)

```
tidy_lyrics_no_stop %>%
              count(word) %>%
               arrange(desc( n)) %>%
              slice_head(n=75) %>%
with(wordcloud(words =word, freq=n,
 random.order=FALSE, scale= c(3, 0.25),
 colors= brewer.pal(4,"Dark2")))
```



# Next: Sentiment Analysis

- What else can we do besides word/stem frequency?
- When you read a text, you can infer whether it expresses a positive or negative emotion:
  - ”I’m so happy!”
  - “I feel sad.”
- We can use the tools of text mining to analyze the emotional content of text

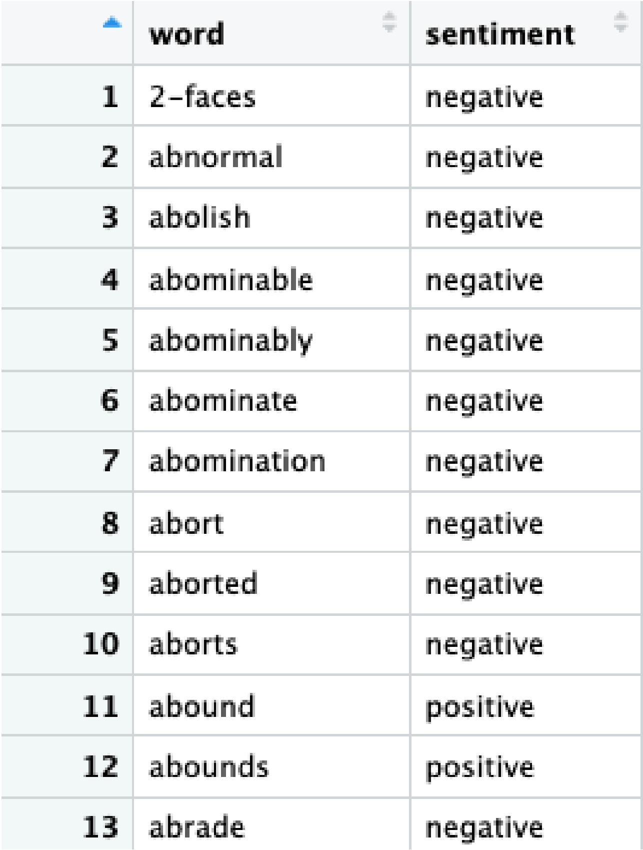
# Sentiment Analysis

- Our approach:
  - tokenize into single words
  - evaluate sentiment scores of individual words
  - summarize: add up the individual sentiment scores for each word in the text
  - visualize

# Sentiment Lexicons

- A **sentiment lexicon** (dictionary) contain many English words and their associated sentiment
- You can load sentiment lexicons using the **get_sentiments**() function from the tidytext package
- the **bing** lexicon assigns words as negative or positive

```
bing_lexicon<-get_sentiments(lexicon="bing")
```



| | word | sentiment |
|---|---|---|
| 1 | 2-faces | negative |
| 2 | abnormal | negative |
| 3 | abolish | negative |
| 4 | abominable | negative |
| 5 | abominably | negative |
| 6 | abominate | negative |
| 7 | abomination | negative |
| 8 | abort | negative |
| 9 | aborted | negative |
| 10 | aborts | negative |
| 11 | abound | positive |
| 12 | abounds | positive |
| 13 | abrade | negative |

# Downside of single word sentiment analysis

- We will misclassify a sentiment when the word is paired with a negative:
  - “I’m not happy!”
  - “I don’t feel sad”

# Ok, let's analyze Taylor Swifts Lyrics

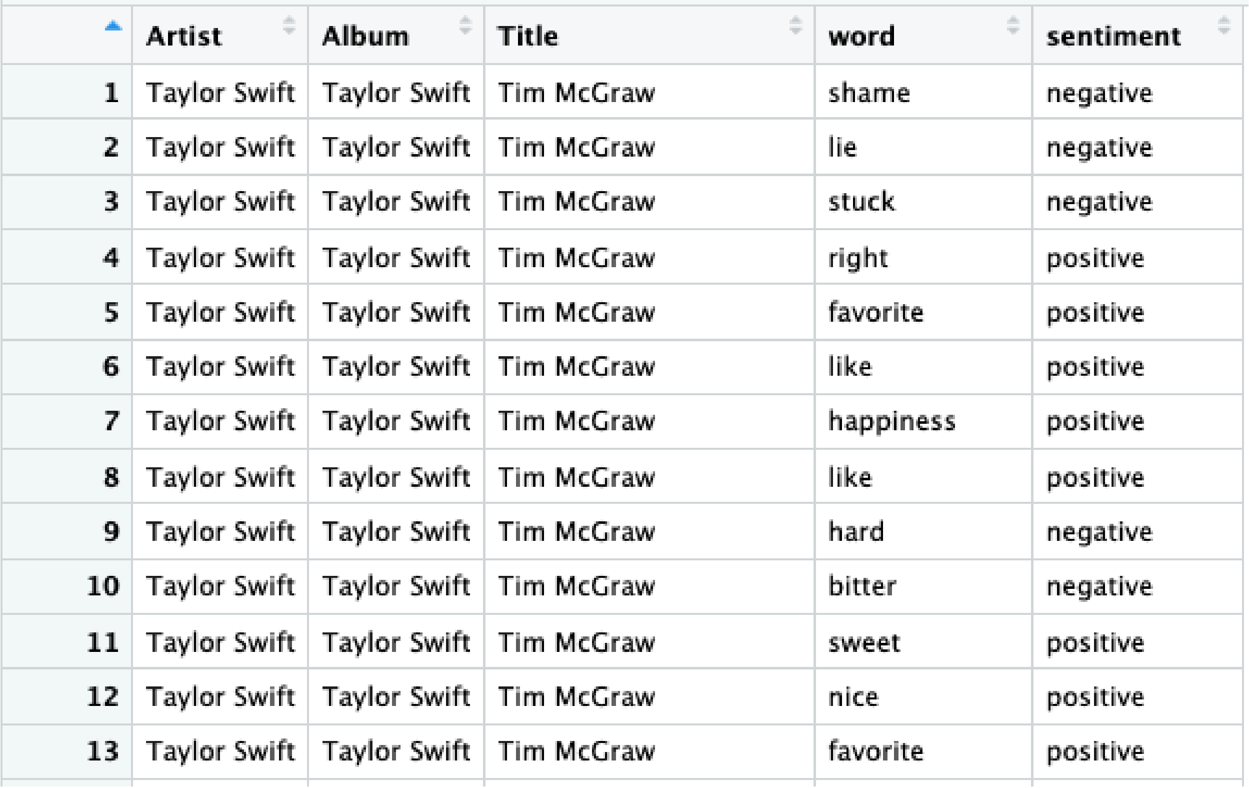
- We already have the tokenized version of her lyrics
- We are going to perform an inner join with the bing lexicon (keep words in Taylor Swift lyrics that can be matched to a sentiment)

```
tidy_lyrics_bing<- inner_join(x=tidy_lyrics, y=bing_lexicon)
```

# Bing Sentiments

```
tidy_lyrics_bing<- inner_join(x=tidy_lyrics, y=bing_lexicon)
```

- Now we have a sentiment column that we can use to analyze Taylor Swift’s lyrics



| | Artist | Album | Title | word | sentiment |
|---|---|---|---|---|---|
| 1 | Taylor Swift | Taylor Swift | Tim McGraw | shame | negative |
| 2 | Taylor Swift | Taylor Swift | Tim McGraw | lie | negative |
| 3 | Taylor Swift | Taylor Swift | Tim McGraw | stuck | negative |
| 4 | Taylor Swift | Taylor Swift | Tim McGraw | right | positive |
| 5 | Taylor Swift | Taylor Swift | Tim McGraw | favorite | positive |
| 6 | Taylor Swift | Taylor Swift | Tim McGraw | like | positive |
| 7 | Taylor Swift | Taylor Swift | Tim McGraw | happiness | positive |
| 8 | Taylor Swift | Taylor Swift | Tim McGraw | like | positive |
| 9 | Taylor Swift | Taylor Swift | Tim McGraw | hard | negative |
| 10 | Taylor Swift | Taylor Swift | Tim McGraw | bitter | negative |
| 11 | Taylor Swift | Taylor Swift | Tim McGraw | sweet | positive |
| 12 | Taylor Swift | Taylor Swift | Tim McGraw | nice | positive |
| 13 | Taylor Swift | Taylor Swift | Tim McGraw | favorite | positive |

# Top 10 Positive Words

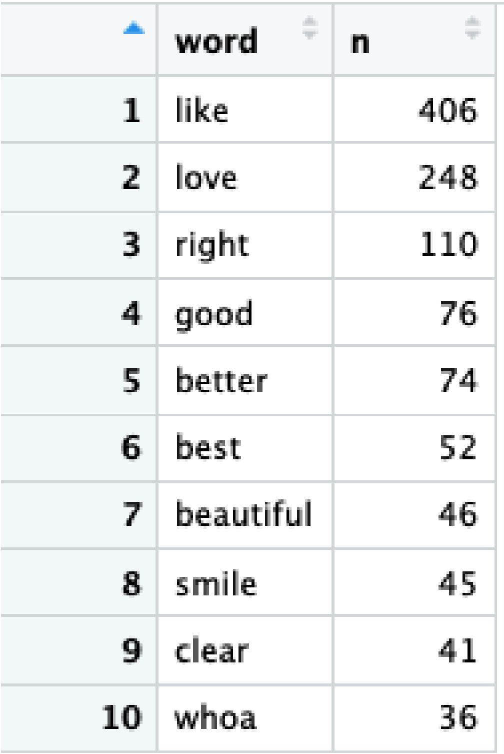
- Filter for words with a positive sentiment
- count the frequency of each positive word
- sort in descending order
- keep the top 10

```
bing_positive_count <- tidy_lyrics_bing%>%
          filter(sentiment=="positive") %>%
          count(word) %>%
           arrange(desc( n)) %>%
           slice_head(n=10)
```

# Top 10 Positive Words

- Because we didn’t filter out stop words, ”like” has surpassed “love” on our top ten list

```
bing_positive_count <- tidy_lyrics_bing%>%
          filter(sentiment=="positive") %>%
          count(word) %>%
           arrange(desc( n)) %>%
           slice_head(n=10)
```



|  | word | n |
|---|---|---|
| 1 | like | 406 |
| 2 | love | 248 |
| 3 | right | 110 |
| 4 | good | 76 |
| 5 | better | 74 |
| 6 | best | 52 |
| 7 | beautiful | 46 |
| 8 | smile | 45 |
| 9 | clear | 41 |
| 10 | whoa | 36 |

# Class Exercise

https://pollev.com/vsovero

- use the tidy rmp data (don’t remove stop words)
- join with the bing lexicon, keep matches
- Make a list of the top 10 negative words used in the rmp comments

# Top 10 words with sentiment

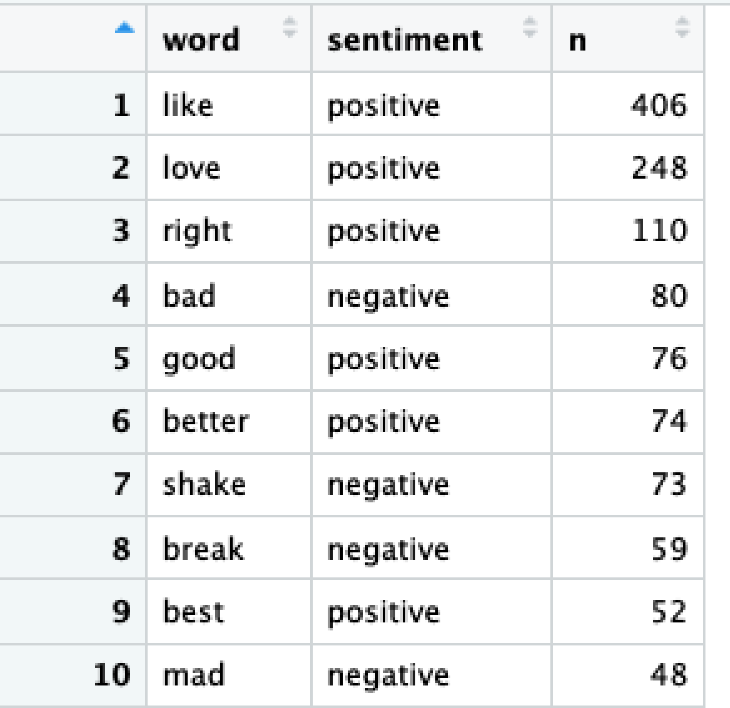
- Of the words that have a positive or negative sentiment, what are the most frequent in Taylor Swift’s lyrics?
- To keep the sentiment information in our frequency table, we have to add it as an argument in the **count**() function

```
bing_count <- tidy_lyrics_bing%>%
              count(word, sentiment) %>%
               arrange(desc( n)) %>%
               slice_head(n=10)
```

# Top 10 words with sentiment

- I guess “shake it off” is a negative sentiment?

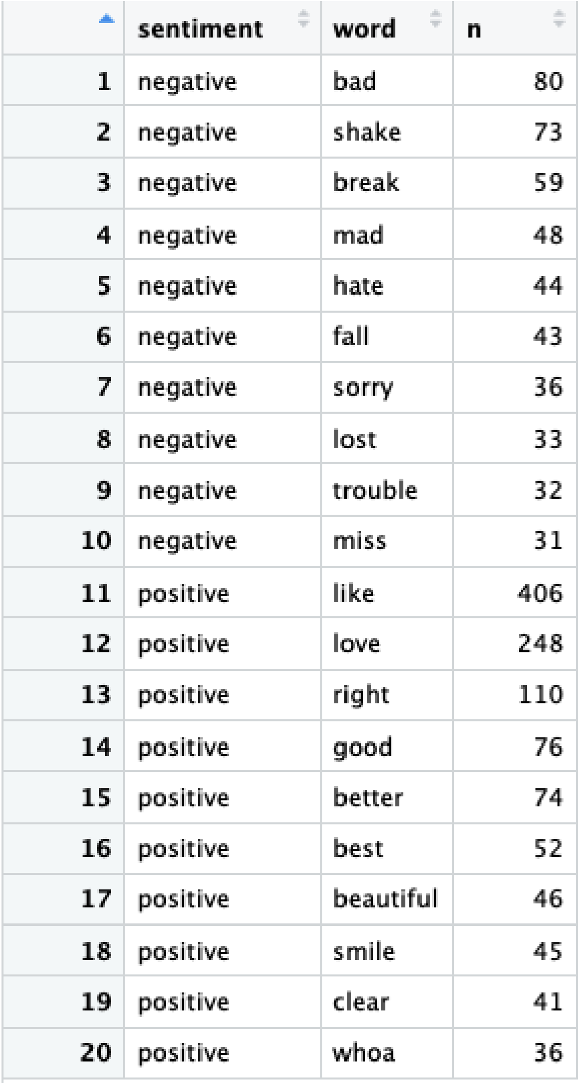
```
bing_word_count <- tidy_lyrics_bing%>%
          count(word, sentiment) %>%
           arrange(desc( n)) %>%
           slice_head(n=10)
```



| | word | sentiment | n |
|---|---|---|---|
| 1 | like | positive | 406 |
| 2 | love | positive | 248 |
| 3 | right | positive | 110 |
| 4 | bad | negative | 80 |
| 5 | good | positive | 76 |
| 6 | better | positive | 74 |
| 7 | shake | negative | 73 |
| 8 | break | negative | 59 |
| 9 | best | positive | 52 |
| 10 | mad | negative | 48 |

# Top 10 words for each sentiment (single table)

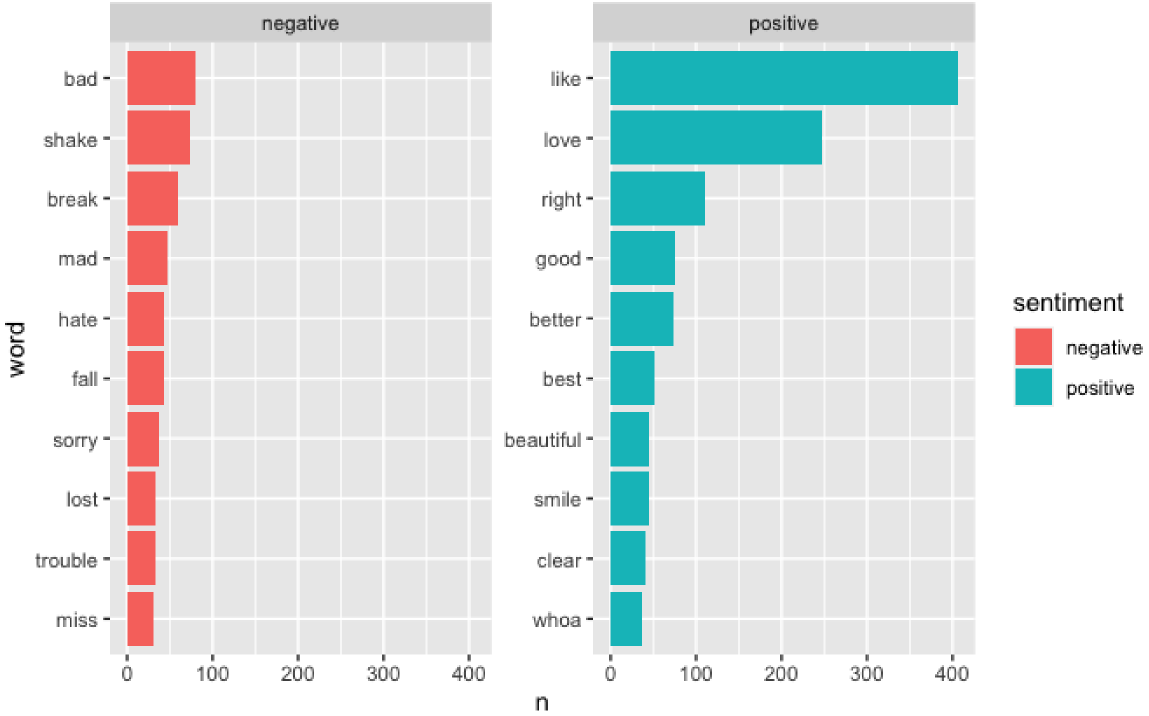
```
bing_word_count <- tidy_lyrics_bing%>%
        group_by(sentiment) %>%
        count(word, sentiment) %>%
        arrange(desc( n)) %>%
        slice_head(n=10)
```



|  | sentiment | word | n |
|---|---|---|---|
| 1 | negative | bad | 80 |
| 2 | negative | shake | 73 |
| 3 | negative | break | 59 |
| 4 | negative | mad | 48 |
| 5 | negative | hate | 44 |
| 6 | negative | fall | 43 |
| 7 | negative | sorry | 36 |
| 8 | negative | lost | 33 |
| 9 | negative | trouble | 32 |
| 10 | negative | miss | 31 |
| 11 | positive | like | 406 |
| 12 | positive | love | 248 |
| 13 | positive | right | 110 |
| 14 | positive | good | 76 |
| 15 | positive | better | 74 |
| 16 | positive | best | 52 |
| 17 | positive | beautiful | 46 |
| 18 | positive | smile | 45 |
| 19 | positive | clear | 41 |
| 20 | positive | whoa | 36 |

# Top 10 words for each sentiment

```
ggplot(mapping=aes(x=n, y=word))+
        geom_col(aes(fill=sentiment))+
        facet_wrap (~sentiment, scales = "free_y")
```



# Class Exercise

- Make a top 10 list of the most frequent words with sentiment for the rmp comments. How many are positive?

# Top sentiments

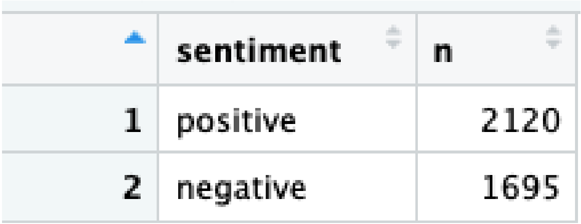
- I’m going to take a wild guess here, but Taylor Swift probably uses more words with positive sentiments versus negative sentiments
- Let’s check: we will count the frequency of the sentiment

```
bing_sentiment_count <- tidy_lyrics_bing%>%
         count(sentiment) %>%
          arrange(desc( n))
```

# Top sentiments

- More words with positive sentiments, but still a surprisingly number of negative words

```
bing_sentiment_count <- tidy_lyrics_bing%>%
        count(sentiment) %>%
        arrange(desc( n))
```

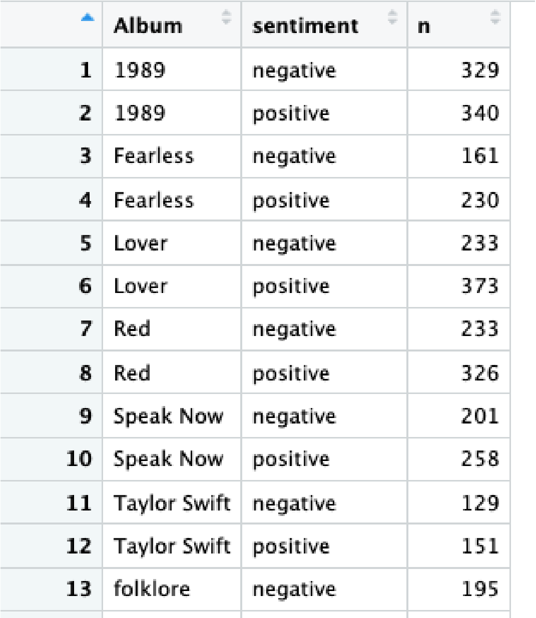


|   | sentiment | n |
|---|---|---|
| 1 | positive | 2120 |
| 2 | negative | 1695 |

# Sentiment by Album

- We can also count the number of positive and negative sentiment words by Album

```
bing_album_sentiment <- tidy_lyrics_bing%>%
  group_by(Album)%>%
  count(sentiment) %>%
  arrange(Album, sentiment)
```



| | Album | sentiment | n |
|---|---|---|---|
| 1 | 1989 | negative | 329 |
| 2 | 1989 | positive | 340 |
| 3 | Fearless | negative | 161 |
| 4 | Fearless | positive | 230 |
| 5 | Lover | negative | 233 |
| 6 | Lover | positive | 373 |
| 7 | Red | negative | 233 |
| 8 | Red | positive | 326 |
| 9 | Speak Now | negative | 201 |
| 10 | Speak Now | positive | 258 |
| 11 | Taylor Swift | negative | 129 |
| 12 | Taylor Swift | positive | 151 |
| 13 | folklore | negative | 195 |

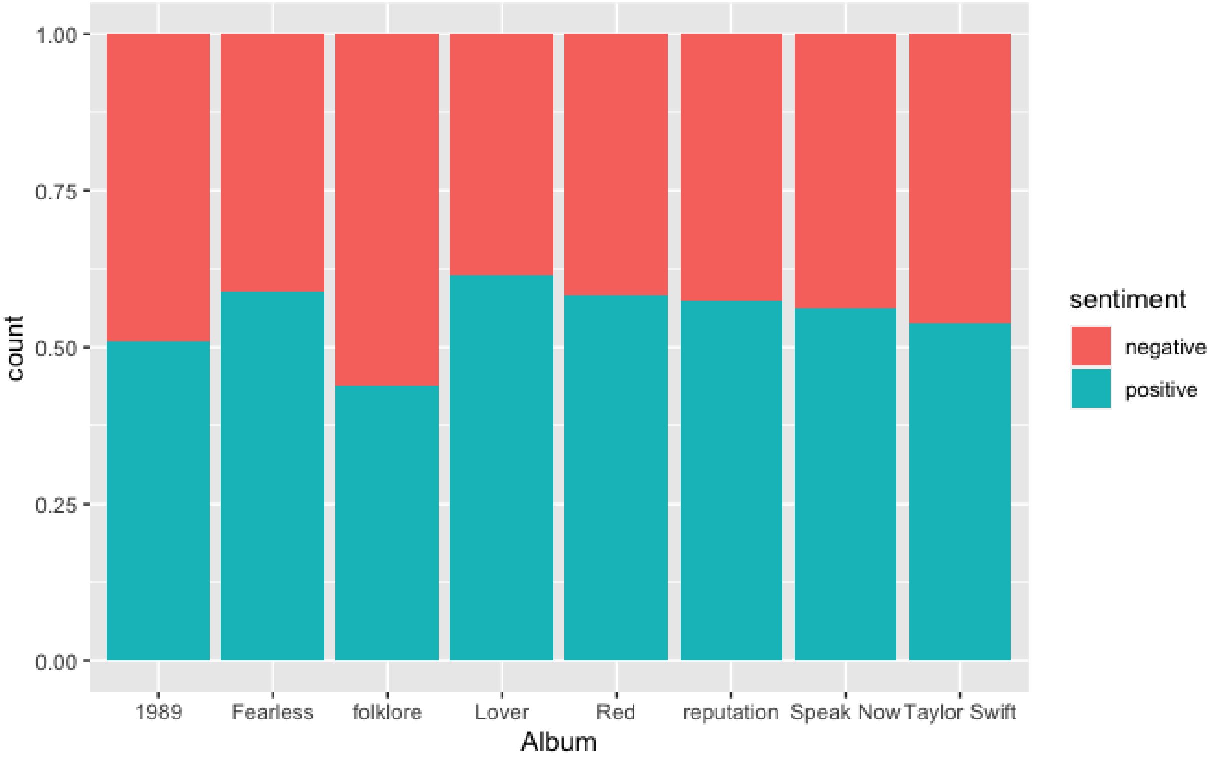
# Class Exercise

- count the number of positive and negative sentiment words by student_star
- What is the lowest star rating that has more positive than negative words?

# Sentiments by Album

- Let’s create a bar chart of sentiments by album
- We get proportions instead of counts with the **position**=‘fill’ argument

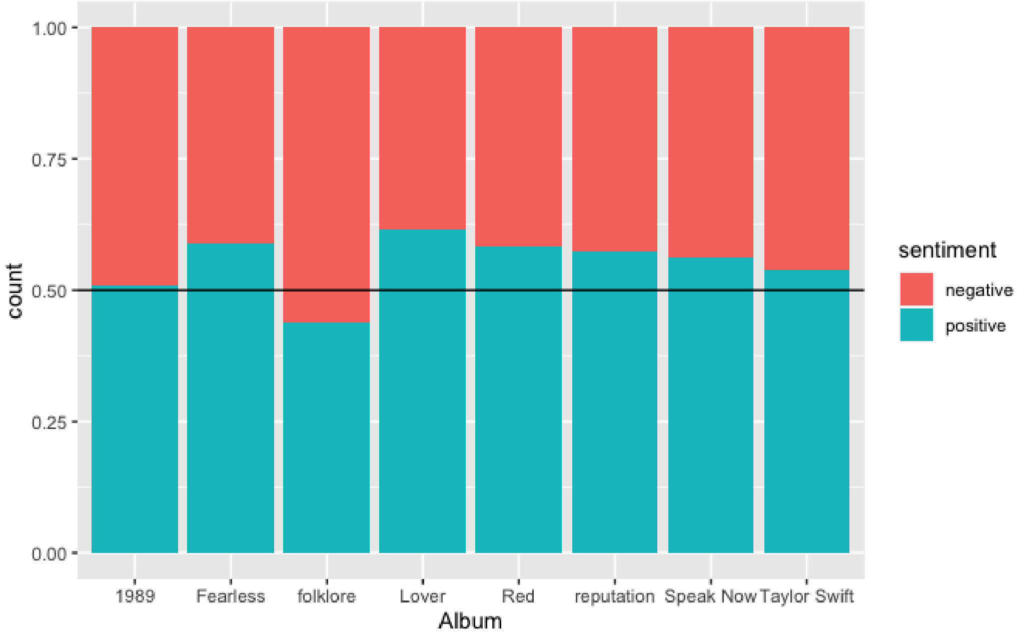
```
ggplot(data=tidy_lyrics_bing,
    mapping=aes(x=Album))+
 geom_bar(aes(fill=sentiment), position='fill')
```



# Sentiments by Album

```
ggplot(data=tidy_lyrics_bing,
      mapping=aes(x=Album))+
 geom_bar(aes(fill=sentiment), position='fill’ )+
      geom_hline(yintercept=.5)
```

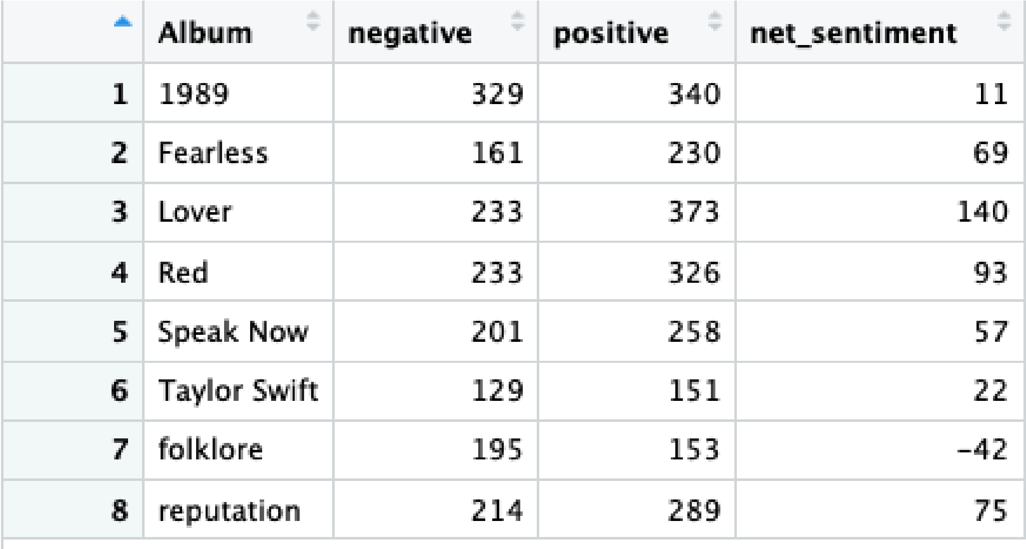
- We can add a reference line at .5 to denote the even split of positive/negative sentiments
- makes it easier to see that folklore was more negative than positive, and 1989 was almost evenly split



# Net Sentiment

- We can also calculate the net sentiment of an Album by comparing the total number of positive words to the total number of negative words:

net sentiment=#positive-#negative

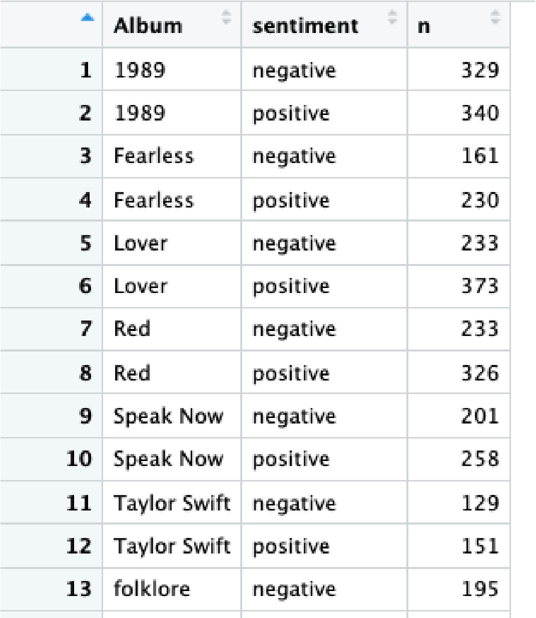


| | Album | negative | positive | net_sentiment |
|---|---|---|---|---|
| 1 | 1989 | 329 | 340 | 11 |
| 2 | Fearless | 161 | 230 | 69 |
| 3 | Lover | 233 | 373 | 140 |
| 4 | Red | 233 | 326 | 93 |
| 5 | Speak Now | 201 | 258 | 57 |
| 6 | Taylor Swift | 129 | 151 | 22 |
| 7 | folklore | 195 | 153 | -42 |
| 8 | reputation | 214 | 289 | 75 |

# Net Sentiment by Album

- We already have the count the number of positive and negative sentiment words by Album

```
bing_album_sentiment <- tidy_lyrics_bing%>%
  group_by(Album)%>%
  count(sentiment) %>%
  arrange(Album, sentiment)
```



| | Album | sentiment | n |
|---|---|---|---|
| 1 | 1989 | negative | 329 |
| 2 | 1989 | positive | 340 |
| 3 | Fearless | negative | 161 |
| 4 | Fearless | positive | 230 |
| 5 | Lover | negative | 233 |
| 6 | Lover | positive | 373 |
| 7 | Red | negative | 233 |
| 8 | Red | positive | 326 |
| 9 | Speak Now | negative | 201 |
| 10 | Speak Now | positive | 258 |
| 11 | Taylor Swift | negative | 129 |
| 12 | Taylor Swift | positive | 151 |
| 13 | folklore | negative | 195 |

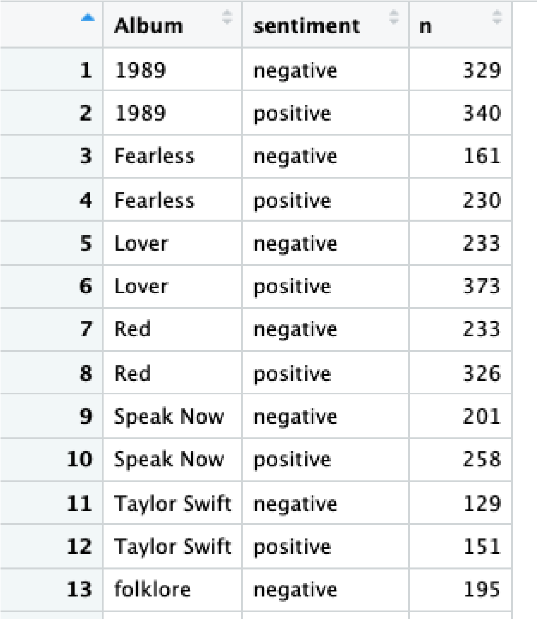
# Net Sentiment by Album

- Next, we use **pivot_wider**() to create columns for the positive sentiment word count and the negative sentiment word count
- Arguments:
  - The column to take the variable **names** from
  - The column to take **values** from

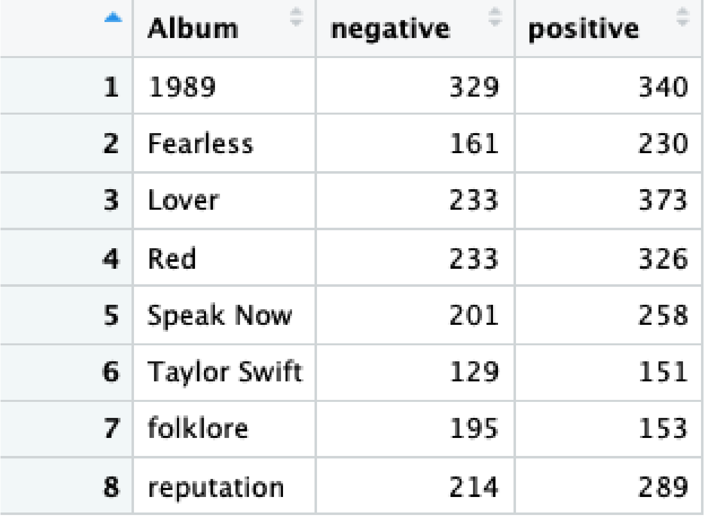
```
bing_album_sentiment_wider<- bing_album_sentiment %>%
  pivot_wider(names_from =sentiment,
                values_from =n)
```

# pivot_wider()

```
bing_album_sentiment_wider<- bing_album_sentiment %>%
 pivot_wider(names_from =sentiment,  values_from =n)
```



|  | Album | sentiment | n |
|---|---|---|---|
| 1 | 1989 | negative | 329 |
| 2 | 1989 | positive | 340 |
| 3 | Fearless | negative | 161 |
| 4 | Fearless | positive | 230 |
| 5 | Lover | negative | 233 |
| 6 | Lover | positive | 373 |
| 7 | Red | negative | 233 |
| 8 | Red | positive | 326 |
| 9 | Speak Now | negative | 201 |
| 10 | Speak Now | positive | 258 |
| 11 | Taylor Swift | negative | 129 |
| 12 | Taylor Swift | positive | 151 |
| 13 | folklore | negative | 195 |

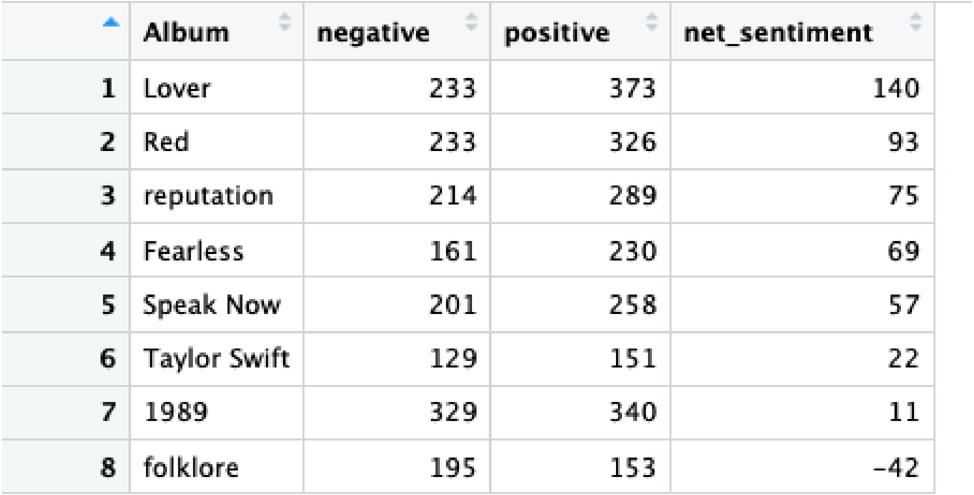


|  | Album | negative | positive |
|---|---|---|---|
| 1 | 1989 | 329 | 340 |
| 2 | Fearless | 161 | 230 |
| 3 | Lover | 233 | 373 |
| 4 | Red | 233 | 326 |
| 5 | Speak Now | 201 | 258 |
| 6 | Taylor Swift | 129 | 151 |
| 7 | folklore | 195 | 153 |
| 8 | reputation | 214 | 289 |

# Net Sentiment by Album

- Last step: create a variable that calculates the net sentiment
- Bonus: sort the albums by net sentiment (most to least positive)

```
bing_album_net_sentiment<-bing_album_sentiment_wider%>%
  mutate(net_sentiment=positive-negative)%>%
  arrange(desc(net_sentiment))
```

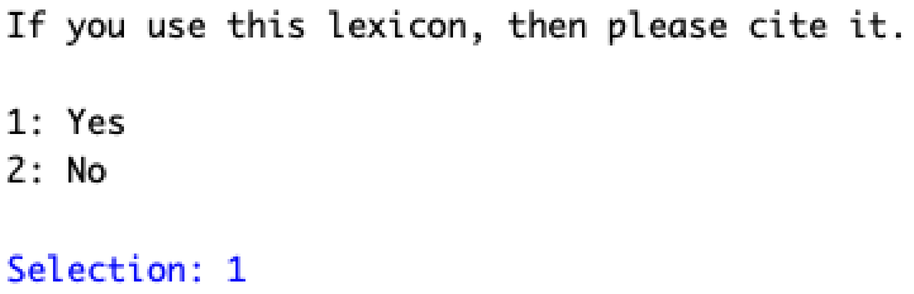


| | Album | negative | positive | net_sentiment |
|---|---|---|---|---|
| 1 | Lover | 233 | 373 | 140 |
| 2 | Red | 233 | 326 | 93 |
| 3 | reputation | 214 | 289 | 75 |
| 4 | Fearless | 161 | 230 | 69 |
| 5 | Speak Now | 201 | 258 | 57 |
| 6 | Taylor Swift | 129 | 151 | 22 |
| 7 | 1989 | 329 | 340 | 11 |
| 8 | folklore | 195 | 153 | -42 |

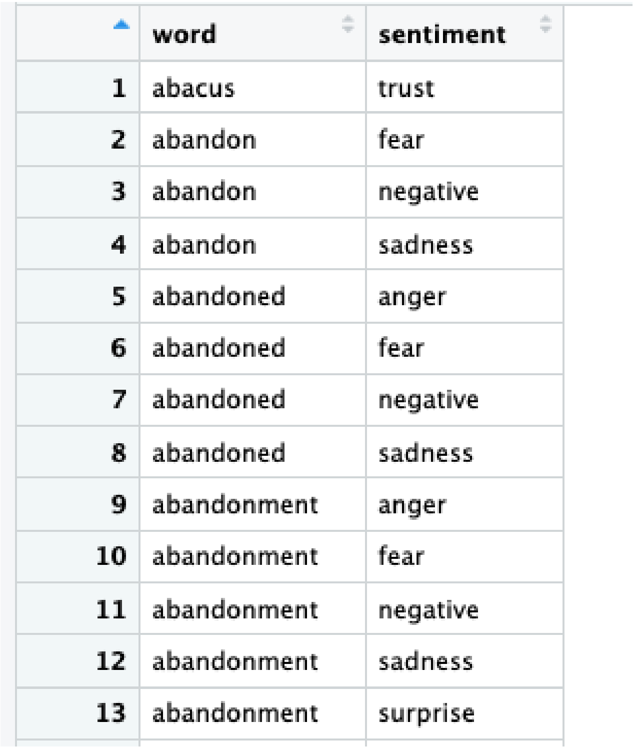
# Sentiment Lexicons

- the **nrc** lexicon categorizes words into ten emotions:
  - positive
  - negative
  - anger
  - fear
  - joy
  - disgust
  - anticipation
  - surprise
  - trust
  - sadness
- **Note**: there is an additional step when loading this lexicon (you have to type “1” when prompted in the console)

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



```
If you use this lexicon, then please cite it.

1: Yes
2: No

Selection: 1
```

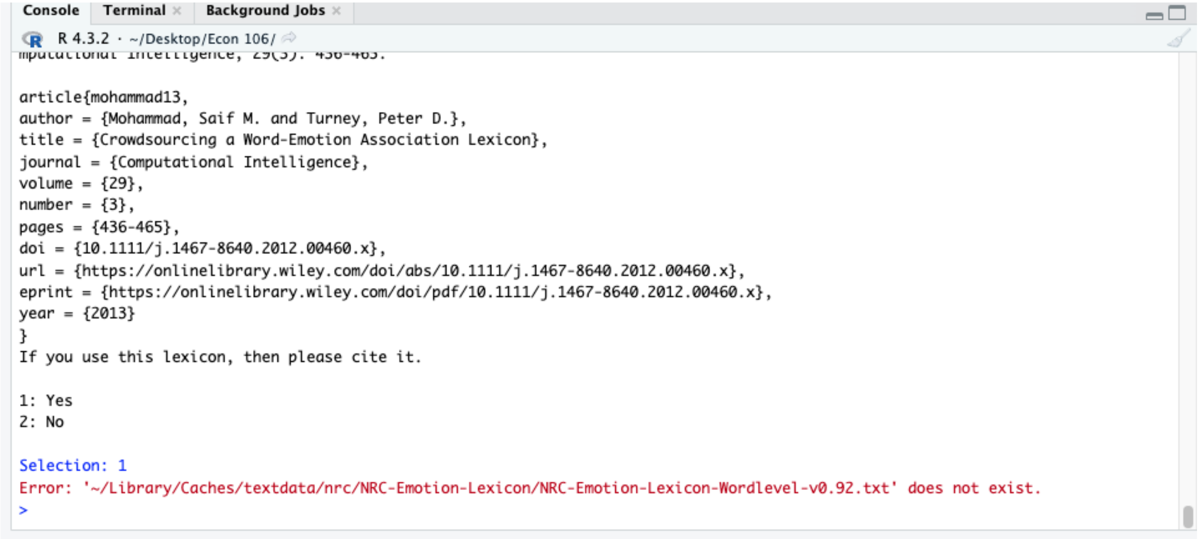


| | word | sentiment |
|---|---|---|
| 1 | abacus | trust |
| 2 | abandon | fear |
| 3 | abandon | negative |
| 4 | abandon | sadness |
| 5 | abandoned | anger |
| 6 | abandoned | fear |
| 7 | abandoned | negative |
| 8 | abandoned | sadness |
| 9 | abandonment | anger |
| 10 | abandonment | fear |
| 11 | abandonment | negative |
| 12 | abandonment | sadness |
| 13 | abandonment | surprise |

# NRC Lexicon loading issue

- some of you might see this error pop up
- If that happens, type the following into your console window:

```
textdata::lexicon_nrc(delete=TRUE)
textdata::lexicon_nrc()
```

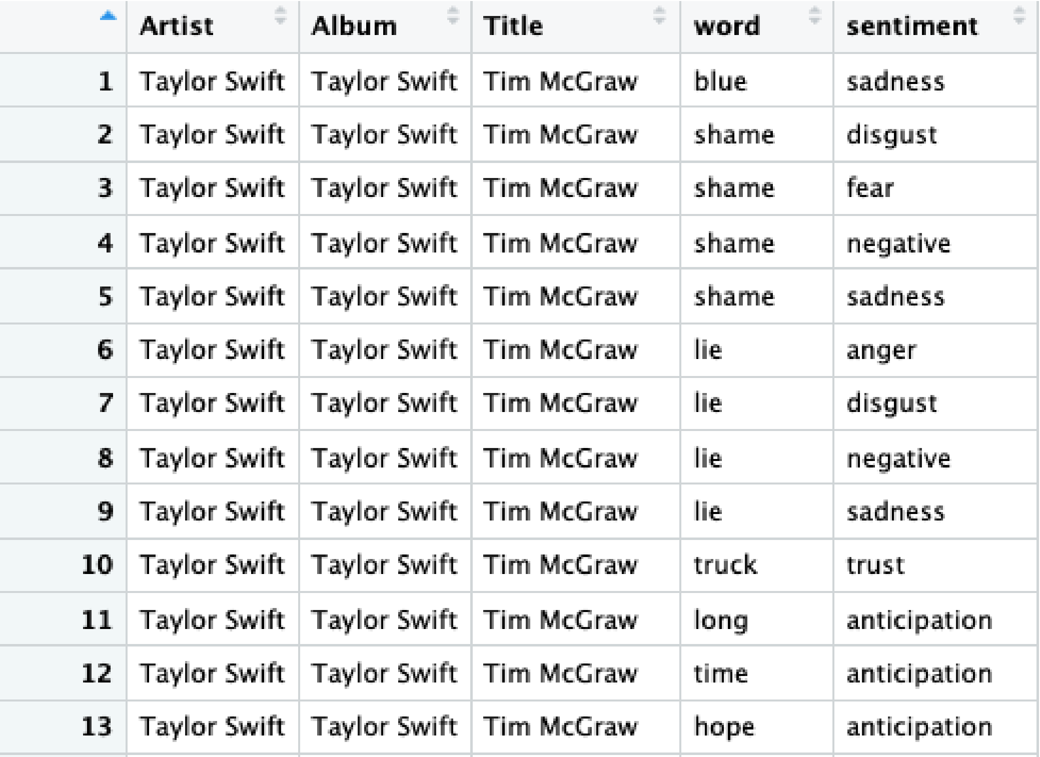


```
mputational Intelligence, 29(3): 436-465.

article{mohammad13,
author = {Mohammad, Saif M. and Turney, Peter D.},
title = {Crowdsourcing a Word-Emotion Association Lexicon},
journal = {Computational Intelligence},
volume = {29},
number = {3},
pages = {436-465},
doi = {10.1111/j.1467-8640.2012.00460.x},
url = {https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.2012.00460.x},
eprint = {https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8640.2012.00460.x},
year = {2013}
}
If you use this lexicon, then please cite it.

1: Yes
2: No

Selection: 1
Error: '~/Library/Caches/textdata/nrc/NRC-Emotion-Lexicon/NRC-Emotion-Lexicon-Wordlevel-v0.92.txt' does not exist.
>
```

# Other Sentiments

- Let’s use the nrc lexicon to assign an emotion to the words in Taylor Swift’s songs
- Note: some words are associated with more than one emotion (you will get a warning about many-to-many matches in R)

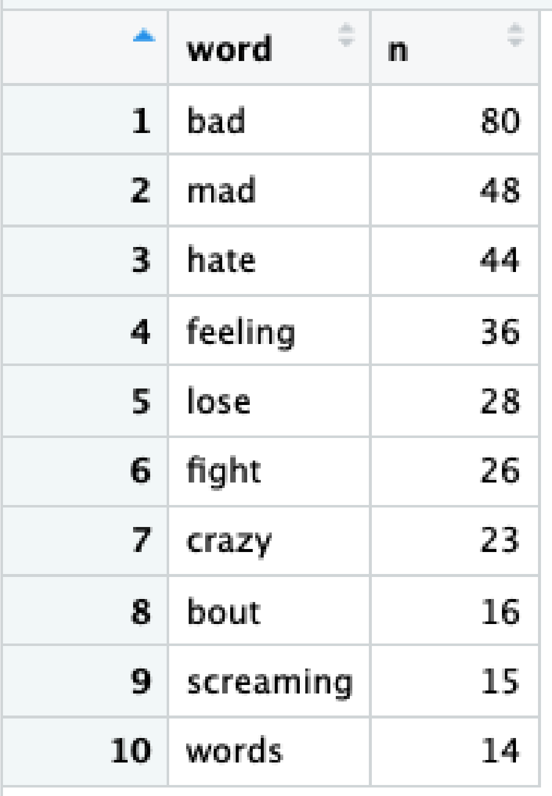
```
tidy_lyrics_nrc<-
inner_join(x=tidy_lyrics, y=nrc_lexicon)
```



| | Artist | Album | Title | word | sentiment |
|---|---|---|---|---|---|
| 1 | Taylor Swift | Taylor Swift | Tim McGraw | blue | sadness |
| 2 | Taylor Swift | Taylor Swift | Tim McGraw | shame | disgust |
| 3 | Taylor Swift | Taylor Swift | Tim McGraw | shame | fear |
| 4 | Taylor Swift | Taylor Swift | Tim McGraw | shame | negative |
| 5 | Taylor Swift | Taylor Swift | Tim McGraw | shame | sadness |
| 6 | Taylor Swift | Taylor Swift | Tim McGraw | lie | anger |
| 7 | Taylor Swift | Taylor Swift | Tim McGraw | lie | disgust |
| 8 | Taylor Swift | Taylor Swift | Tim McGraw | lie | negative |
| 9 | Taylor Swift | Taylor Swift | Tim McGraw | lie | sadness |
| 10 | Taylor Swift | Taylor Swift | Tim McGraw | truck | trust |
| 11 | Taylor Swift | Taylor Swift | Tim McGraw | long | anticipation |
| 12 | Taylor Swift | Taylor Swift | Tim McGraw | time | anticipation |
| 13 | Taylor Swift | Taylor Swift | Tim McGraw | hope | anticipation |

# Top 10 Anger Words

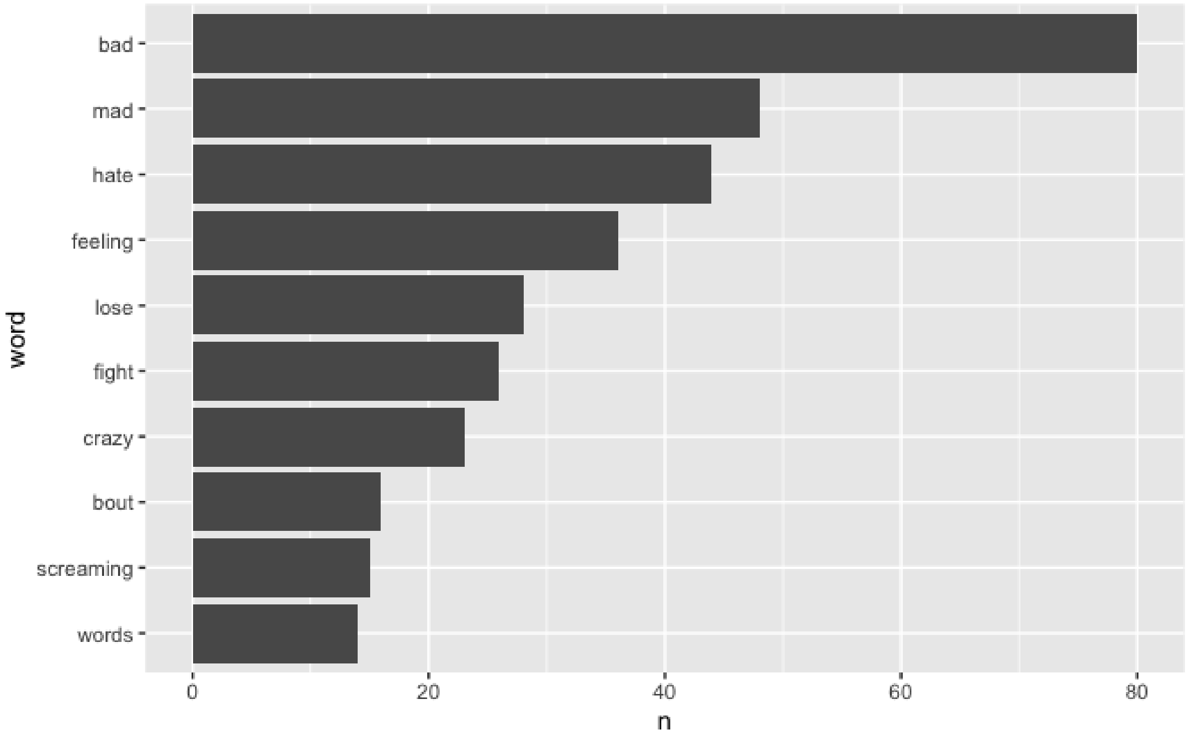
```
nrc_anger_count <- tidy_lyrics_nrc%>%
              filter(sentiment=="anger") %>%
              count(word) %>%
               arrange(desc( n)) %>%
               slice_head(n=10)
```



| | word | n |
|---|---|---|
| 1 | bad | 80 |
| 2 | mad | 48 |
| 3 | hate | 44 |
| 4 | feeling | 36 |
| 5 | lose | 28 |
| 6 | fight | 26 |
| 7 | crazy | 23 |
| 8 | bout | 16 |
| 9 | screaming | 15 |
| 10 | words | 14 |

# Top 10 Anger Words

```
ggplot(data= nrc_anger_count,
          mapping=aes(x= reorder(word, n), y=n))+
  geom_col()+
  coord_flip()
```


bad
mad
hate
feeling
lose
fight
crazy
bout
screaming
words
word
n