Econ 106

Lecture 13

slides derived from:

https://www.tidytextmining.com/tidytext

Reminders

- Research Milestone #2 due Sunday, 11:59pm
- Please review the feedback from MS #1, let me or Fan know if you have any questions

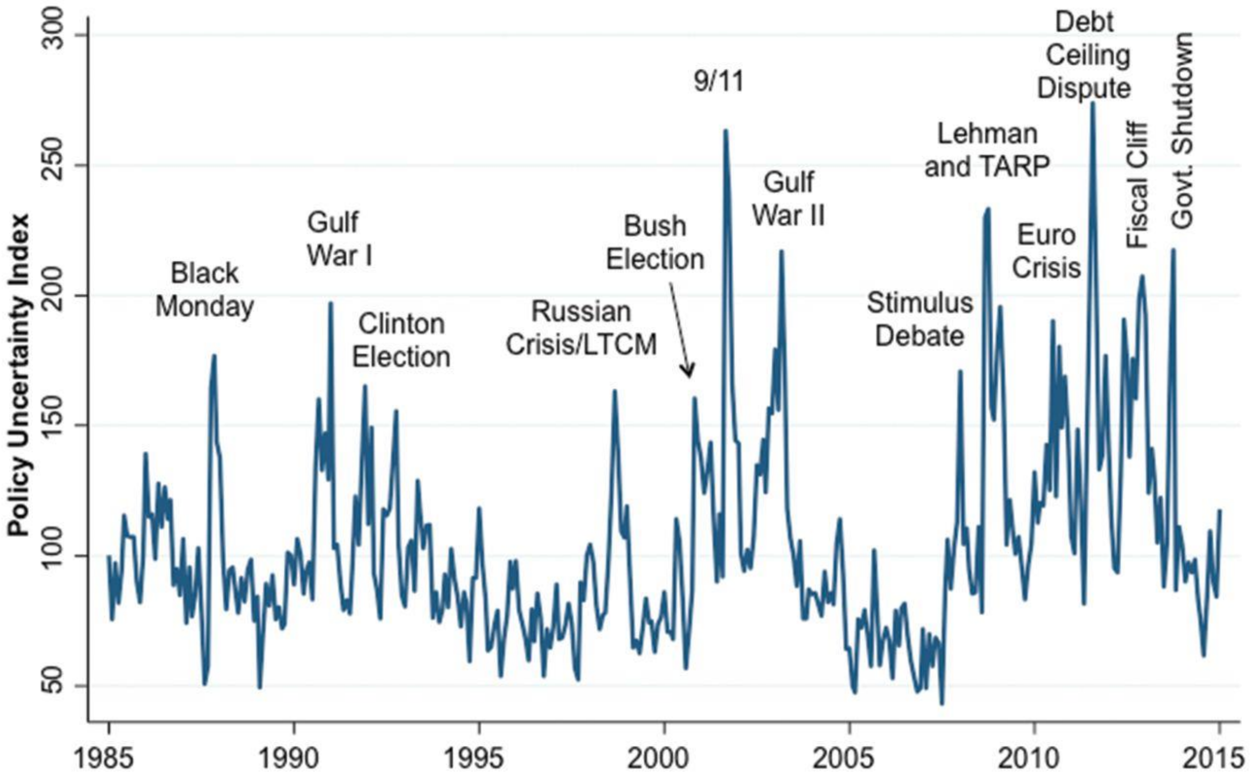
https://pollev.com/vsovero

Outline

- Text as Data:
    - tokenization
    - stop words
    - stemming
    - n-grams

Text as Data: Tracking Policy Uncertainty

- Authors track number of mentions of economic policy uncertainty in newpapers
- Policy uncertainty is associated with:
  - Greater stock price volatility
  - Reduced investment and employment



Index reflects scaled monthly counts of articles containing 'uncertain' or 'uncertainty', 'economic' or 'economy', and one or more policy relevant terms: 'regulation', 'federal reserve', 'deficit', 'congress', 'legislation', or 'white house'. The series is normalized to mean 100 from 1985-2009 and based on queries run on 2 February, 2015 for the USA Today, Miami Herald, Chicago Tribune, Washington Post, LA Times, Boston Globe, SF Chronicle, Dallas Morning News, NY Times, and the Wall Street Journal.

Q J Econ, Volume 131, Issue 4, November 2016, Pages 1593–1636, https://doi.org/10.1093/qje/qjw024

Text as Data: Gender in Economics

- Econjobrumors is a popular online forum for economics graduate students
- Author tracks words used in posts that refer to males vs. females

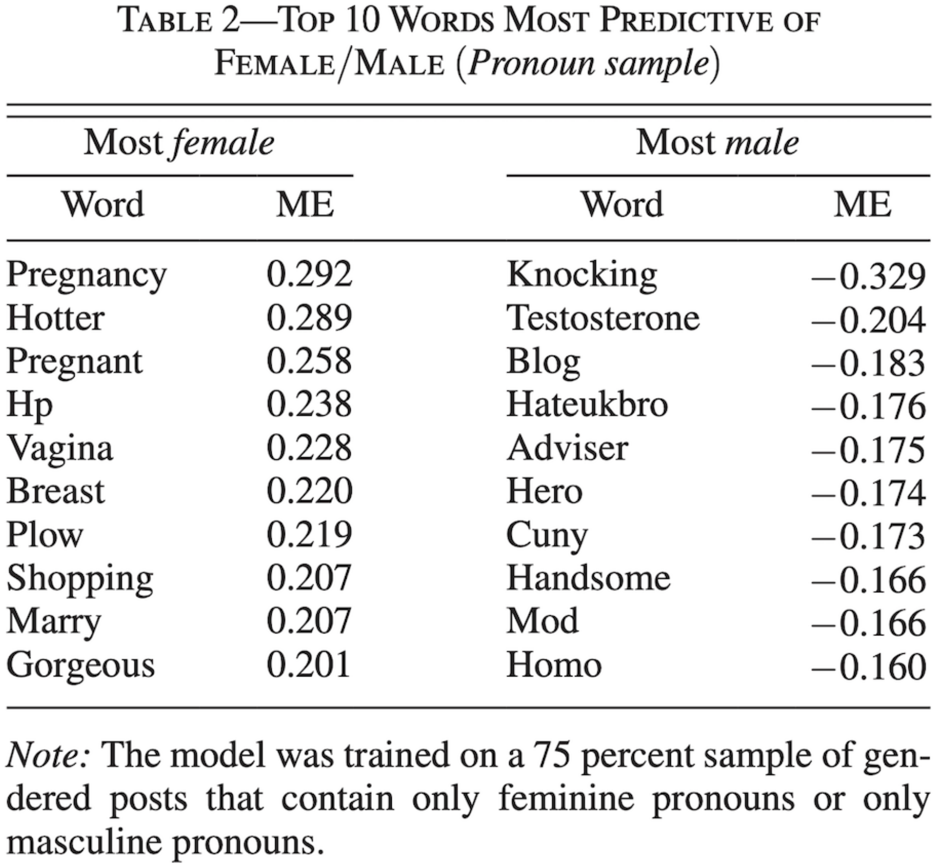


TABLE 2—TOP 10 WORDS MOST PREDICTIVE OF FEMALE/MALE (*Pronoun sample*)

Most *female*		Most *male*	
Word	ME	Word	ME
Pregnancy	0.292	Knocking	−0.329
Hotter	0.289	Testosterone	−0.204
Pregnant	0.258	Blog	−0.183
Hp	0.238	Hateukbro	−0.176
Vagina	0.228	Adviser	−0.175
Breast	0.220	Hero	−0.174
Plow	0.219	Cuny	−0.173
Shopping	0.207	Handsome	−0.166
Marry	0.207	Mod	−0.166
Gorgeous	0.201	Homo	−0.160

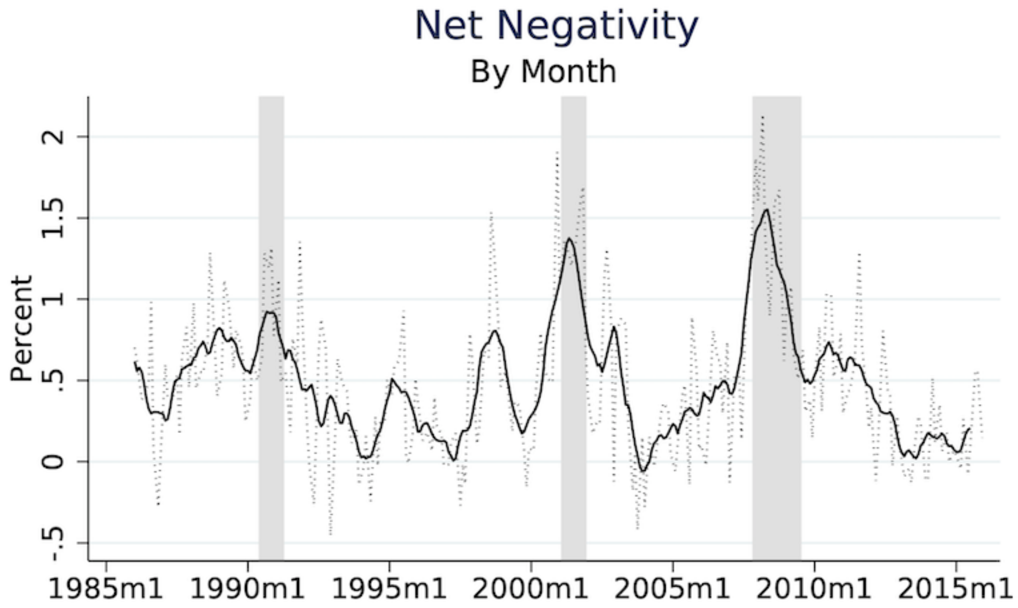
Note: The model was trained on a 75 percent sample of gendered posts that contain only feminine pronouns or only masculine pronouns.

Wu, Alice H. 2018. "Gendered Language on the Economics Job Market Rumors Forum." AEA Papers and Proceedings, 108: 175-79.

Text as Data: Tracking the Central Bank's Preferences

- Authors conduct a sentiment analysis on the transcripts of the FOMC meetings
- Used to estimate the FOMC preferences regarding output and stock market performance

Figure 2: Transcripts of FOMC Meetings

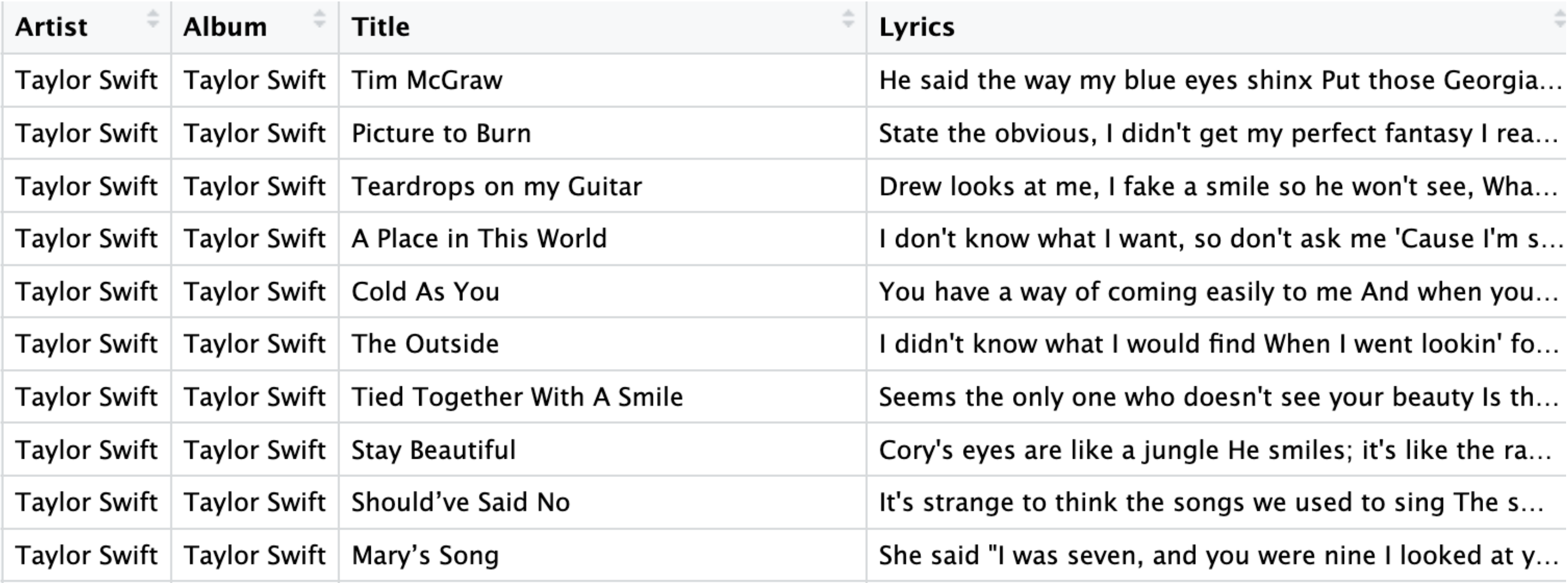


Dotted line plots coefficients on time dummies from regression with speaker fixed effects.
Solid line is 11-month centered moving average.
Gray shaded areas indicate recessions.

Adam Hale Shapiro, Daniel J Wilson, Taking the Fed at its Word: A New Approach to Estimating Central Bank Objectives using Text Analysis, The Review of Economic Studies, Volume 89, Issue 5, October 2022, 2768–2805.

How do we analyze text?

- Text is often stored as strings (many words in a single row)
- For example, the entire lyrics for a Taylor Swift song is in each row:



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

String search with strngr package

- Does a Taylor Swift lyric contain the word “love”?
- Let’s use **str_detect**():
    - Arguments:
        - name of variable with the string
        - pattern you want to detect
    - Output:
        - TRUE/FALSE logical vector
- Use **mutate**() to save it as a new variable

```
taylor_swift_lyrics_love <- taylor_swift_lyrics %>%
        mutate(contains_love=str_detect(Lyrics, "love"))
```

String search

```
taylor_swift_lyrics_love_count <- taylor_swift_lyrics %>%
    mutate(love_count=str_count(Lyrics, "love"))
```

- How many instances of “love” are in each Taylor Swift lyric?
- Let’s use **str_count**():
  - Arguments:
    - name of variable with the string
    - pattern you want to detect
  - Output:
    - numeric vector
- Use **mutate**() to save it as a new variable

Class Exercise

- Find out the number of times Taylor Swift lyrics include the string ”shake it off”
- Find out how many songs contain the string “shake”

https://pollev.com/vsovero

Word Boundaries

- We want to find the standalone word “love”
- Not “glove”
- we can use the symbol for word boundary \\b (where a word must start or end)

```
taylor_swift_lyrics_love <- taylor_swift_lyrics %>%
        mutate(contains_love=str_detect(Lyrics, "\\b love\\b "))
```

Multiple strings

- We want to find either of these :
  - “love”
  - “loving”
  - “lover”
- use | to specify that the pattern can contain “love” or “loving” or “lover”

```
taylor_swift_lyrics_love <- taylor_swift_lyrics %>%
          mutate(contains_love=
str_detect(Lyrics, "\\b love\\b | \\b loving\\b | \\b lover\\b"))
```

Be Careful about Upper/Lower Case

- Case matters when searching for strings:
  - “Love” vs. “love”

```
taylor_swift_lyrics_love <- taylor_swift_lyrics %>%
            mutate(contains_love=
str_detect(Lyrics, "\\b Love\\b  |  \\b love\\b "))
```

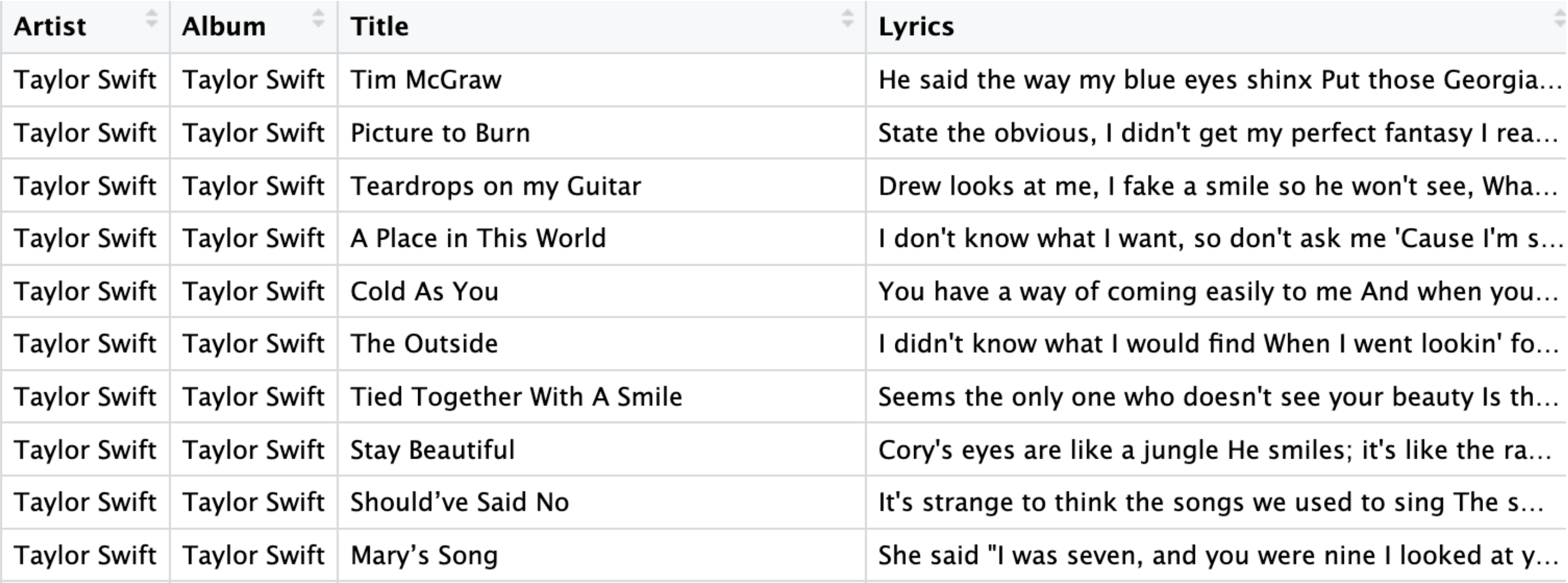
https://pollev.com/vsovero

How else do we analyze text?

- Sometimes we don’t know which words or phrases we are searching for
- Instead, we might want to find out the most common words or phrases in text
- A **token** is a meaningful unit of text, such as a word, that we are interested in using for analysis
- **tokenization** is the process of splitting text into tokens

Tidy Text Data

- To make our text data tidy, we need to create a new row for every token in the Lyrics column



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

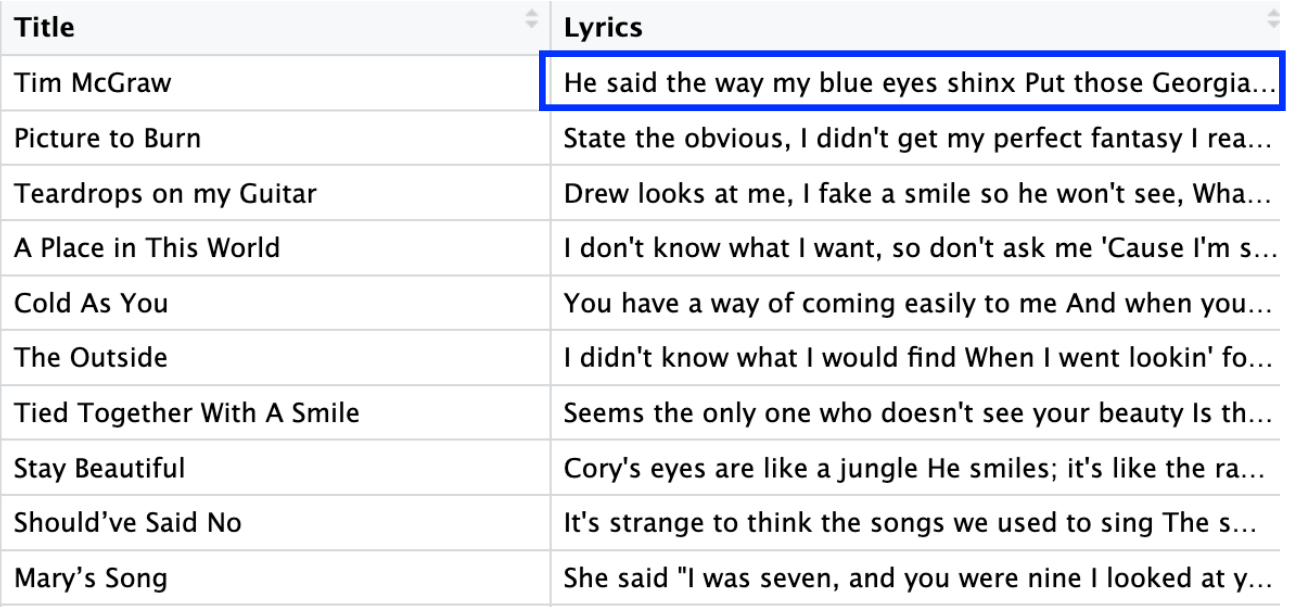
Converting data to tidy text

- we will use the **unnest_tokens**() function from the tidytext library
- Arguments:
  - the name of the input column with the text
  - the name of the output column where you want to place the tokens
- Output:
  - a tidy text data frame where each row represents a token

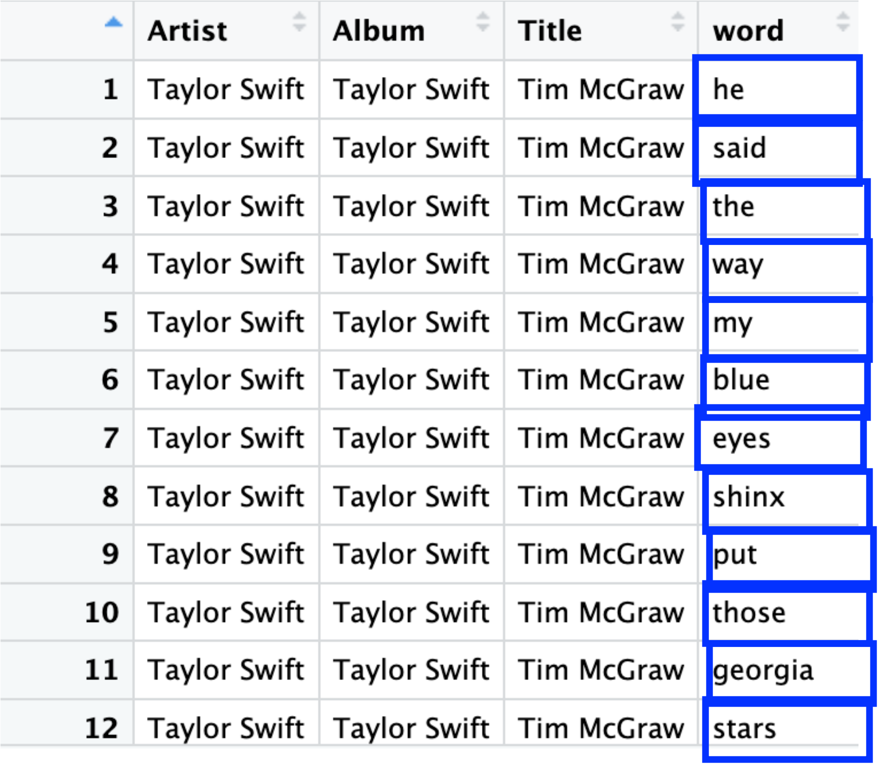
```
tidy_lyrics <- taylor_swift_lyrics %>%
  unnest_tokens(output=word, input=Lyrics)
```

Tidy Text Data

```
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

https://pollev.com/vsovero

Frequency Table of Top Words

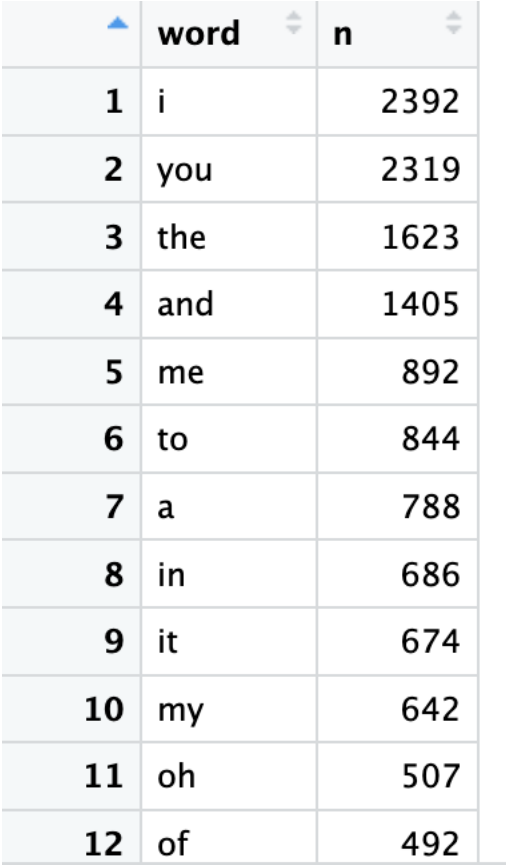
- Ok, what are the top 20 words in Taylor Swift lyrics?
- **count**(word) creates a frequency table for the word variable
- **arrange**(**desc**(n)) sorts the frequency table from largest to smallest value of n

```
tidy_lyrics_count <- tidy_lyrics %>%
        count(word) %>%
         arrange(desc( n))
```

Hm, not very impressive

- The top words don’t seem specific to Taylor Swift
- They’re mainly “filler” words that everyone uses

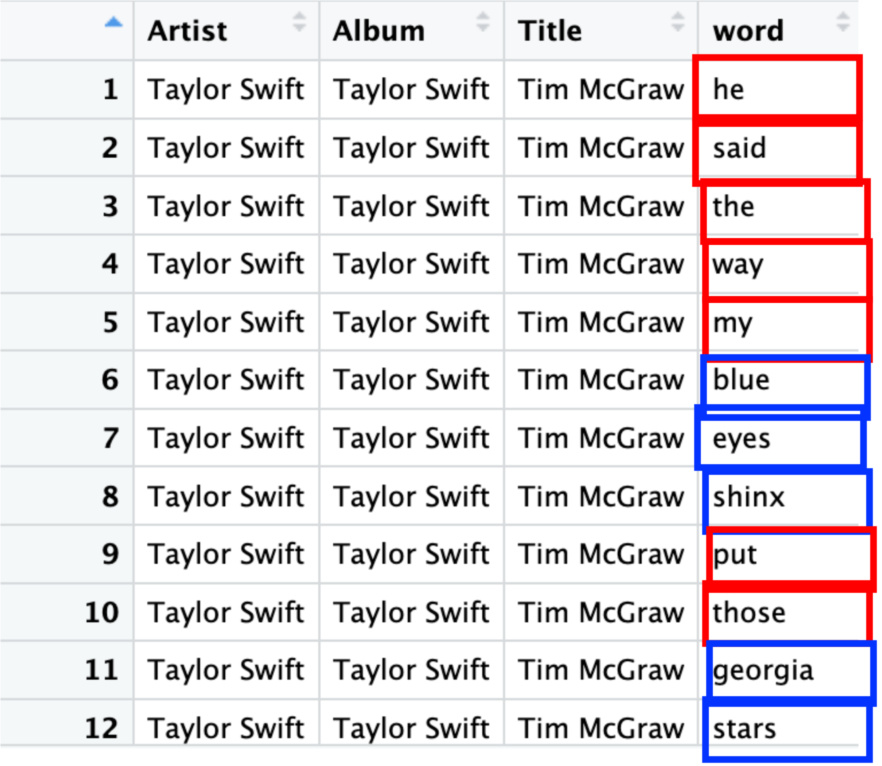
```
tidy_lyrics_count <- tidy_lyrics %>%
count(word) %>%
arrange(desc( n))
```



	word	n
1	i	2392
2	you	2319
3	the	1623
4	and	1405
5	me	892
6	to	844
7	a	788
8	in	686
9	it	674
10	my	642
11	oh	507
12	of	492

Further cleanup: remove stop words

- There are lots of common words that we may want to remove from our data:
    - the
    - a
    - is
    - are
- Why? There carry little meaning in most text analysis
- These are referred to as **stop words**

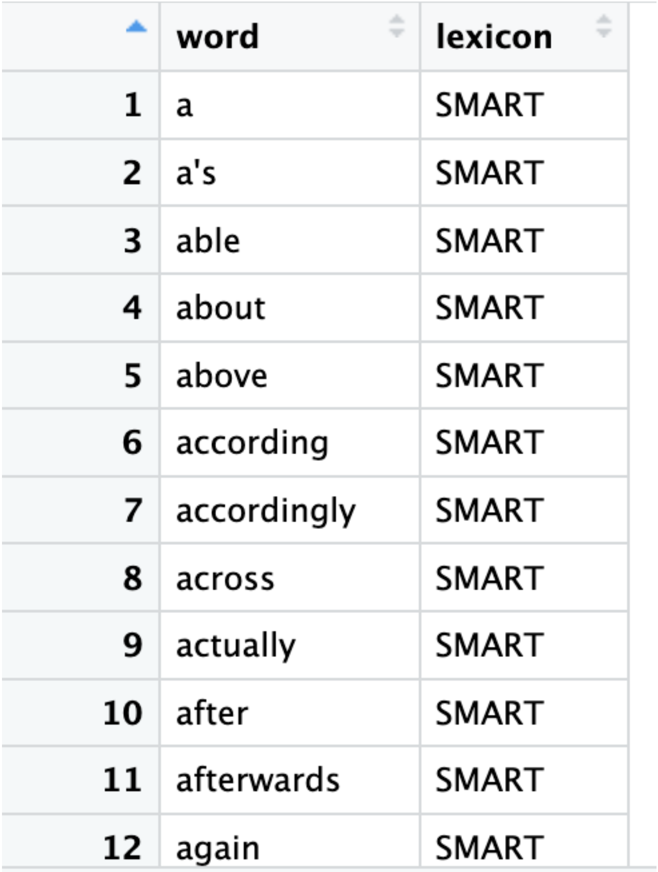


	Artist	Album	Title	word
1	Taylor Swift	Taylor Swift	Tim McGraw	he
2	Taylor Swift	Taylor Swift	Tim McGraw	said
3	Taylor Swift	Taylor Swift	Tim McGraw	the
4	Taylor Swift	Taylor Swift	Tim McGraw	way
5	Taylor Swift	Taylor Swift	Tim McGraw	my
6	Taylor Swift	Taylor Swift	Tim McGraw	blue
7	Taylor Swift	Taylor 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

Removing Stop Words

```
data(stop_words)
```

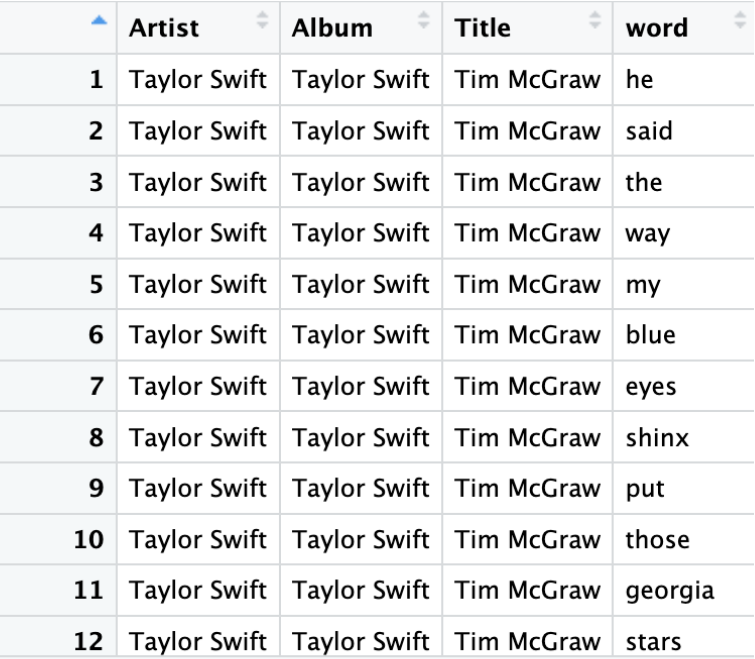
- You could try to use **filter**() to remove stop words, but there are way too many for this approach
- the tidytext package includes a data frame of common stop words
- Let’s use that to remove stop words from Taylor Swift lyrics



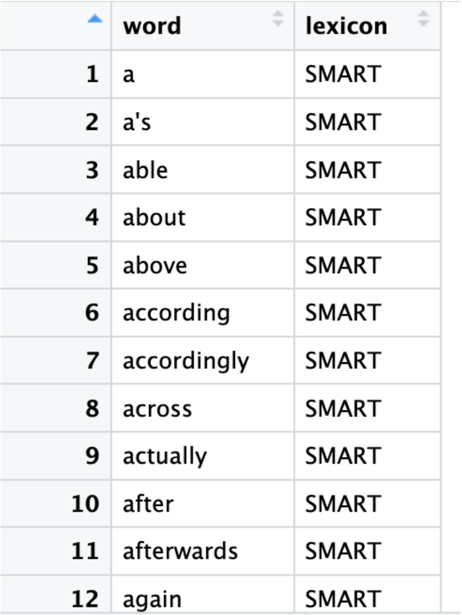
	word	lexicon
1	a	SMART
2	a's	SMART
3	able	SMART
4	about	SMART
5	above	SMART
6	according	SMART
7	accordingly	SMART
8	across	SMART
9	actually	SMART
10	after	SMART
11	afterwards	SMART
12	again	SMART

Use anti_join() to Remove Stop Words

- We are going to use **anti_join**() to remove stop words in the tidy_lyrics
- Any word in tidy_lyrics that matches to the stop words data will be removed



	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



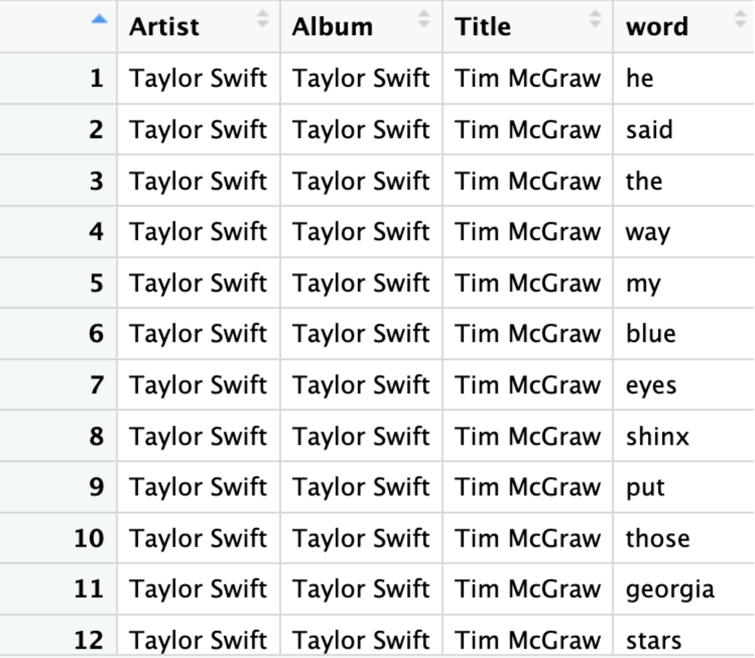
	word	lexicon
1	a	SMART
2	a's	SMART
3	able	SMART
4	about	SMART
5	above	SMART
6	according	SMART
7	accordingly	SMART
8	across	SMART
9	actually	SMART
10	after	SMART
11	afterwards	SMART
12	again	SMART

Use anti_join() to Remove Stop Words

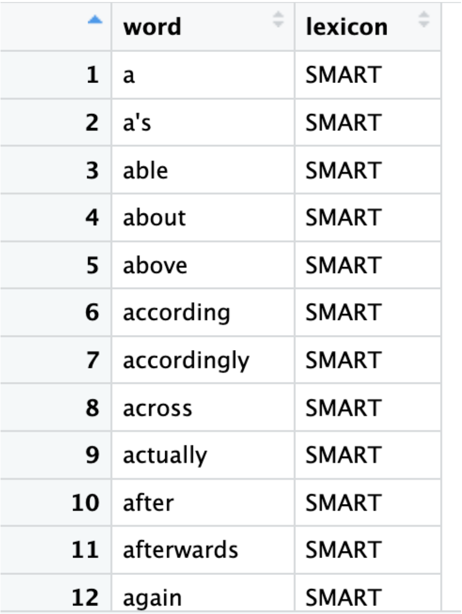
- Arguments:
  - x: left data frame
  - y: right data frame
  - by: linking variable

- Output:
  - Everything in the left data frame that does not have a match to the right data frame

```
tidy_lyrics_remove_stop <- anti_join(x=tidy_lyrics,
                                     y=stop_words,
                                     by="word" )
```



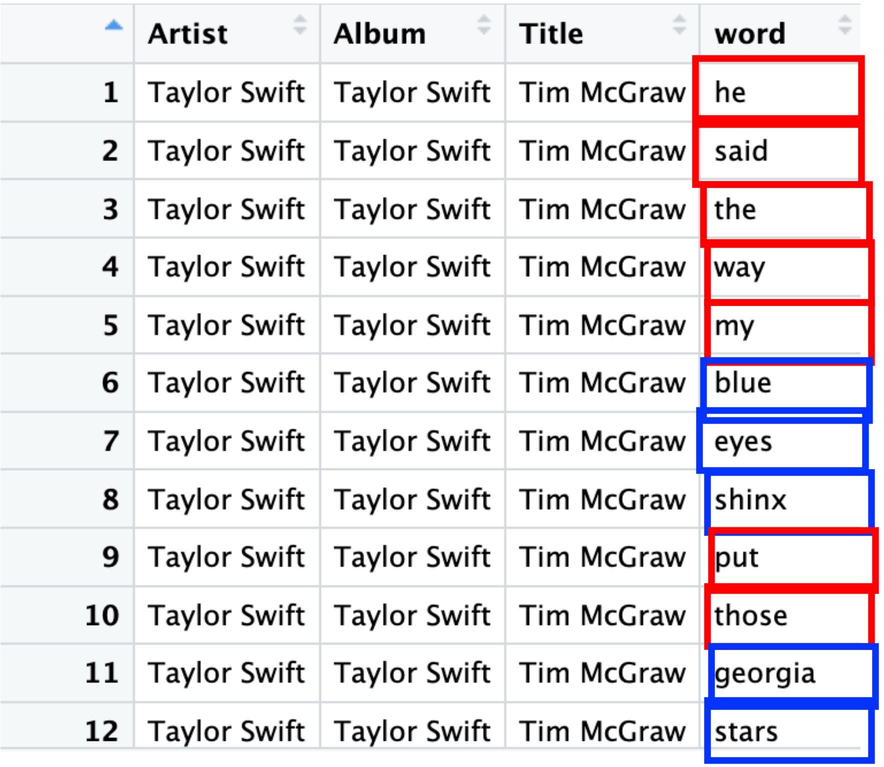
	Artist	Album	Title	word
1	Taylor Swift	Taylor Swift	Tim McGraw	he
2	Taylor Swift	Taylor Swift	Tim McGraw	said
3	Taylor Swift	Taylor Swift	Tim McGraw	the
4	Taylor Swift	Taylor Swift	Tim McGraw	way
5	Taylor Swift	Taylor Swift	Tim McGraw	my
6	Taylor Swift	Taylor Swift	Tim McGraw	blue
7	Taylor Swift	Taylor 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



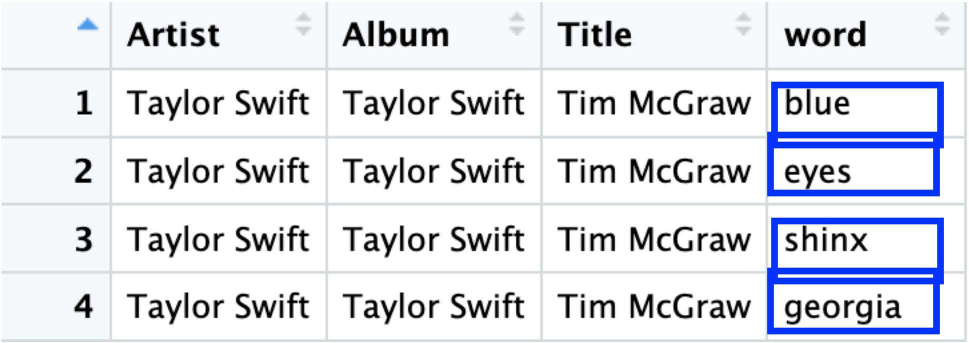
	word	lexicon
1	a	SMART
2	a's	SMART
3	able	SMART
4	about	SMART
5	above	SMART
6	according	SMART
7	accordingly	SMART
8	across	SMART
9	actually	SMART
10	after	SMART
11	afterwards	SMART
12	again	SMART

Taking out the stop words

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

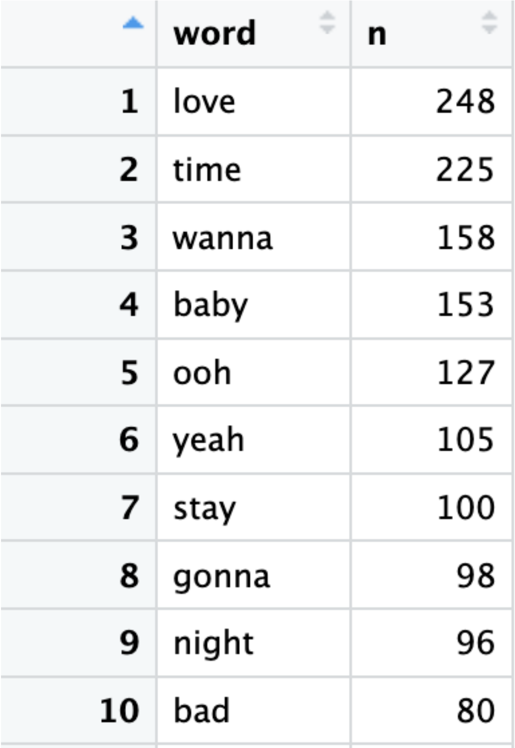


	Artist	Album	Title	word
1	Taylor Swift	Taylor Swift	Tim McGraw	blue
2	Taylor Swift	Taylor Swift	Tim McGraw	eyes
3	Taylor Swift	Taylor Swift	Tim McGraw	shinx
4	Taylor Swift	Taylor Swift	Tim McGraw	georgia

Ok, that looks better

- This list looks more specific to Taylor Swift

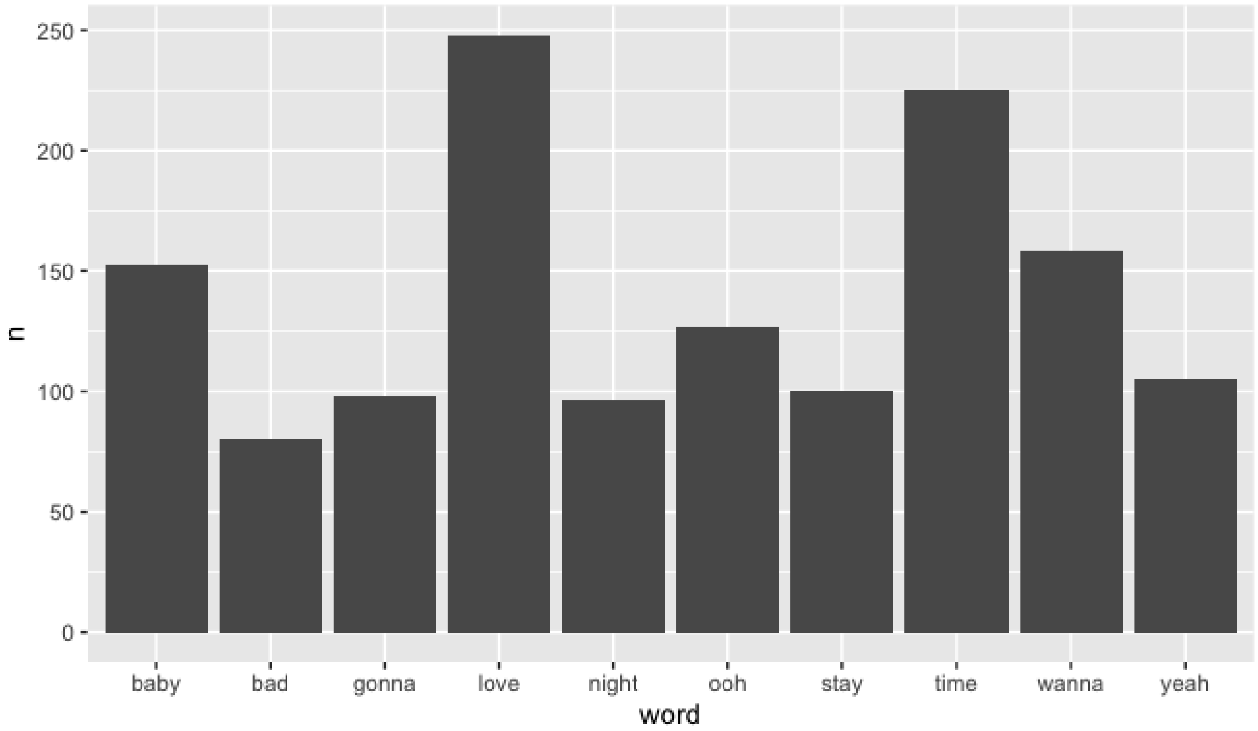
```
tidy_lyrics_top_ten<-
tidy_lyrics_no_stop %>%
            count(word) %>%
             arrange(desc( n)) %>%
             slice_head(n=10)
```



	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

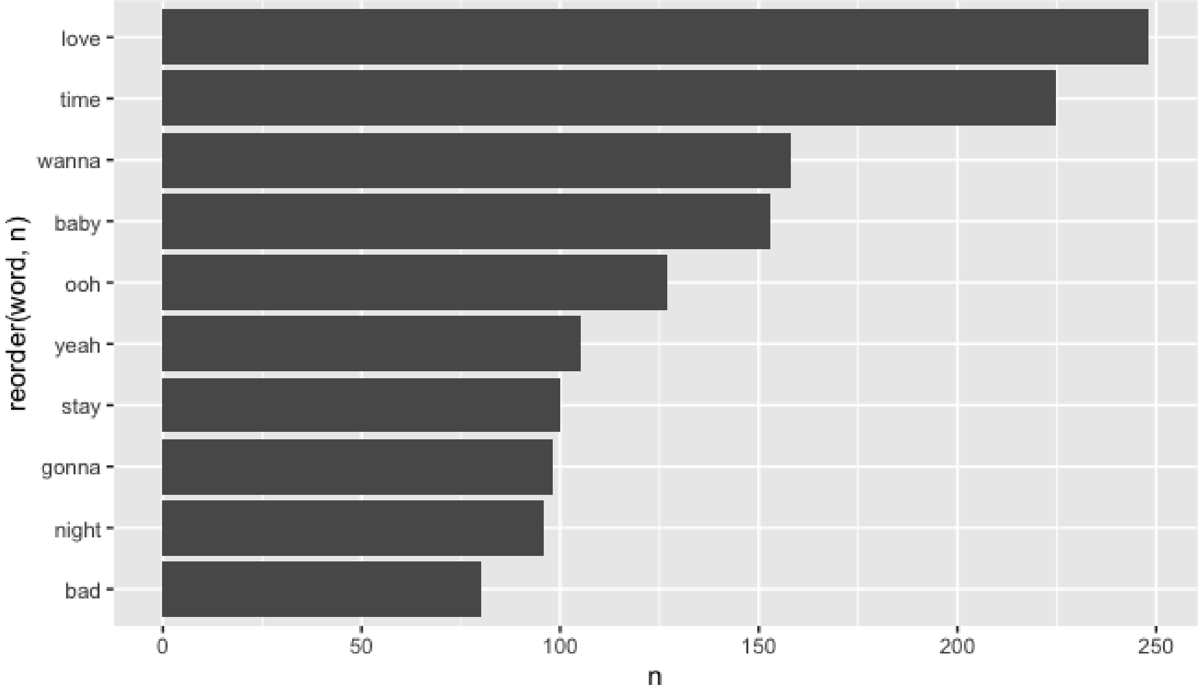
Top 10 Bar Chart (ordered alphabetically)

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



Top 10 Bar Chart (ordered by frequency and flipped)

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



Class Exercise

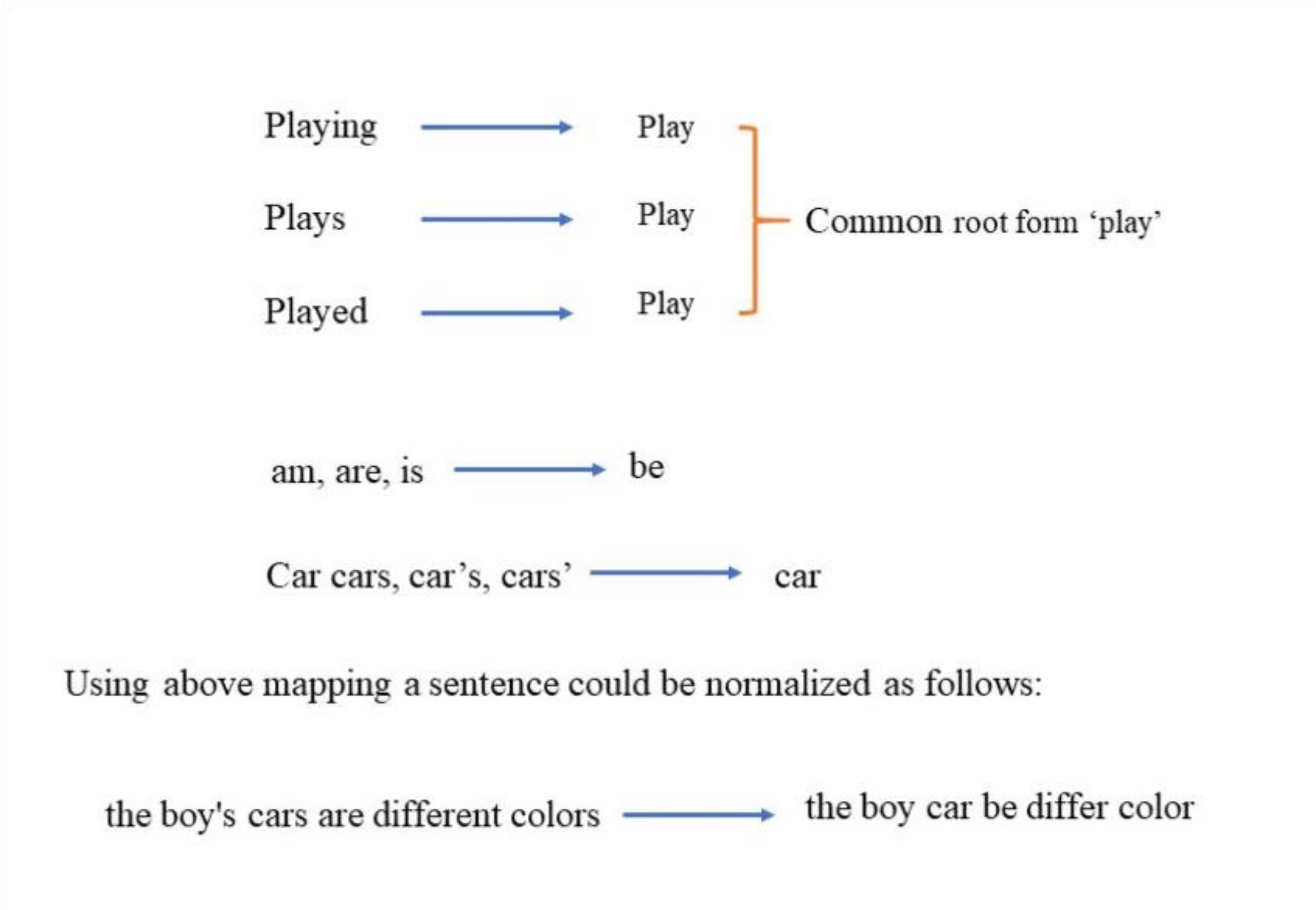
- Create a frequency table of the top ten words for the folklore album
- Put it into a bar chart ordered from highest to lowest frequency

Stemming

- How do we account for the fact that there are many words that have the same base word?
- For example, there are a lot of versions of “love”:
  - loves
  - lover
  - loving
  - loved
- These are different versions of one base word, which is called a **stem**
- Let’s convert the words to their stems to examine the frequency of base words

Stemming

stemming: converting words to their stem (base word)



Playing → Play

Plays → Play

Played → Play

Common root form ‘play’

am, are, is → be

Car cars, car’s, cars’ → car

Using above mapping a sentence could be normalized as follows:

the boy's cars are different colors → the boy car be differ color

Stemming

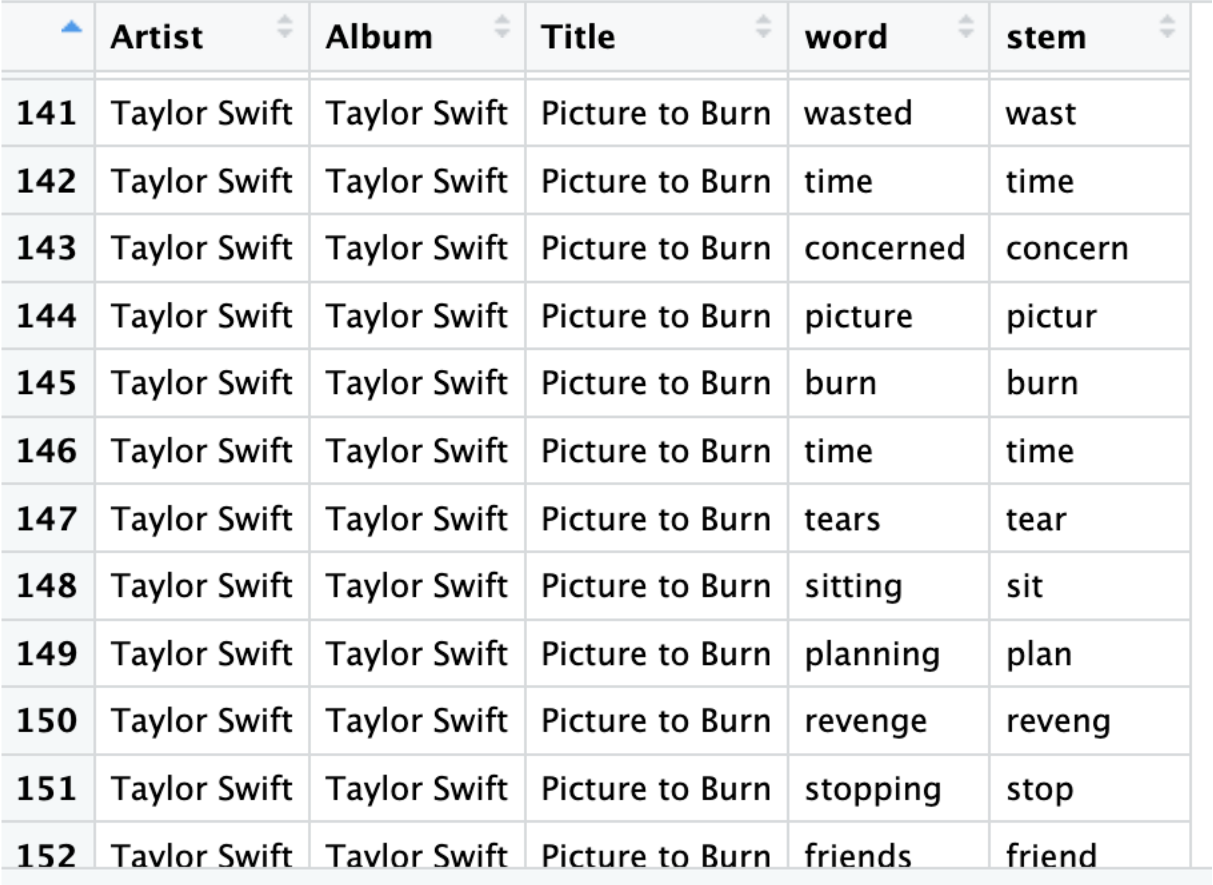
- We will use the **wordStem**() function from the SnowballC package
- Arguments:
  - the name of the variable you want converted to stems
  - the language (English)
- Output:
  - a new variable that contains the stems

```
tidy_lyrics_stem<-tidy_lyrics_no_stop %>%
  mutate(stem = wordStem(word, "en"))
```

Stemming

```
tidy_lyrics_stem<-tidy_lyrics_no_stop %>%
mutate(stem = wordStem(word, "en"))
```

- In this song a lot of stems are different from the original word (planning vs plan, stopping vs stop, etc.)

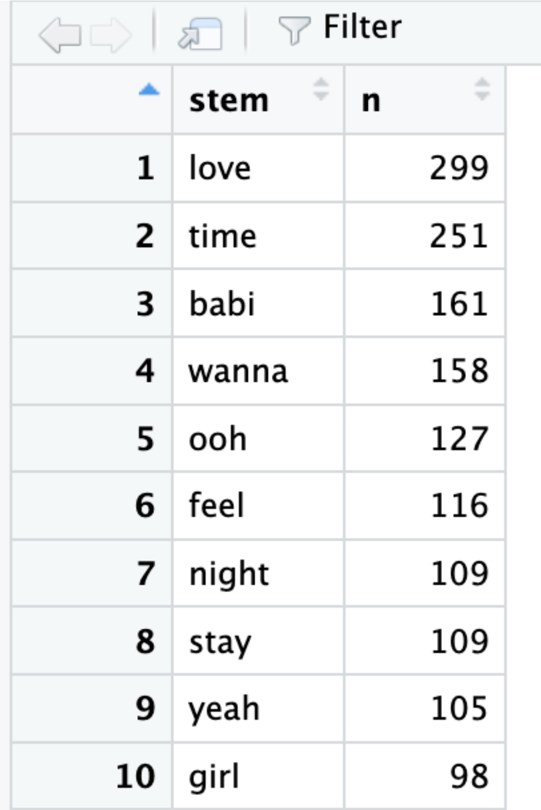


	Artist	Album	Title	word	stem
141	Taylor Swift	Taylor Swift	Picture to Burn	wasted	wast
142	Taylor Swift	Taylor Swift	Picture to Burn	time	time
143	Taylor Swift	Taylor Swift	Picture to Burn	concerned	concern
144	Taylor Swift	Taylor Swift	Picture to Burn	picture	pictur
145	Taylor Swift	Taylor Swift	Picture to Burn	burn	burn
146	Taylor Swift	Taylor Swift	Picture to Burn	time	time
147	Taylor Swift	Taylor Swift	Picture to Burn	tears	tear
148	Taylor Swift	Taylor Swift	Picture to Burn	sitting	sit
149	Taylor Swift	Taylor Swift	Picture to Burn	planning	plan
150	Taylor Swift	Taylor Swift	Picture to Burn	revenge	reveng
151	Taylor Swift	Taylor Swift	Picture to Burn	stopping	stop
152	Taylor Swift	Taylor Swift	Picture to Burn	friends	friend

Top Stems

- Many stems are actual words (*love*, *time*)
- Some are not (*babi* is the stem for baby, babies)

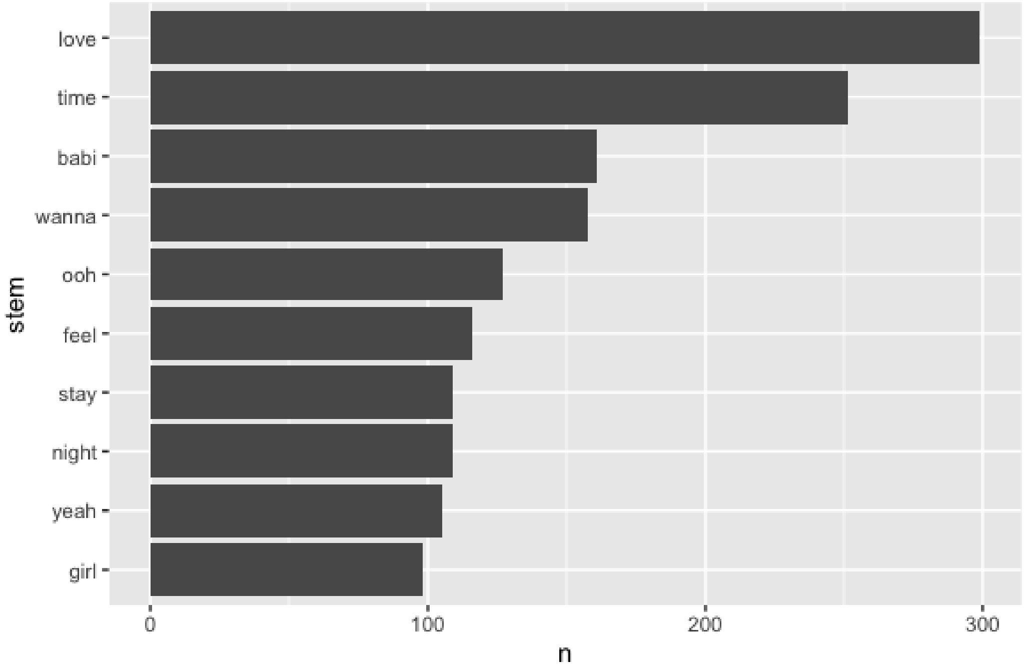
```
tidy_lyrics_stem_count <-
tidy_lyrics_stem%>%
            count(stem) %>%
             arrange(desc( n)) %>%
             slice_head(n=10)
```

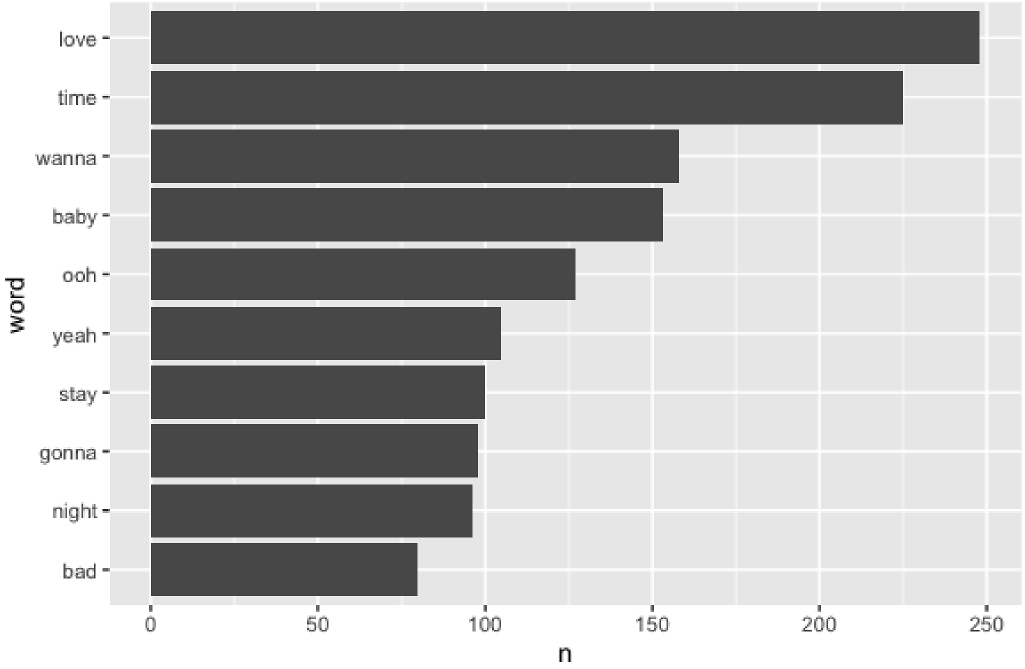


	stem	n
1	love	299
2	time	251
3	babi	161
4	wanna	158
5	ooh	127
6	feel	116
7	night	109
8	stay	109
9	yeah	105
10	girl	98

Top Stems vs. Top Words

- Looks pretty similar to our top 10 word count
- feel wasn’t in the word top 10, but it’s stem is used a lot (feeling, feel, feels)





Class Exercise

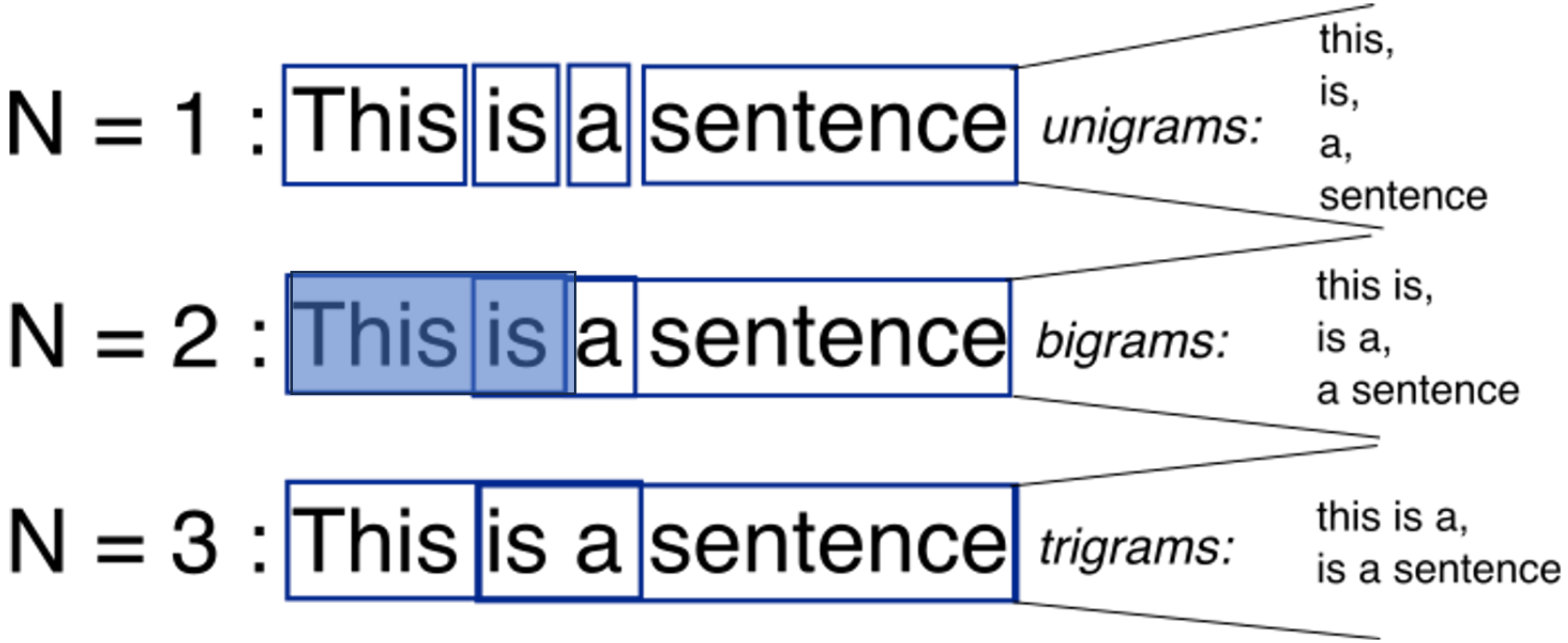
- For the folklore album:
  - find the top ten stems (remember to remove stop words first)
- Put it into a bar chart ordered from highest to lowest frequency

N-grams

- So far, we've have tokenized text into single words
- However, this doesn't give us the ability to examine words in context
- To do this, we can tokenize into consecutive sequences of words (**n-grams**):
    - an n-gram of 2 is two pairs of consecutive words (**bigram)**
    - an n-gram of 1 is a single word (**unigram**)

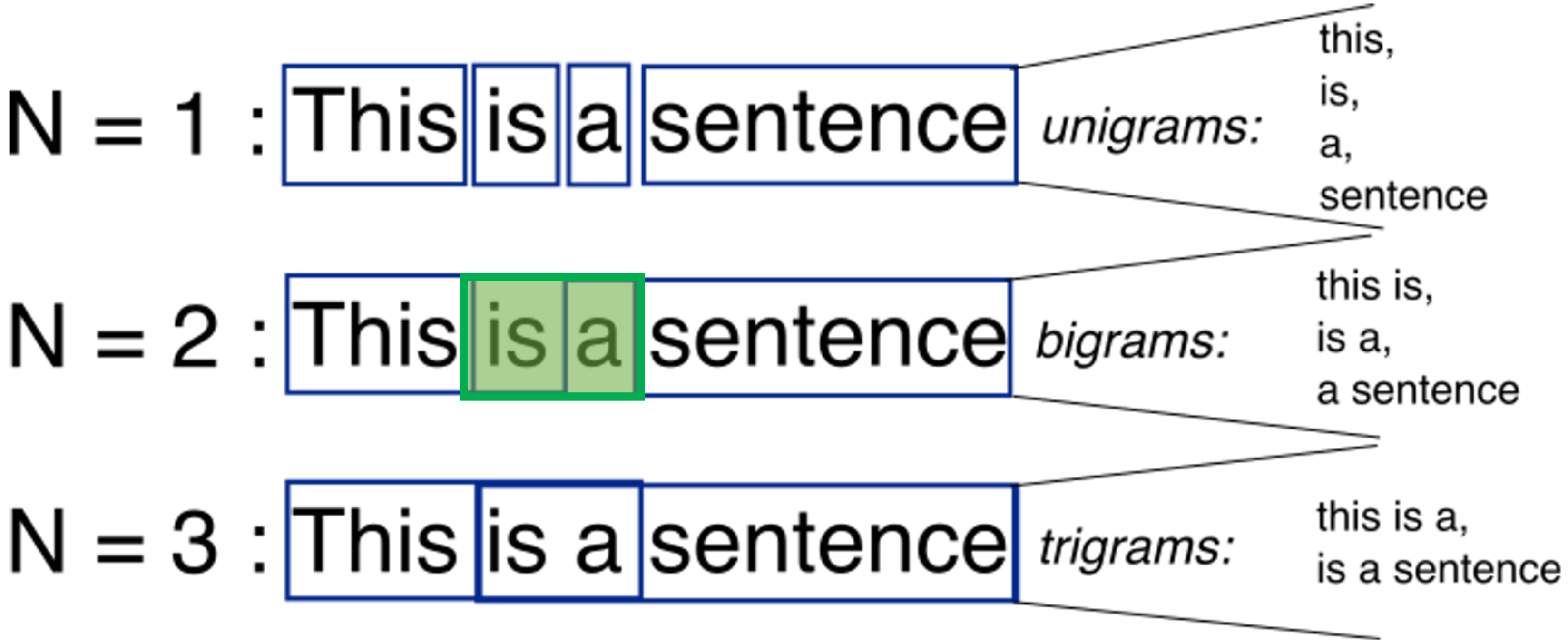
N grams

- Keep in mind that n-grams greater than 1 will generate some overlap



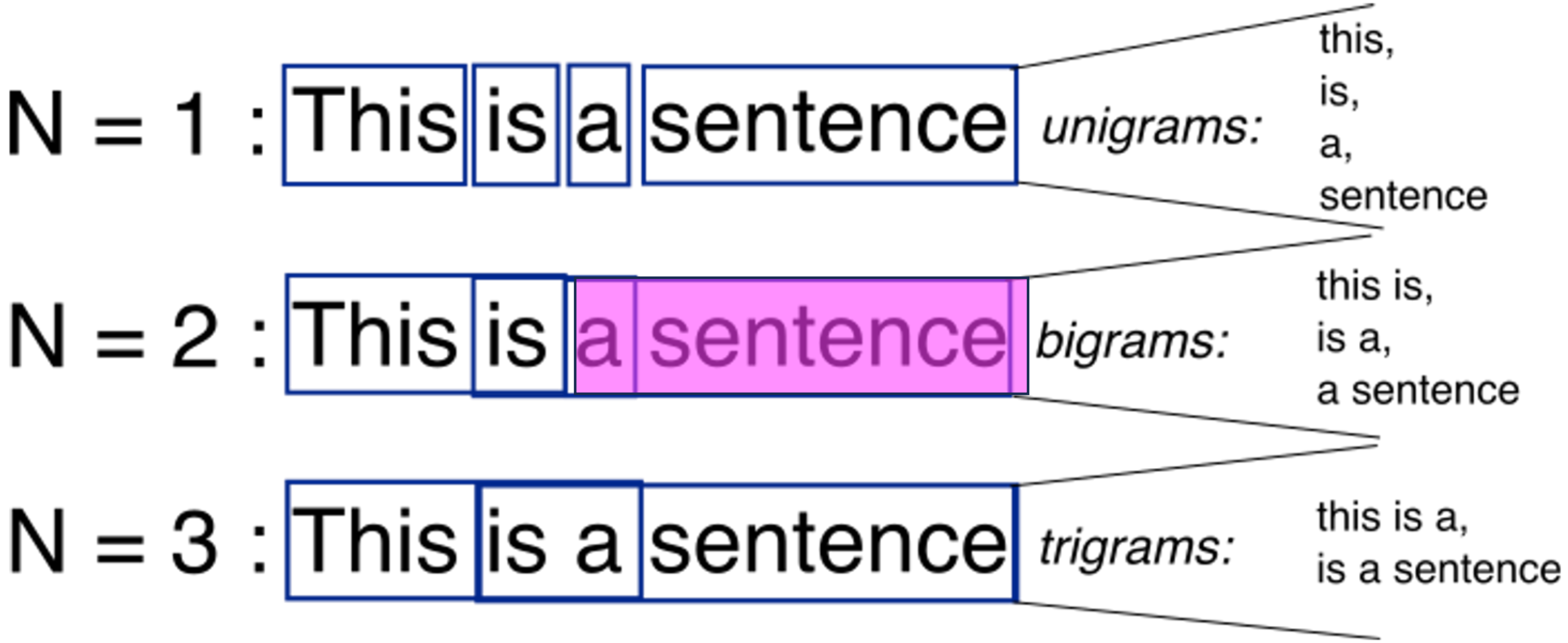
N grams

- Keep in mind that n-grams greater than 1 will generate some overlap



N grams

- Keep in mind that n-grams greater than 1 will generate some overlap



N = 1 : This is a sentence unigrams: this, is, a, sentence

N = 2 : This is a sentence bigrams: this is, is a, a sentence

N = 3 : This is a sentence trigrams: this is a, is a sentence

https://pollev.com/vsovero

Tokenizing into n-grams

- Additional Arguments:
  - token= "ngrams"
  - n: the number of words we wish to capture in each n-gram.

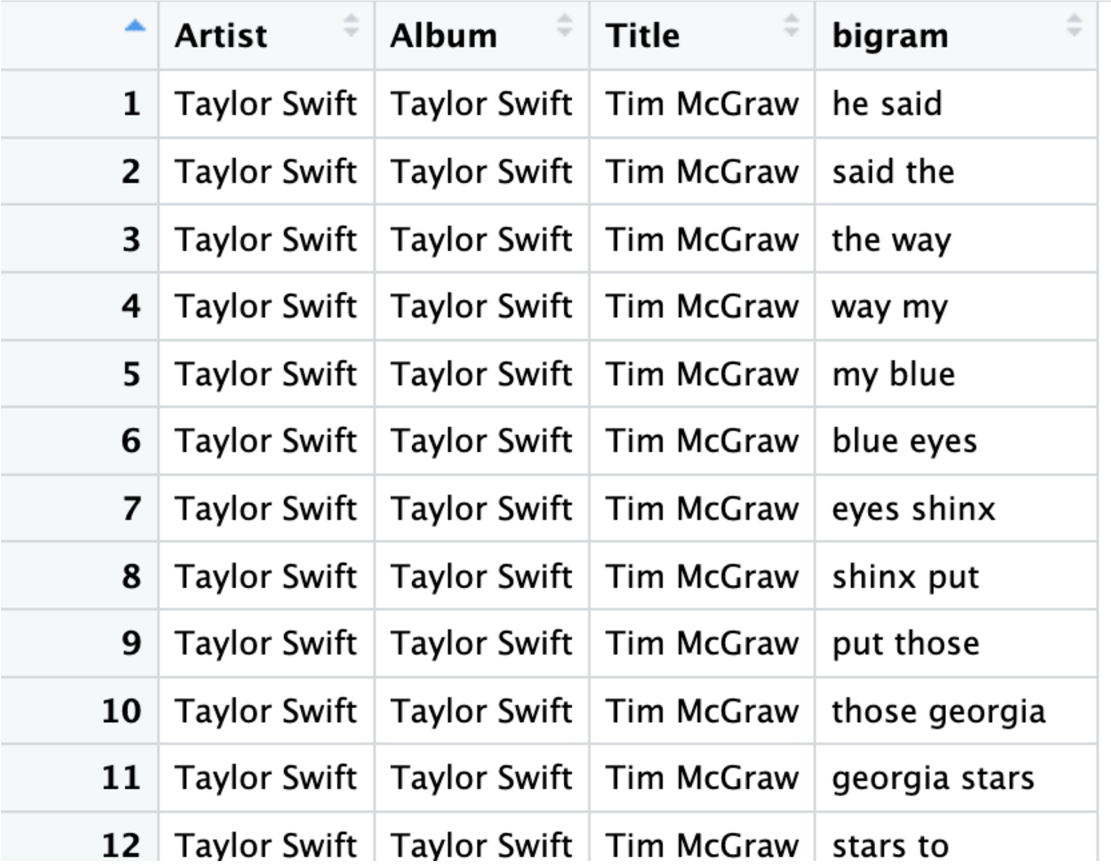
```
tidy_lyrics_bigram <- taylor_swift_lyrics %>%
 unnest_tokens(output=word, input=Lyrics,
                      token="ngrams" , n=2)
```

Tokenizing into n-grams

```
tidy_lyrics_bigram <- taylor_swift_lyrics %>%
  unnest_tokens(output=bigram, input=Lyrics, token="ngrams" , n=2)
```

Removing Stop Words from Bigrams

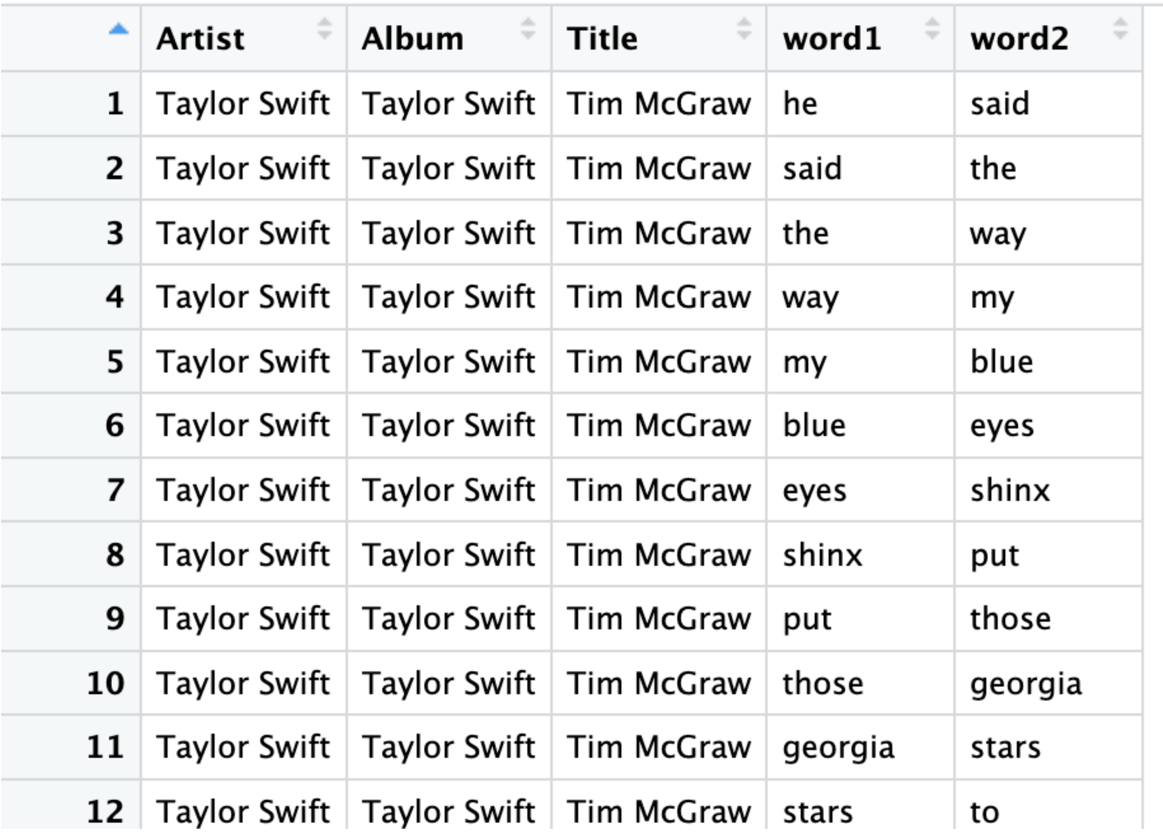
- It’s going to take a little more word to remove the stop words:
    - Split the bigram into two columns
    - Filter out stop_words from the two columns
    - Unite columns to put the bigram back together



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

Split the bigram

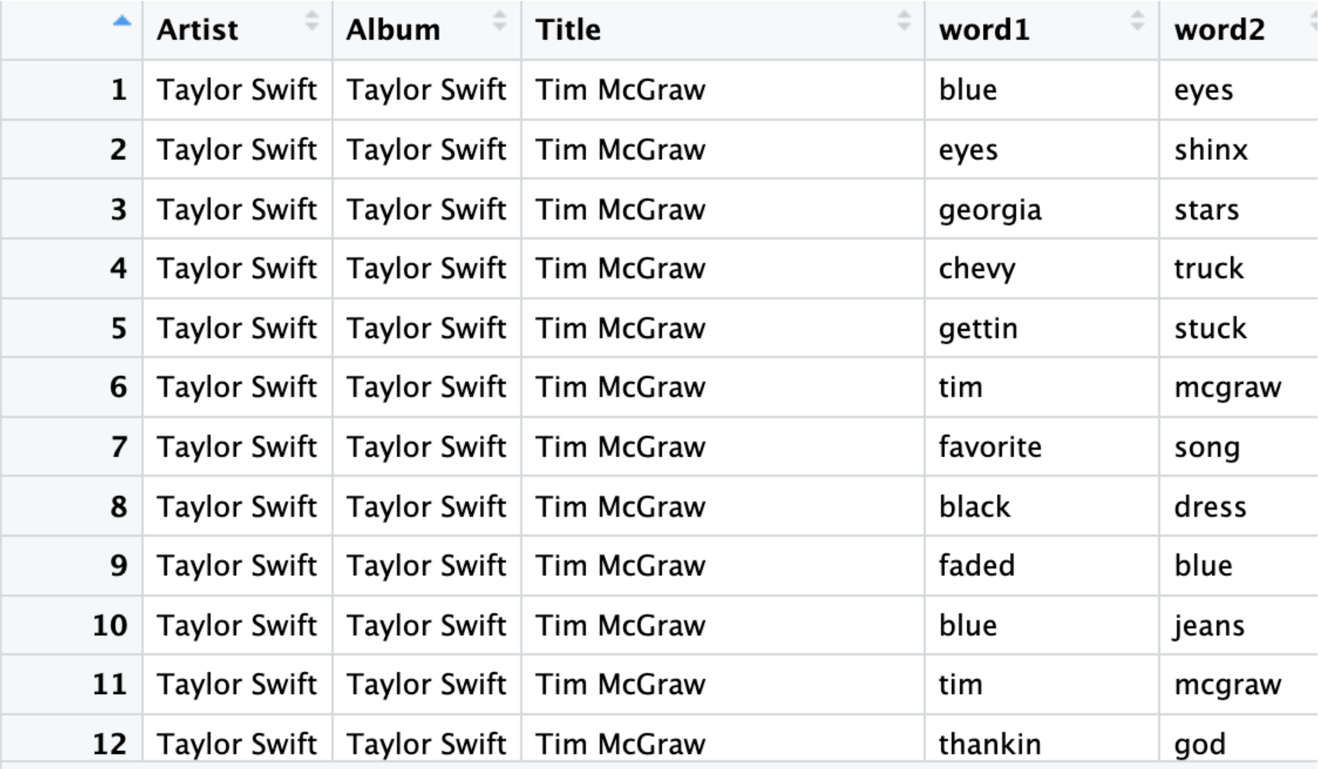
```
tidy_lyrics_bigram_separated <- tidy_lyrics_bigram %>%
separate(bigram, into = c("word1", "word2"), sep = " ")
```



	Artist	Album	Title	word1	word2
1	Taylor Swift	Taylor Swift	Tim McGraw	he	said
2	Taylor Swift	Taylor Swift	Tim McGraw	said	the
3	Taylor Swift	Taylor Swift	Tim McGraw	the	way
4	Taylor Swift	Taylor Swift	Tim McGraw	way	my
5	Taylor Swift	Taylor Swift	Tim McGraw	my	blue
6	Taylor Swift	Taylor Swift	Tim McGraw	blue	eyes
7	Taylor Swift	Taylor Swift	Tim McGraw	eyes	shinx
8	Taylor Swift	Taylor Swift	Tim McGraw	shinx	put
9	Taylor Swift	Taylor Swift	Tim McGraw	put	those
10	Taylor Swift	Taylor Swift	Tim McGraw	those	georgia
11	Taylor Swift	Taylor Swift	Tim McGraw	georgia	stars
12	Taylor Swift	Taylor Swift	Tim McGraw	stars	to

Filter Out Stop Words in Each Column

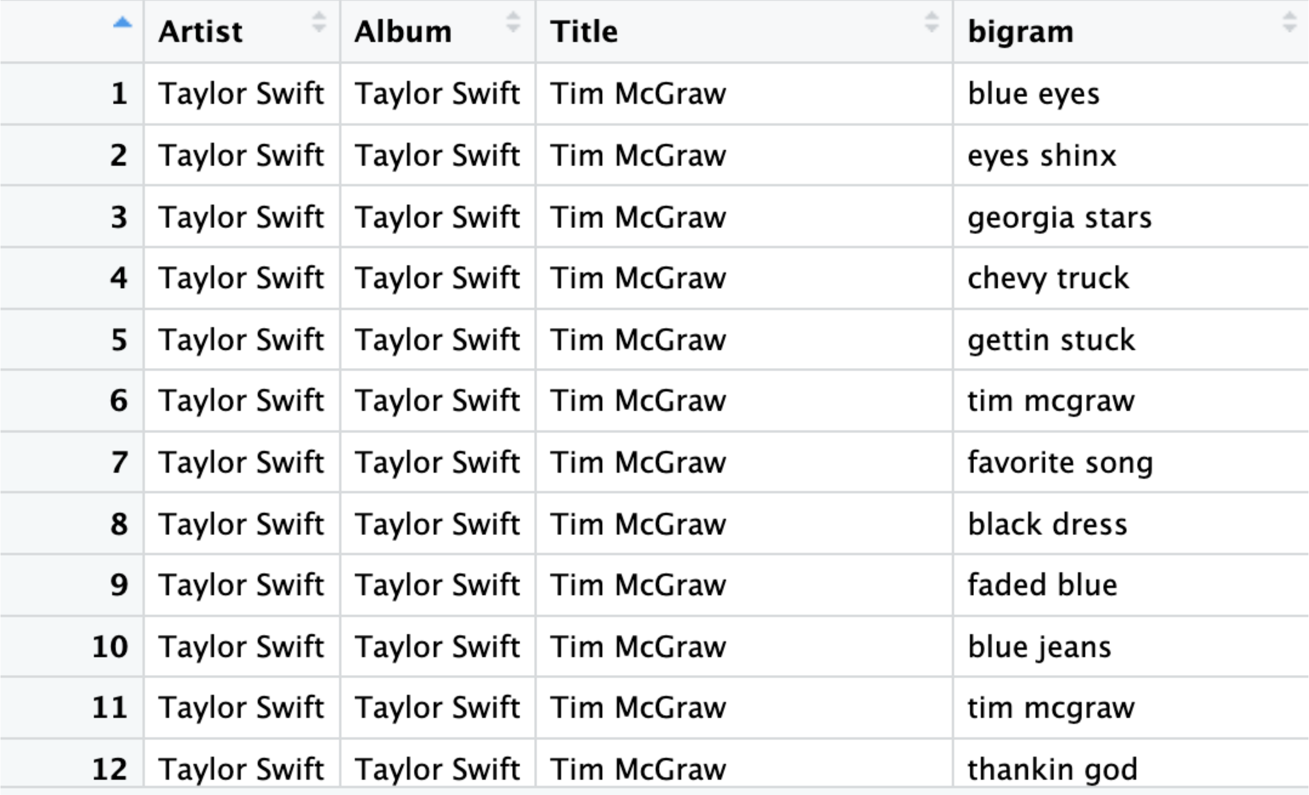
```
tidy_lyrics_bigram_no_stop <- tidy_lyrics_bigram_separated %>%
  filter((!word1 %in% stop_words$word) %>%
  filter((!word2 %in% stop_words$word)
```



	Artist	Album	Title	word1	word2
1	Taylor Swift	Taylor Swift	Tim McGraw	blue	eyes
2	Taylor Swift	Taylor Swift	Tim McGraw	eyes	shinx
3	Taylor Swift	Taylor Swift	Tim McGraw	georgia	stars
4	Taylor Swift	Taylor Swift	Tim McGraw	chevy	truck
5	Taylor Swift	Taylor Swift	Tim McGraw	gettin	stuck
6	Taylor Swift	Taylor Swift	Tim McGraw	tim	mcgraw
7	Taylor Swift	Taylor Swift	Tim McGraw	favorite	song
8	Taylor Swift	Taylor Swift	Tim McGraw	black	dress
9	Taylor Swift	Taylor Swift	Tim McGraw	faded	blue
10	Taylor Swift	Taylor Swift	Tim McGraw	blue	jeans
11	Taylor Swift	Taylor Swift	Tim McGraw	tim	mcgraw
12	Taylor Swift	Taylor Swift	Tim McGraw	thankin	god

Unite the bigram

```
tidy_lyrics_bigram_united <- tidy_lyrics_bigram_no_stop %>%
 unite(bigram,   c("word1", "word2"), sep = " ")
```

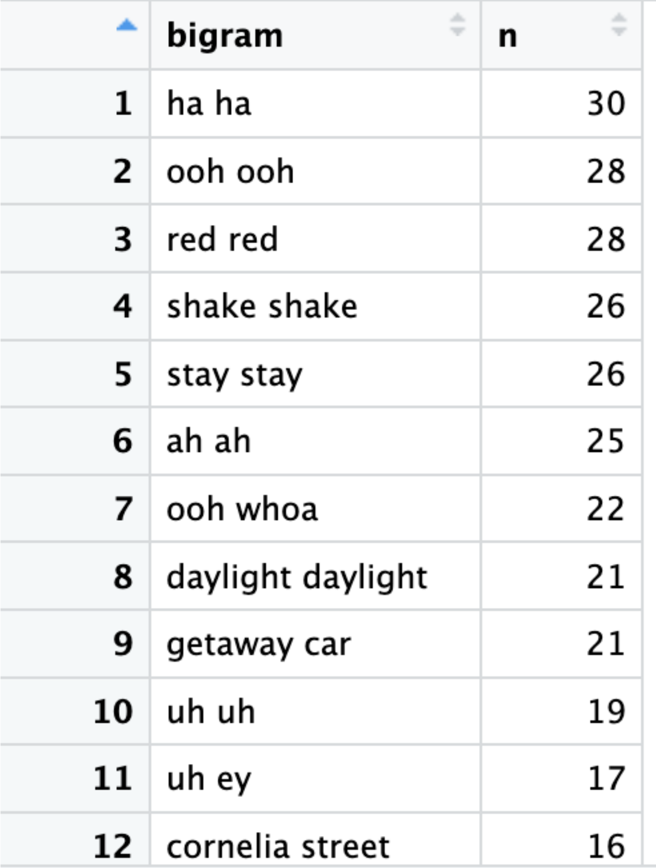


	Artist	Album	Title	bigram
1	Taylor Swift	Taylor Swift	Tim McGraw	blue eyes
2	Taylor Swift	Taylor Swift	Tim McGraw	eyes shinx
3	Taylor Swift	Taylor Swift	Tim McGraw	georgia stars
4	Taylor Swift	Taylor Swift	Tim McGraw	chevy truck
5	Taylor Swift	Taylor Swift	Tim McGraw	gettin stuck
6	Taylor Swift	Taylor Swift	Tim McGraw	tim mcgraw
7	Taylor Swift	Taylor Swift	Tim McGraw	favorite song
8	Taylor Swift	Taylor Swift	Tim McGraw	black dress
9	Taylor Swift	Taylor Swift	Tim McGraw	faded blue
10	Taylor Swift	Taylor Swift	Tim McGraw	blue jeans
11	Taylor Swift	Taylor Swift	Tim McGraw	tim mcgraw
12	Taylor Swift	Taylor Swift	Tim McGraw	thankin god

Top Bigrams

- Because these are song lyrics, a lot of bigrams are the same word repeated
- a lot of sounds as well (ooh ohh, ha ha)
- You could try to filter out sounds as additional stop words

```
tidy_lyrics_bigram_count <-
tidy_lyrics_bigram_united%>%
count(word) %>%
arrange(desc( n))
```



	bigram	n
1	ha ha	30
2	ooh ooh	28
3	red red	28
4	shake shake	26
5	stay stay	26
6	ah ah	25
7	ooh whoa	22
8	daylight daylight	21
9	getaway car	21
10	uh uh	19
11	uh ey	17
12	cornelia street	16