Text analysis

Jason G. Fleischer, Ph.D.

Asst. Teaching Professor

Department of Cognitive Science, UC San Diego

jfleischer@ucsd.edu

@jasongfleischer

https://jgfleischer.com

Examples of questions that require text analysis

1. Did J.K. Rowling write The Cuckoo's Calling under the pen name Robert Galbraith?
2. What themes are common in 19th century literature?
3. Can we tell the difference between tweets that come from Trump himself or a staffer?
4. Is Hillary the most poisoned name in US History?
5. Can we visualize the narrative structure of The Hobbit?
6. Who has the biggest vocabulary in hip hop?
7. Is there a gender imbalance in who gets to say lines in the movies?

A whole bunch of text processing project ideas

- Spam detection (in text messages, on social media)
- Hate speech detection
- Predictive text (like gmail)
- Summarize articles (TL;DR)
- Summarize trends on social media… or in cutting edge NLP research?
- Plagiarism detection… in code projects?
- Document similarity… prevent the posting of duplicate questions on Campuswire?

Today’s example question: How has pop music changed in the last five years?

Goal: Understand the basics of sentiment analysis and TF-IDF

What data would we need to answer this question?

How has pop music changed in the last five years?

Data: Lyrics to the most popular songs from each year

The data : Top songs from Feb music charts 2017-2021

2017: 152 songs

2018: 139 songs

2019: 127 songs

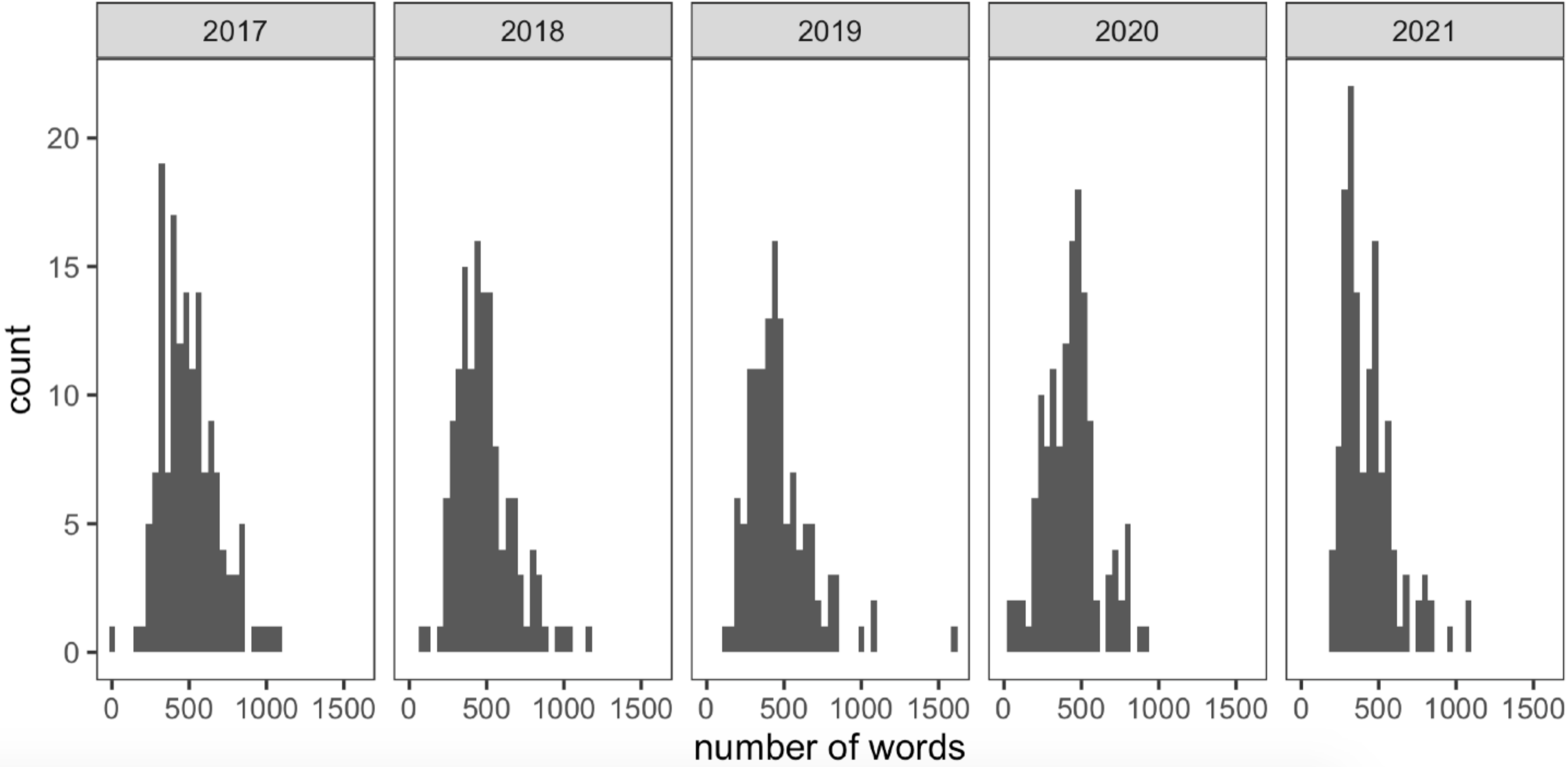
2020: 137 songs

2021: 134 songs

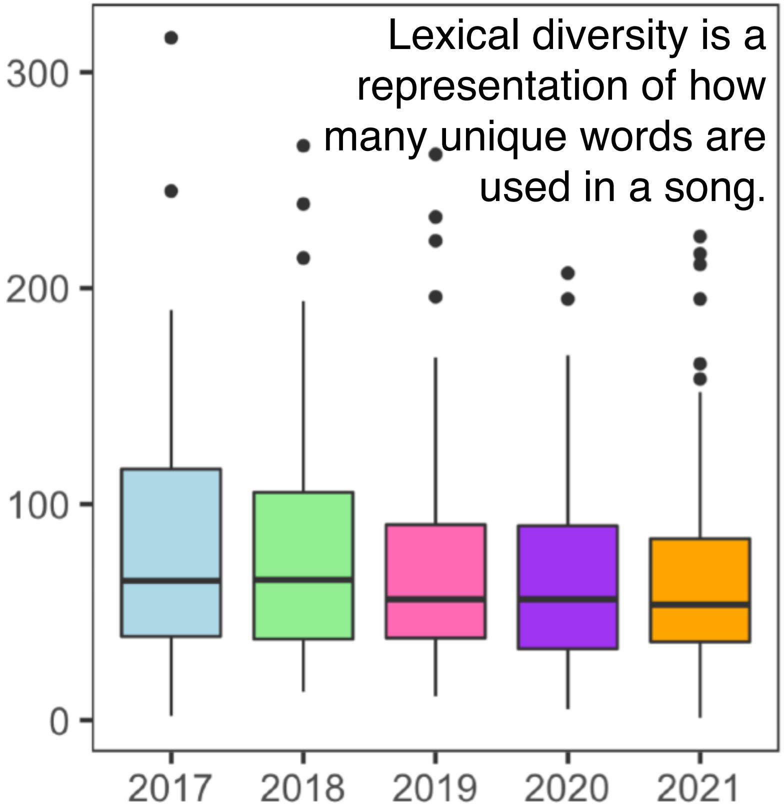
Song data from **Spotify**.
Lyrics from **genius.com**

Questions we can ask...

1. Does the total number of words change over time?
2. Does uniqueness change over time?
3. Does the diversity or density change?
4. What words are most common?
5. What words are most unique to each year?
6. What sentiment do songs convey most frequently?
7. Has sentiment changed over time?
8. What are the sentiment of the #1 songs?
9. What words contribute to the sentiment of these #1 songs?
10. ...what about bigrams? N-grams?

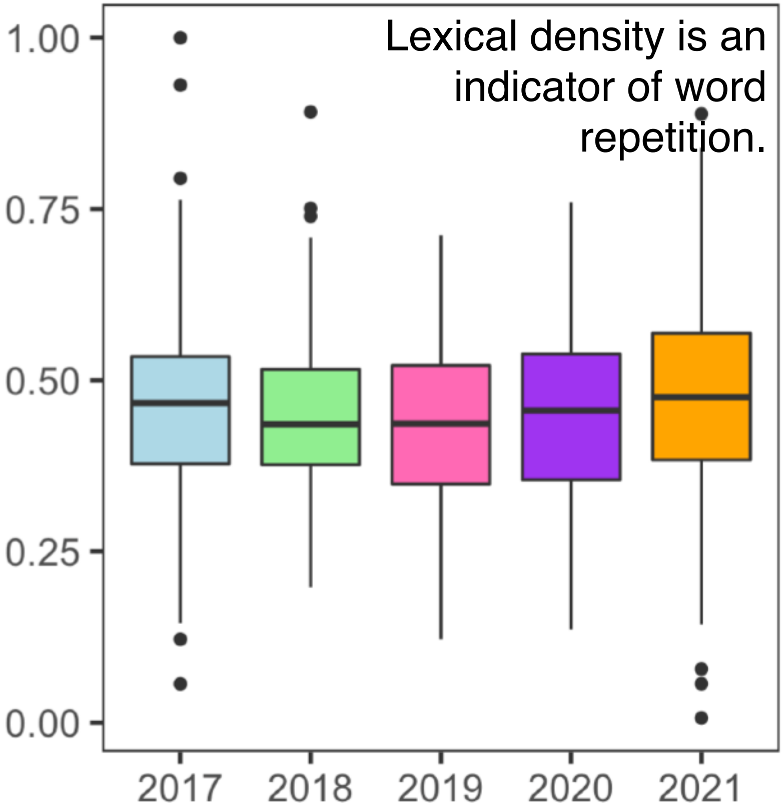

2017
2018
2019
2020
2021
count
0
5
10
15
20
0
500
1000
1500
number of words

Lexical Diversity



Lexical diversity is a representation of how many unique words are used in a song.

Lexical Density

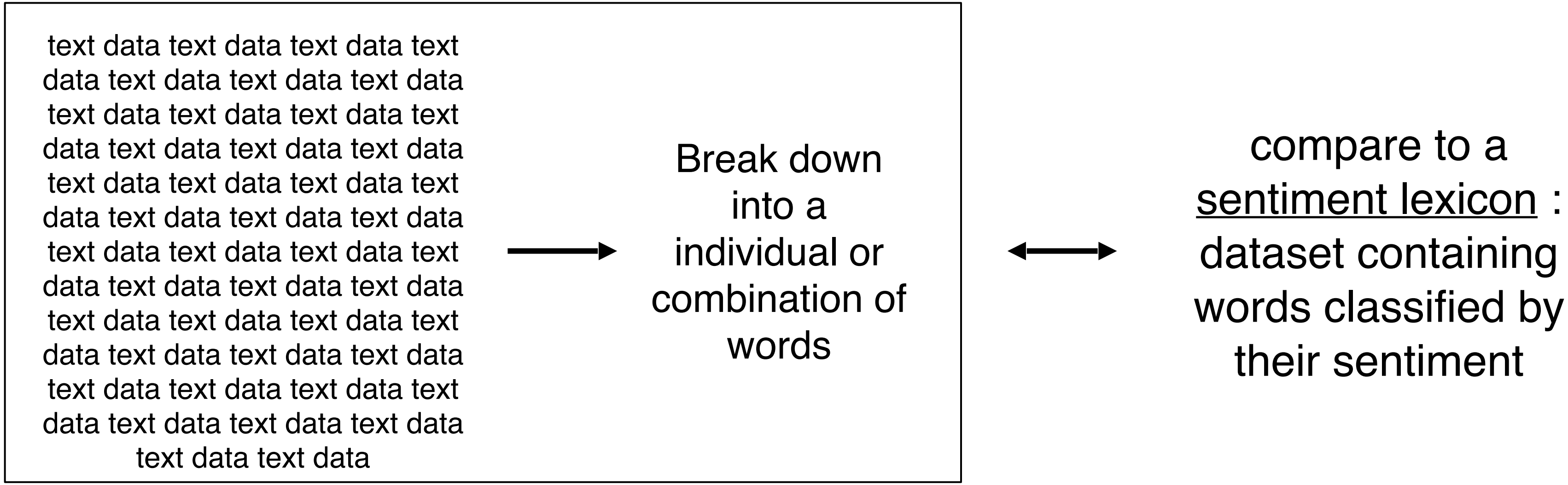


Lexical density is an indicator of word repetition.

Sentiment Analysis

Sentiment Analysis

Programmatically infer emotional content of text

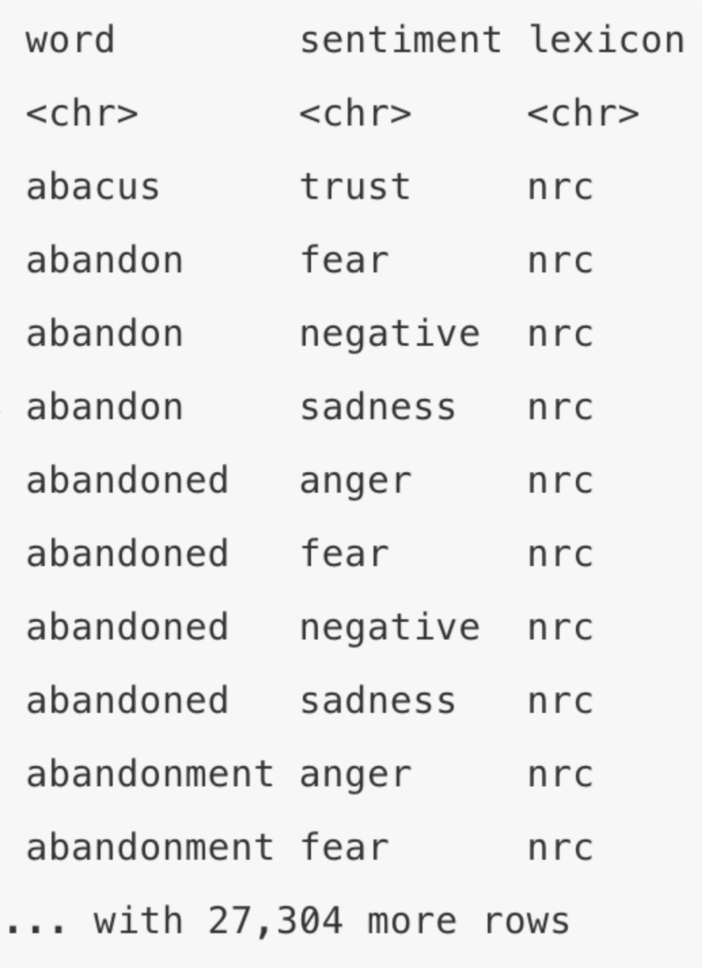


text data text data

Break down into a individual or combination of words

compare to a <u>sentiment lexicon</u> : dataset containing words classified by their sentiment

Part of the “NRC” sentiment **lexicon**:



```
word        sentiment lexicon
<chr>       <chr>     <chr>
abacus      trust     nrc
abandon     fear      nrc
abandon     negative  nrc
abandon     sadness   nrc
abandoned   anger     nrc
abandoned   fear      nrc
abandoned   negative  nrc
abandoned   sadness   nrc
abandonment anger     nrc
abandonment fear      nrc
... with 27,304 more rows
```

http://varianceexplained.org/r/trump-tweets/

When doing sentiment analysis...

token - a meaningful unit of text

- what you use for analysis
- *tokenization* takes corpus of text and splits it into tokens (words, bigrams, etc.)

Check out https://neptune.ai/blog/tokenization-in-nlp

stop words - words not helpful for analysis

- extremely common words such as “a”,“the”, “of”, “to”
- are typically removed from analysis

When doing sentiment analysis...

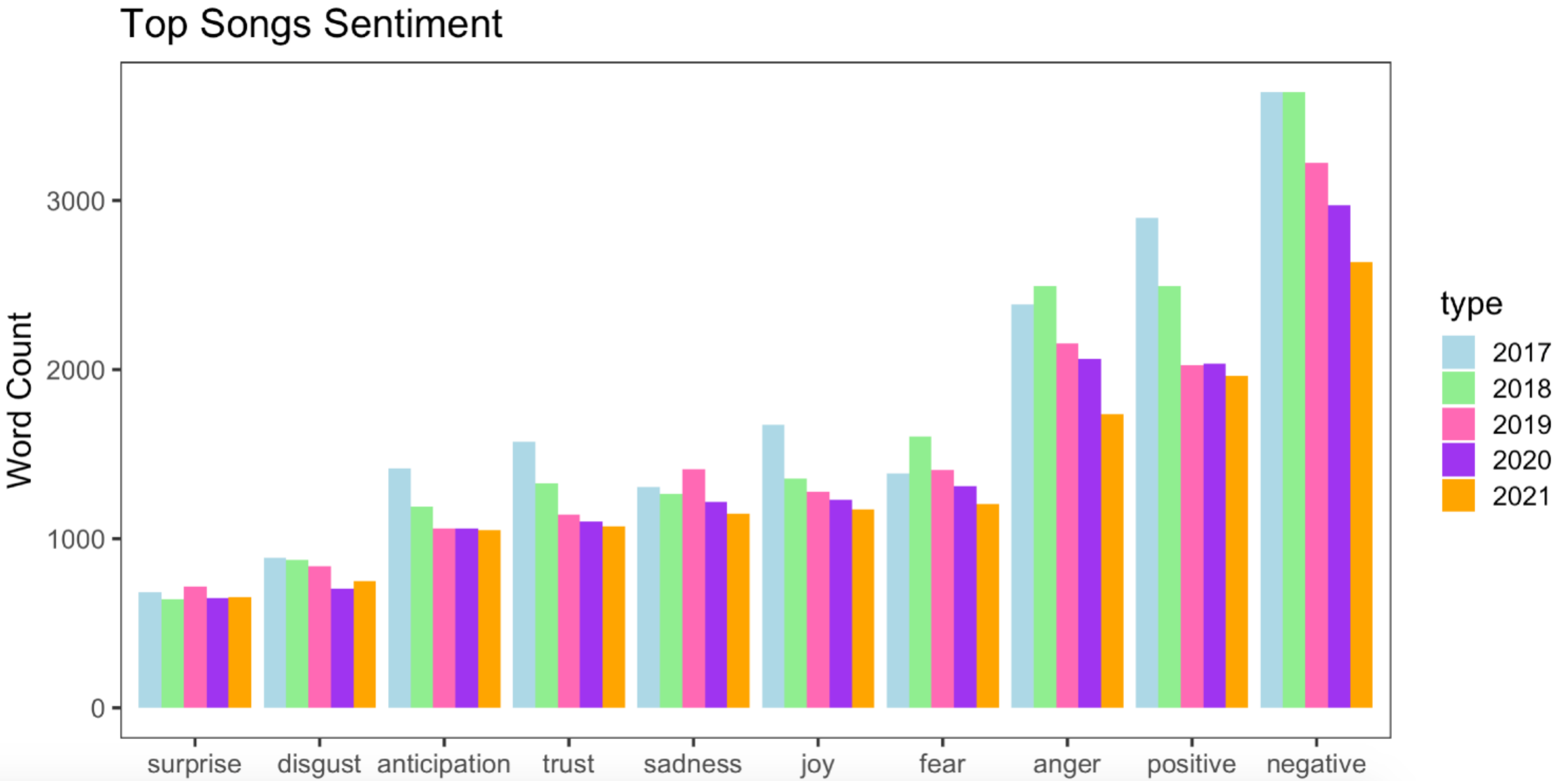
stemming or lemmatization

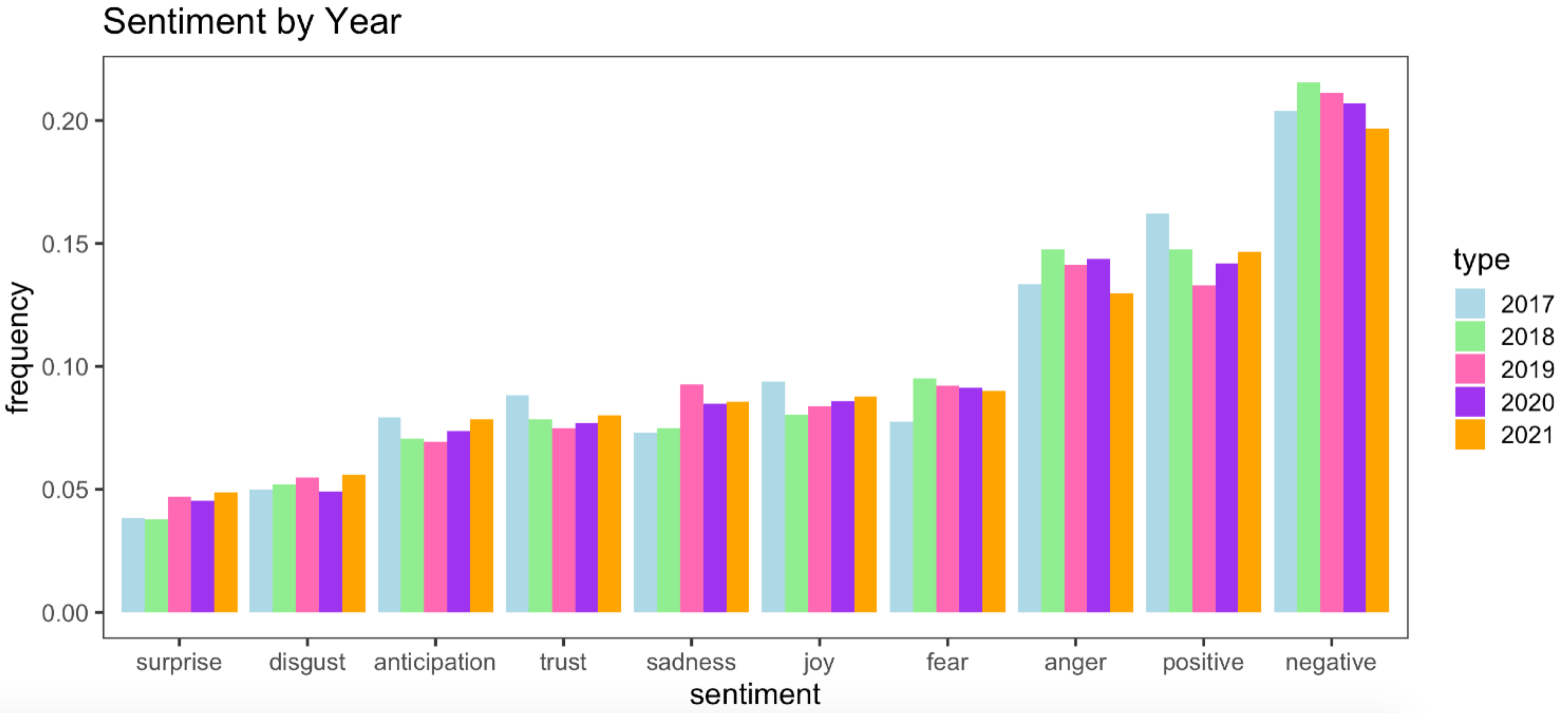
- Identifying the root for each token
- Jumping, jumped, jumps, jump all have the same root ‘jump’
- Where things get tricky: jumper???
- Stemmers and lemmatizers do the same thing, stemmers are word focussed/cruder and lemmatizers try to take into account the context to correctly stem jumper differently depending on if its “I love my new jumper” (In the US jumper=a kind of dress, in the UK jumper=sweater) or its “Shane is a long jumper”

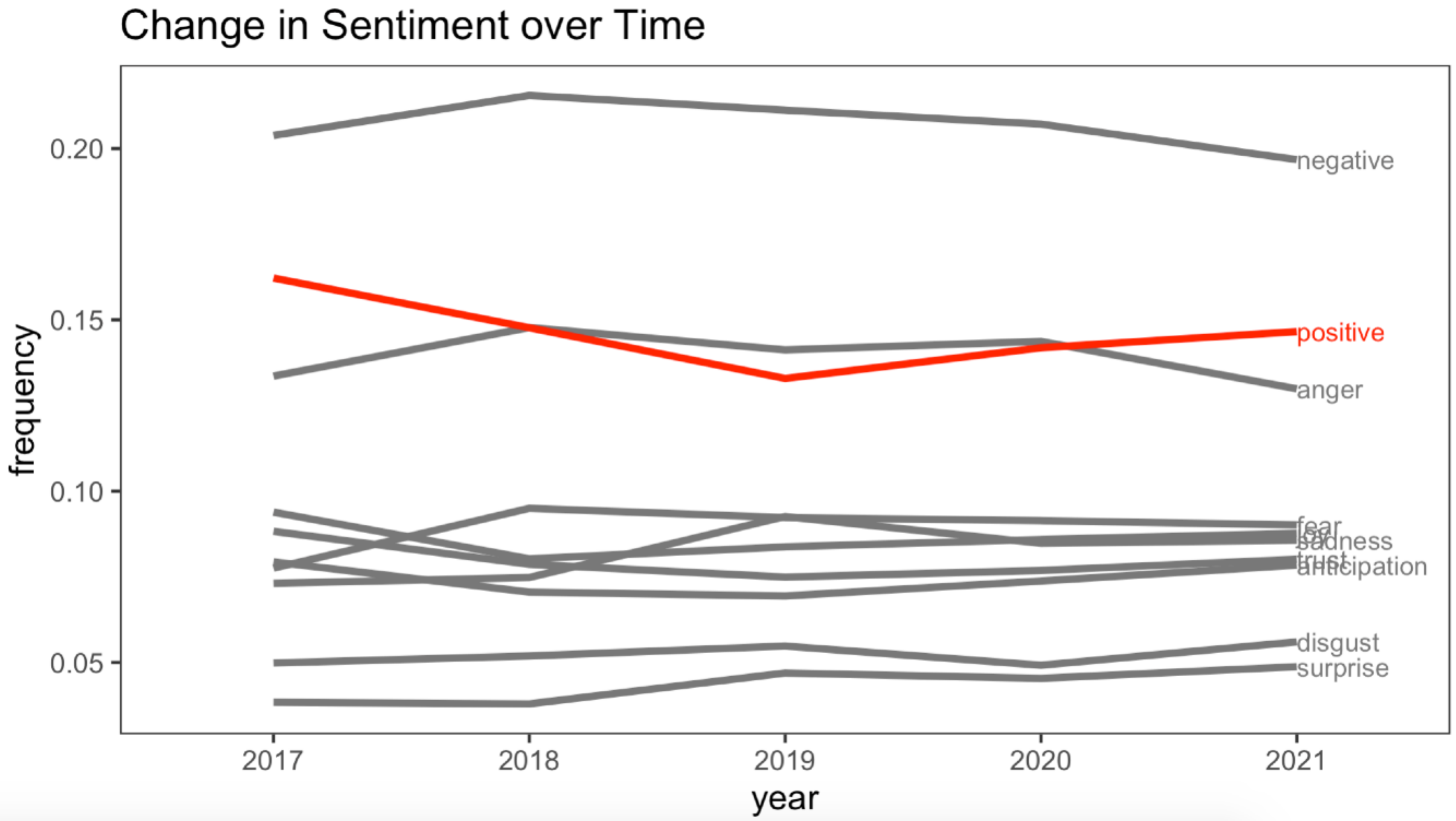
Check out https://www.datacamp.com/tutorial/stemming-lemmatization-python

In text analysis, your choices matter:

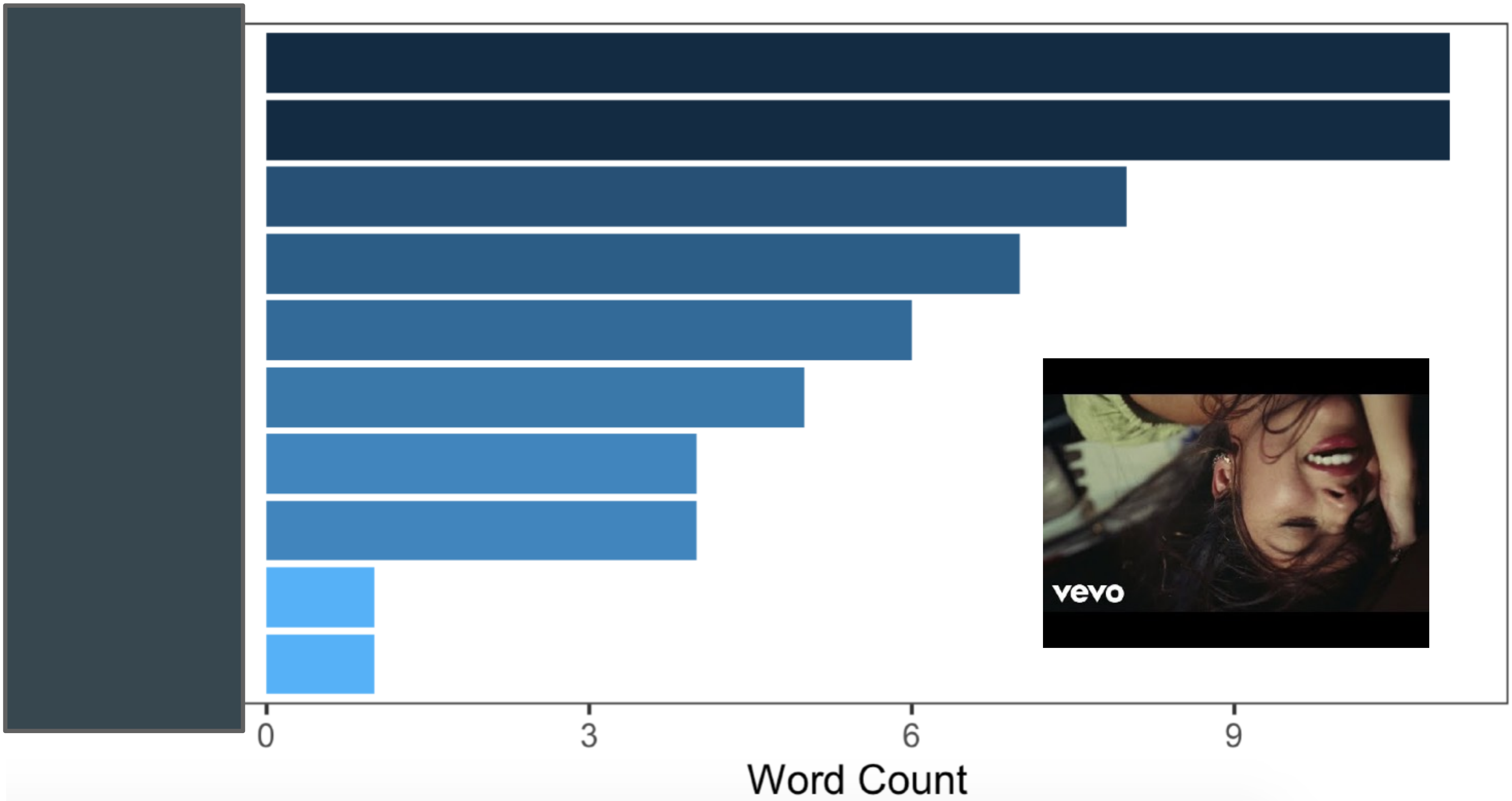
1. How to tokenize?
2. Remove stop words? Remove common words?
3. Use stemming/lemmatization?
4. What sentiment lexicon to use?


Top Songs Sentiment
Word Count
3000
2000
1000
0
surprise
disgust
anticipation
trust
sadness
joy
fear
anger
positive
negative
type
2017
2018
2019
2020
2021


Sentiment by Year
frequency
0.20
0.15
0.10
0.05
0.00
surprise
disgust
anticipation
trust
sadness
joy
fear
anger
positive
negative
sentiment
type
2017
2018
2019
2020
2021


Change in Sentiment over Time

Sentiment: Driver's License

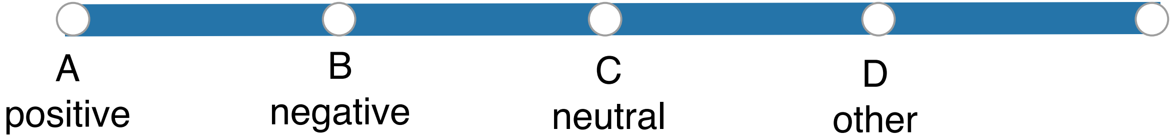

[redacted]

I got my driver's license last week
Just like we always talked about
'Cause you were so excited for me
To finally drive up to your house
But today I drove through the suburbs
Crying 'cause you weren't around
And you're probably with that blonde girl
Who always made me doubt
She's so much older than me
She's everything I'm insecure about
Yeah, today I drove through the suburbs
'Cause how could I ever love someone else?
And I know we weren't perfect but I've never felt this way for no one
And I just can't imagine how you could be so okay now that I'm gone
Guess you didn't mean what you wrote in that song about me
'Cause you said forever, now I drive alone past your street
And all my friends are tired
Of hearing how much I miss you, but
I kinda feel sorry for them
'Cause they'll never know you the way that I do, yeah
Today I drove through the suburbs
And pictured I was driving home to you

Sentiment Limitations

How would you classify the sentiment of the following sentence?

“The idea behind the movie was great, but it could have been better”



A
positive

B
negative

C
neutral

D
other

“Dude!”

“So bad!”

The limitations of sentiment analysis:
Context, homonyms, pragmatics, parts of speech

Mid, Shade

The limitations of sentiment analysis:
Lexicon not up to the task, lexicon specific results

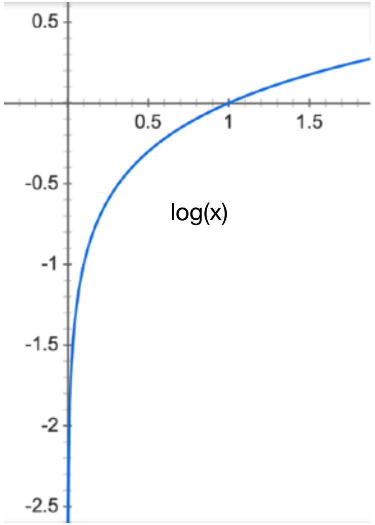
TF-IDF:
Term frequency - Inverse document frequency

Given a corpus of documents, figure out which words are uniquely important to each document

TF-IDF:
Term Frequency - Inverse Document Frequency

Term Frequency (TF) : how frequently a word occurs in a document

Inverse document frequency (IDF) : how important a word is to a document



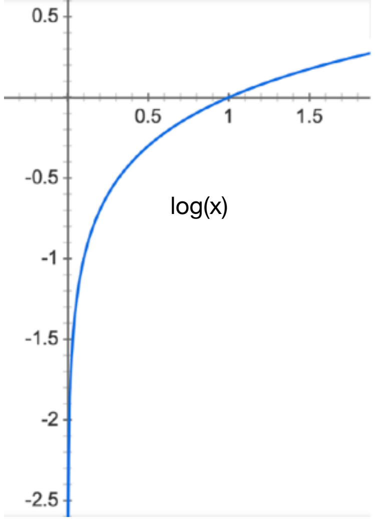
$$idf(\text{term}) = \ln\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents

TF-IDF:

Term Frequency - Inverse Document Frequency

the frequency of a term adjusted for how rarely it is used



$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term *x* within document *y*

$tf_{x,y}$ = frequency of *x* in *y*

df_x = number of documents containing *x*

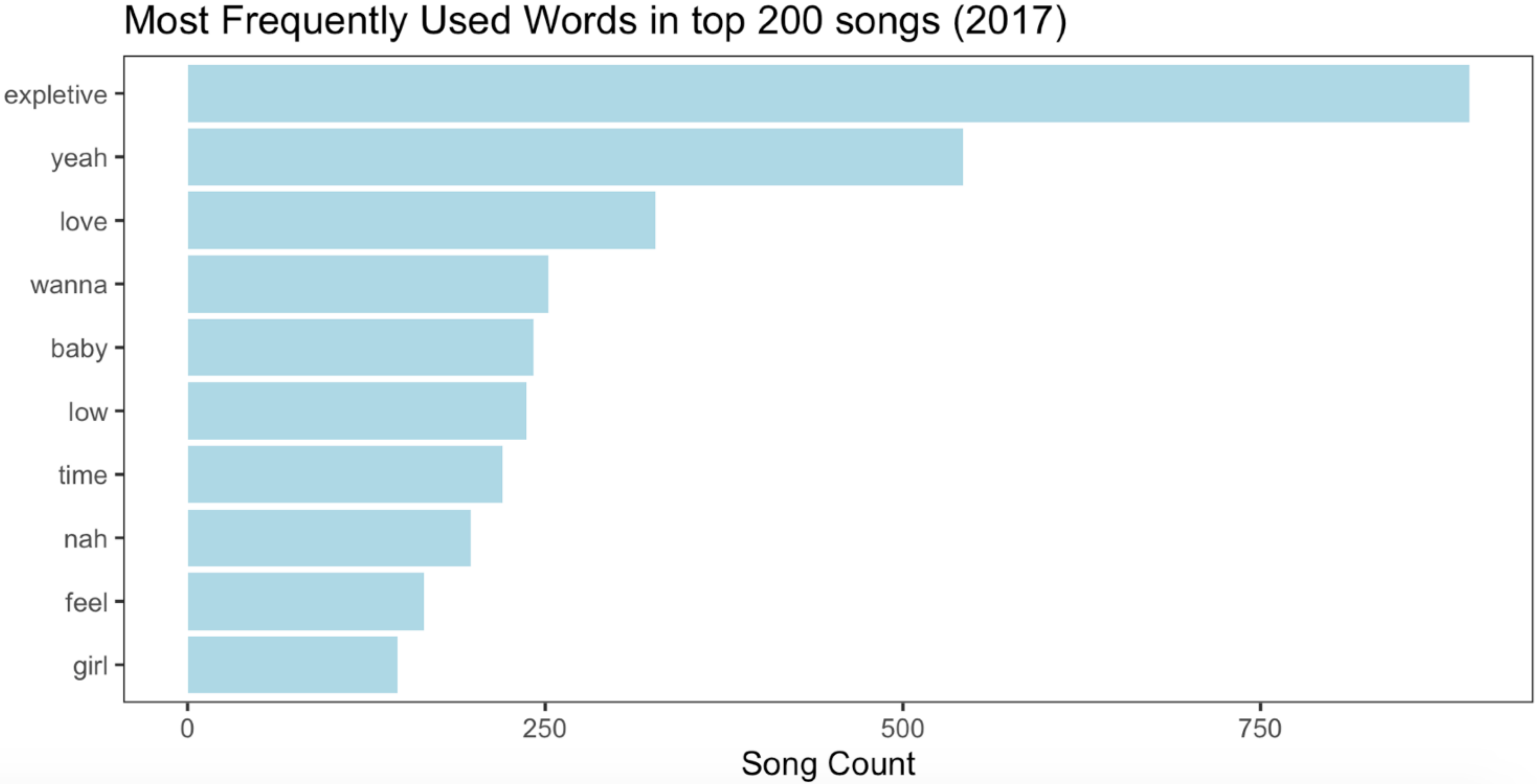
N = total number of documents

https://www.tidytextmining.com

What words are the most unique to the lyrics of each year's top hits?

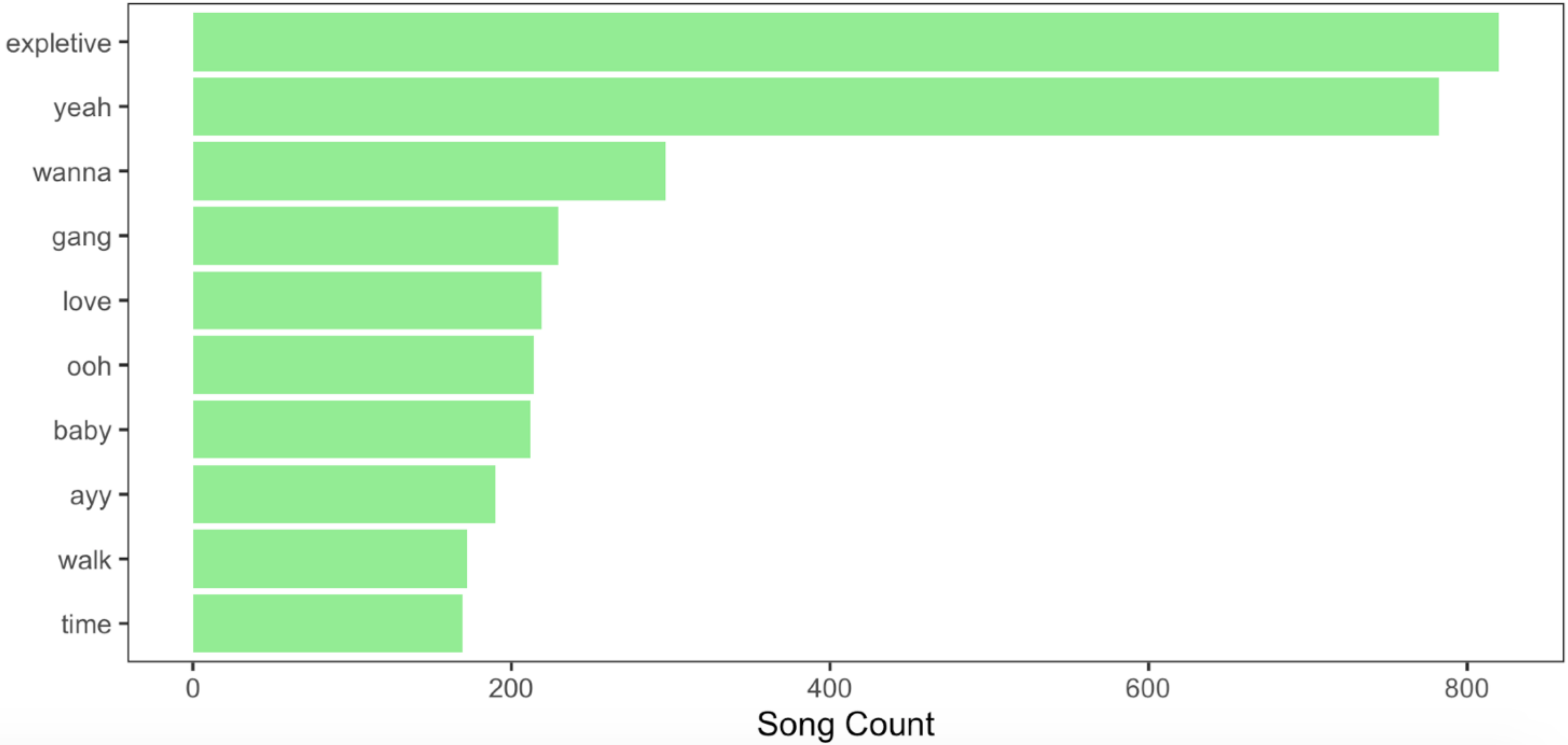
Goal: to use TF-IDF to *find the important words* for the content of each document by decreasing the weight for commonly used words and increasing the weight for words that are not used very much in a collection or corpus of documents

Calculating TF-IDF attempts to find the words that are important (i.e., common) in a text, but not *too* common

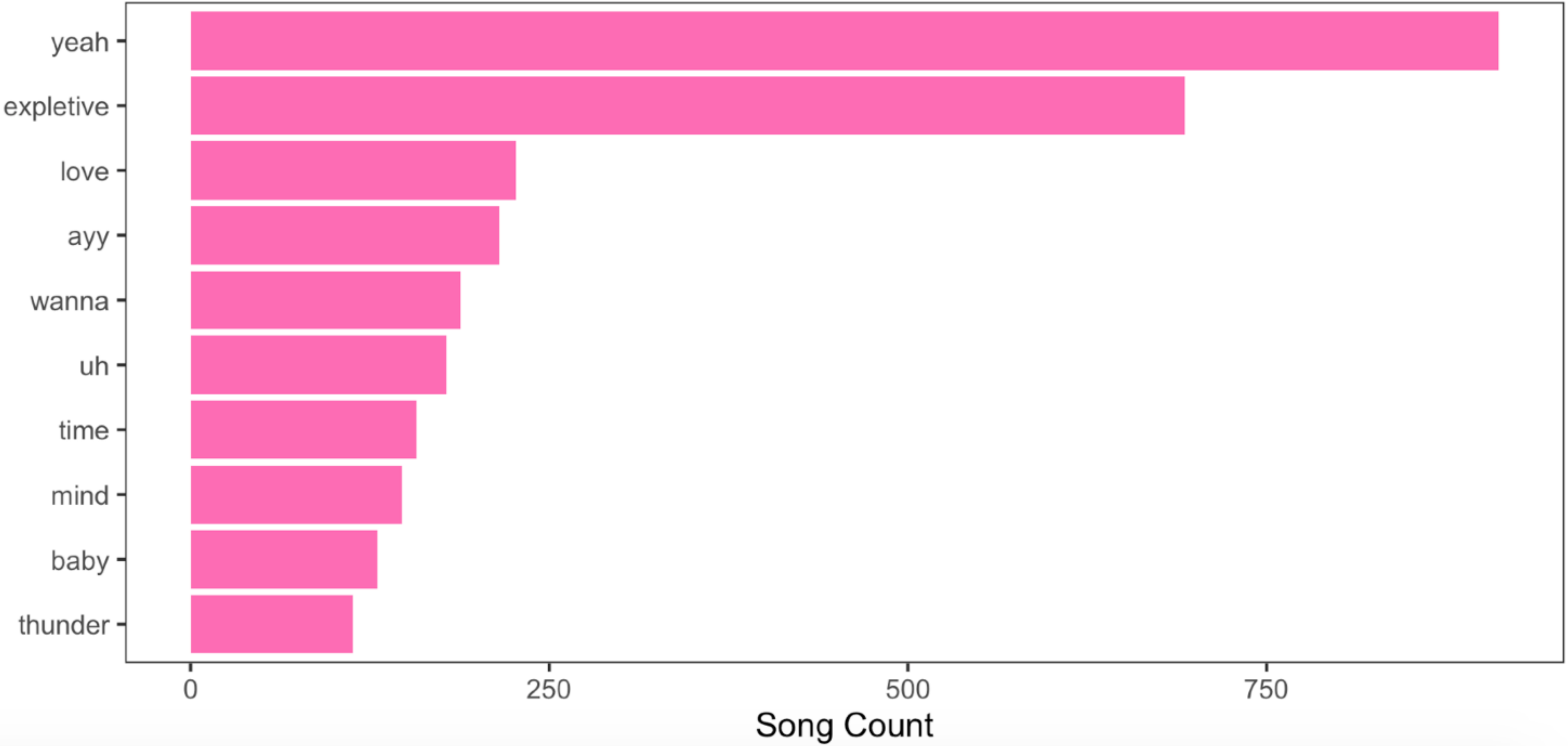


Most Frequently Used Words in top 200 songs (2017)

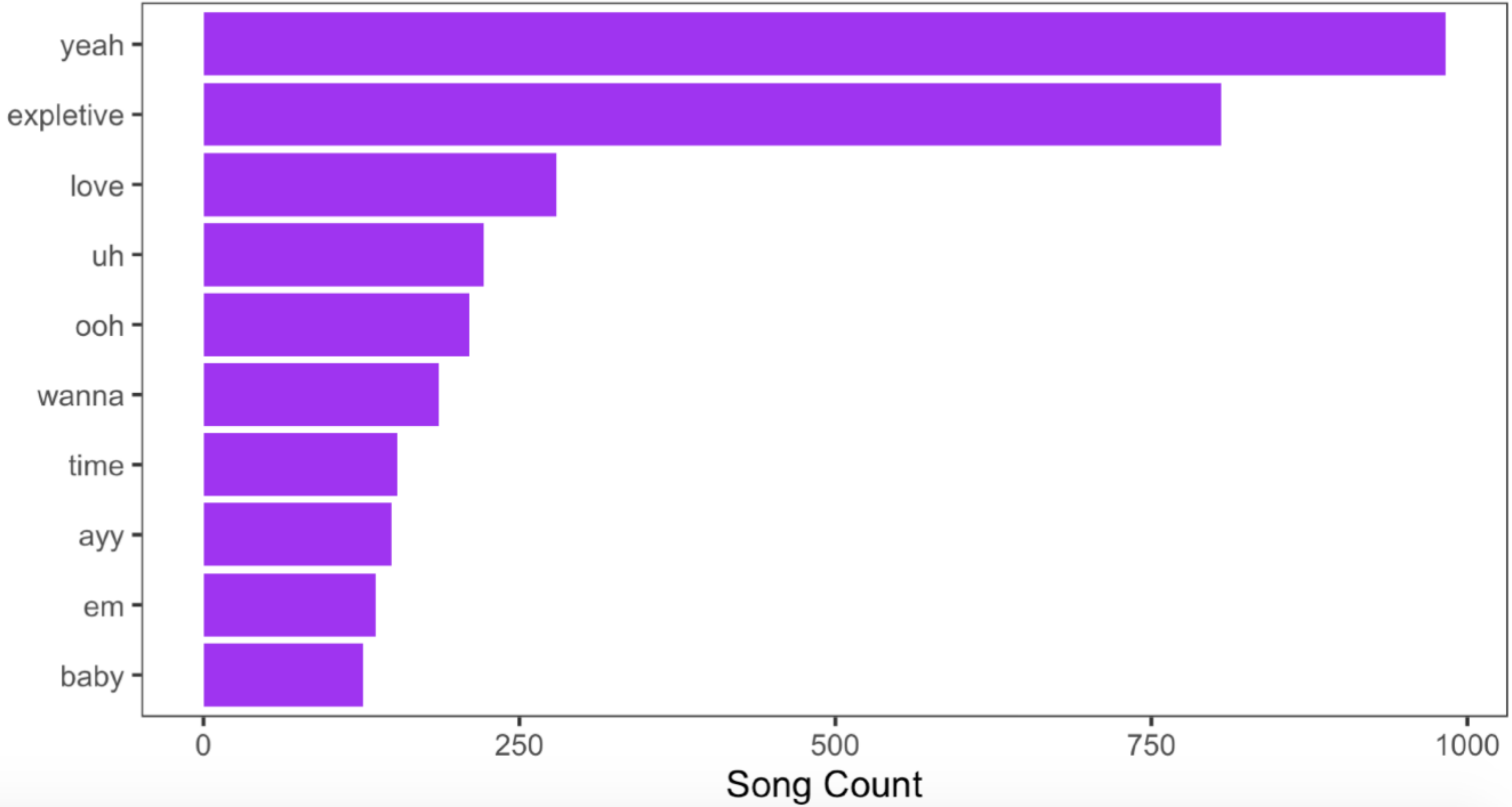
Most Frequently Used Words in top 200 songs (2018)

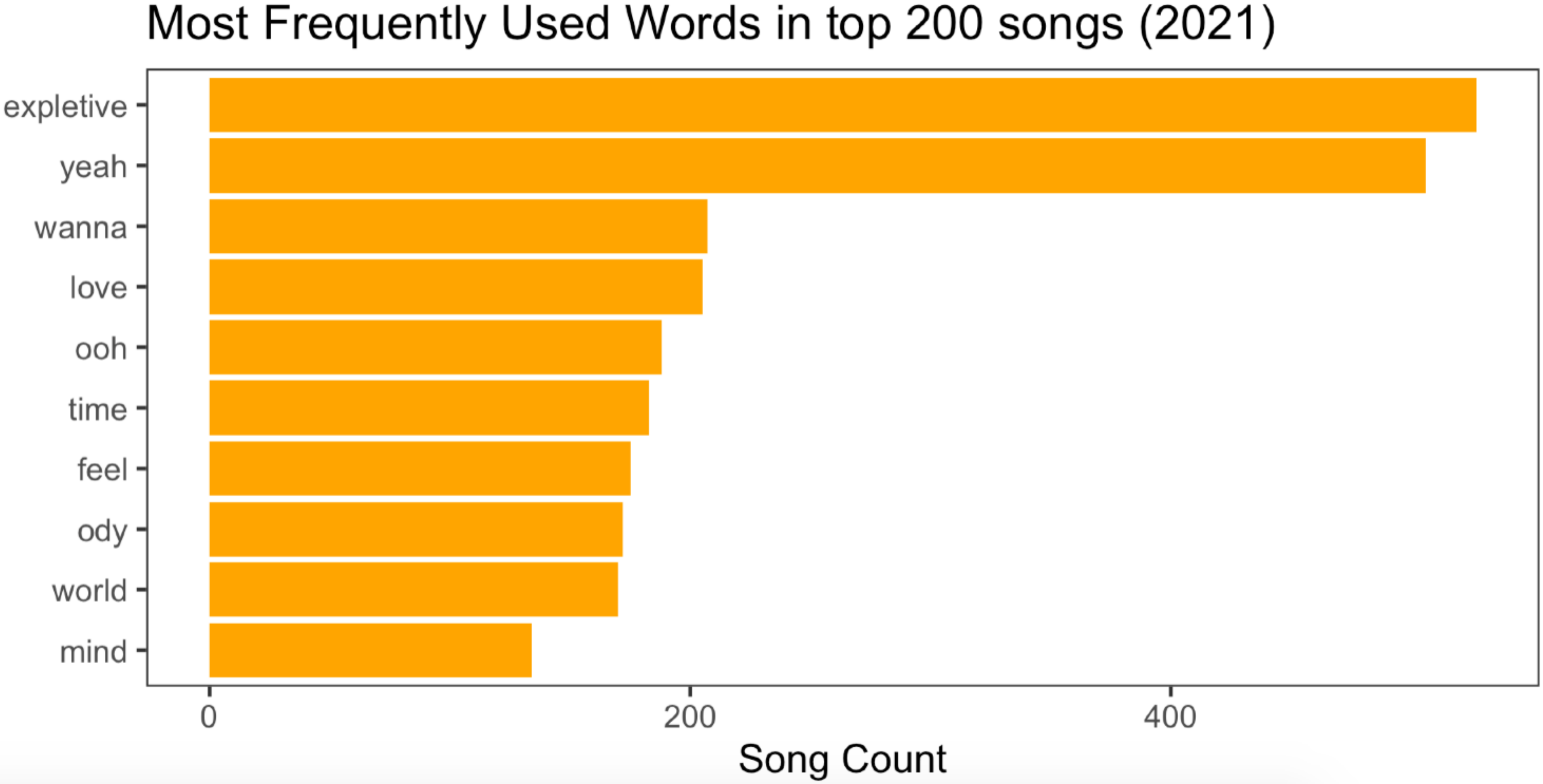


Most Frequently Used Words in top 200 songs (2019)

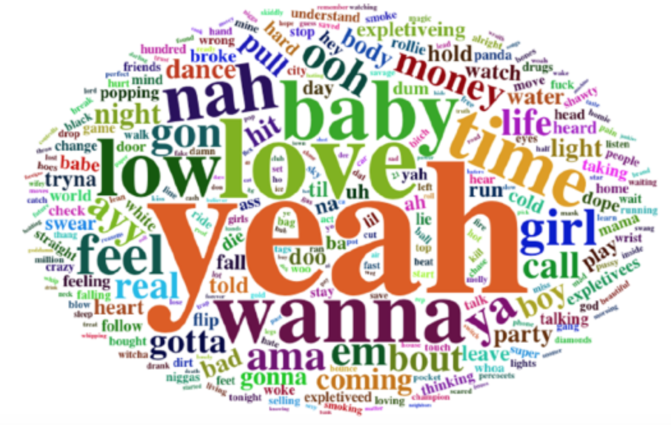


Most Frequently Used Words in top 200 songs (2020)

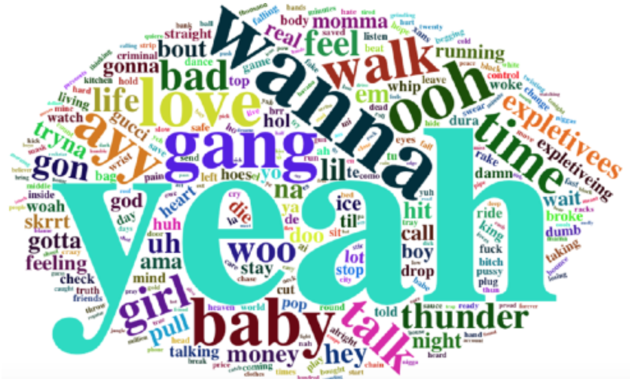



Most Frequently Used Words in top 200 songs (2021)
expletive
yeah
wanna
love
ooh
time
feel
ody
world
mind
0
200
400
Song Count

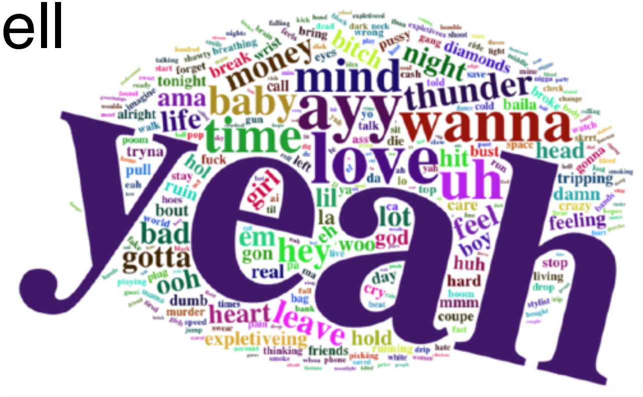
Term Frequency can only tell us so much….



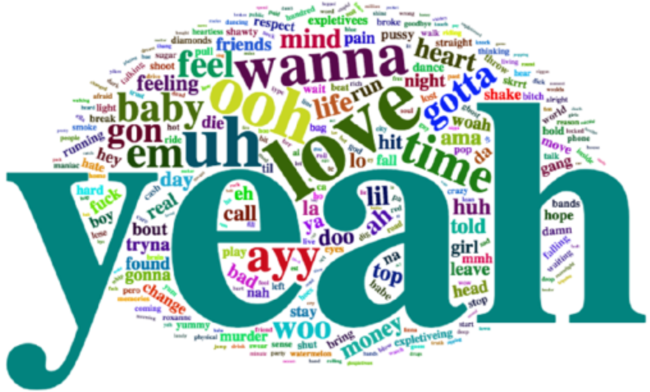
2017



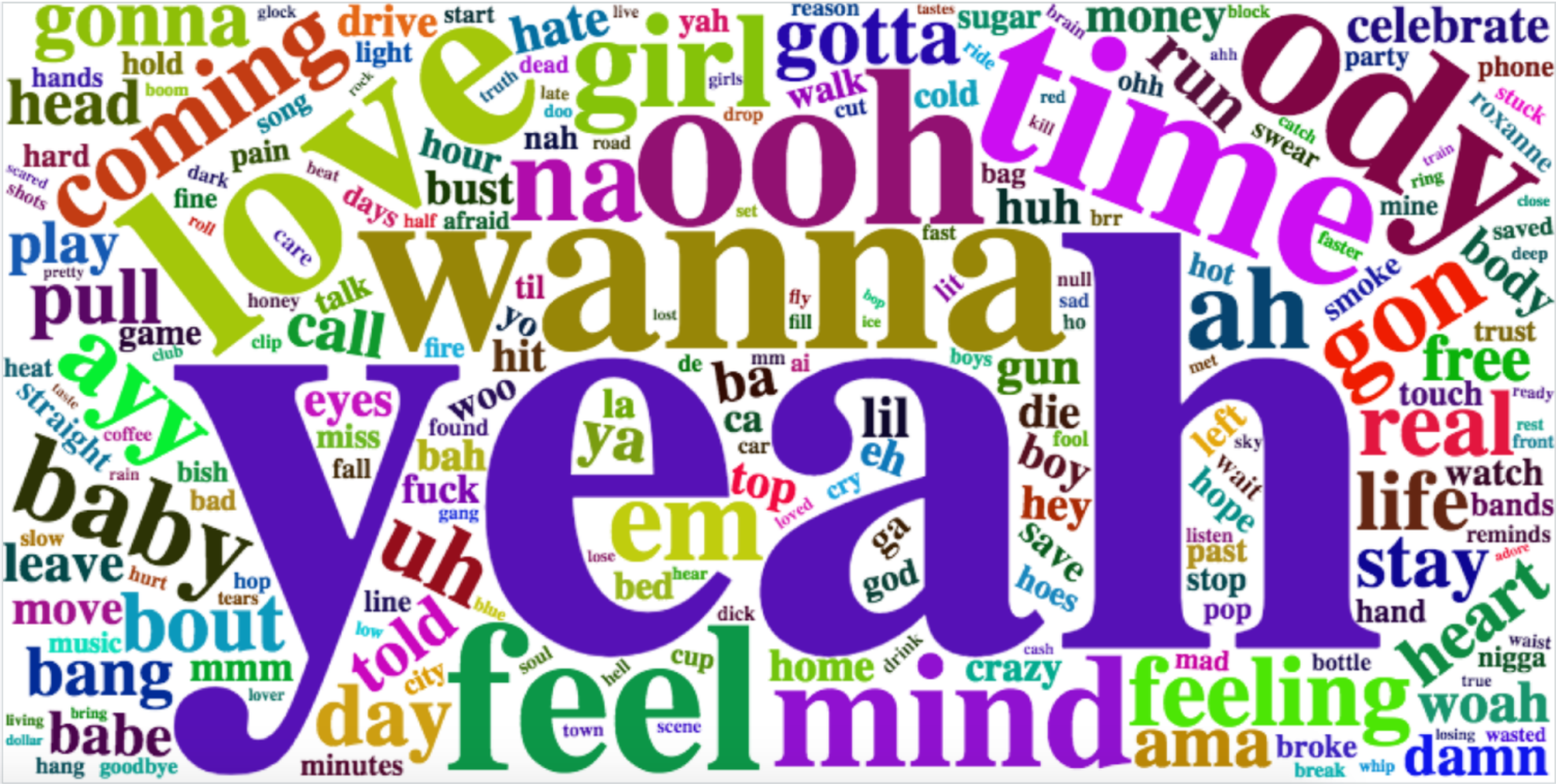
2018



2019



2020


gonna
glock
drive
start
love
hate
live
girl
yah
reason
gotta
tastes
sugar
brain
money
block
celebrate
time
ody
hands
hold
light
coming
head
boom
rock
truth
dead
late
girls
ride
ahh
party
phone
song
doo
drop
walk
cut
cold
ooh
run
ohh
red
kill
stuck
roxanne
hard
hour
nah
road
catch
swear
train
scared
shots
dark
pain
beat
na
bag
ring
close
days
half
afraid
bust
set
huh
brr
mine
roll
care
play
fast
faster
saved
deep
pretty
wanna
body
pull
honey
talk
til
lit
hot
smoke
null
sad
lost
fly
fill
bop
ice
ah
gon
game
call
yo
ho
trust
clip
club
ayy
fire
hit
de
mm
ai
boys
gun
free
heat
met
touch
ready
straight
taste
eyes
woo
ba
la
lil
ca
die
real
rest
front
coffee
found
ya
yeah
miss
bah
car
eh
fool
left
sky
boy
rain
fall
top
cry
wait
watch
life
bish
baby
fuck
hey
hope
bands
bad
gang
loved
em
ga
listen
reminds
slow
uh
save
past
adore
leave
hurt
lose
hear
god
stay
hop
bed
hoes
stop
tears
line
told
dick
blue
move
bout
pop
hand
heart
low
music
cash
waist
drink
nigga
bang
mmm
soul
cup
home
crazy
mad
bottle
city
hell
feeling
true
woah
lover
feel
mind
living
bring
babe
town
scene
dollar
ama
broke
losing
wasted
hang
goodbye
minutes
damn
break
whip

2021

TF-IDF:
Term Frequency - Inverse Document Frequency

Term Frequency (TF) : how frequently a word occurs in a document

Inverse document frequency (IDF) : intended to measure how important a word is to a document

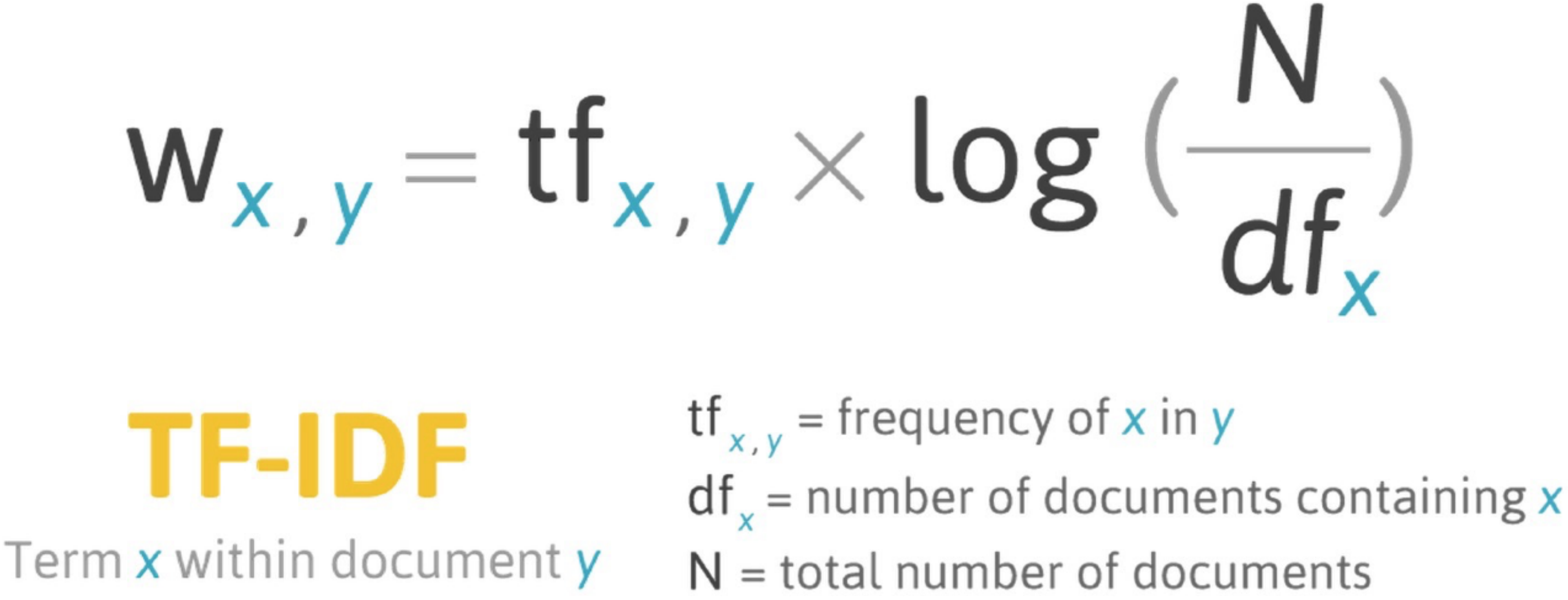
decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents

$$idf(\text{term}) = \ln\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

TF-IDF:

Term Frequency - Inverse Document Frequency

the frequency of a term adjusted for how rarely it is used



$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

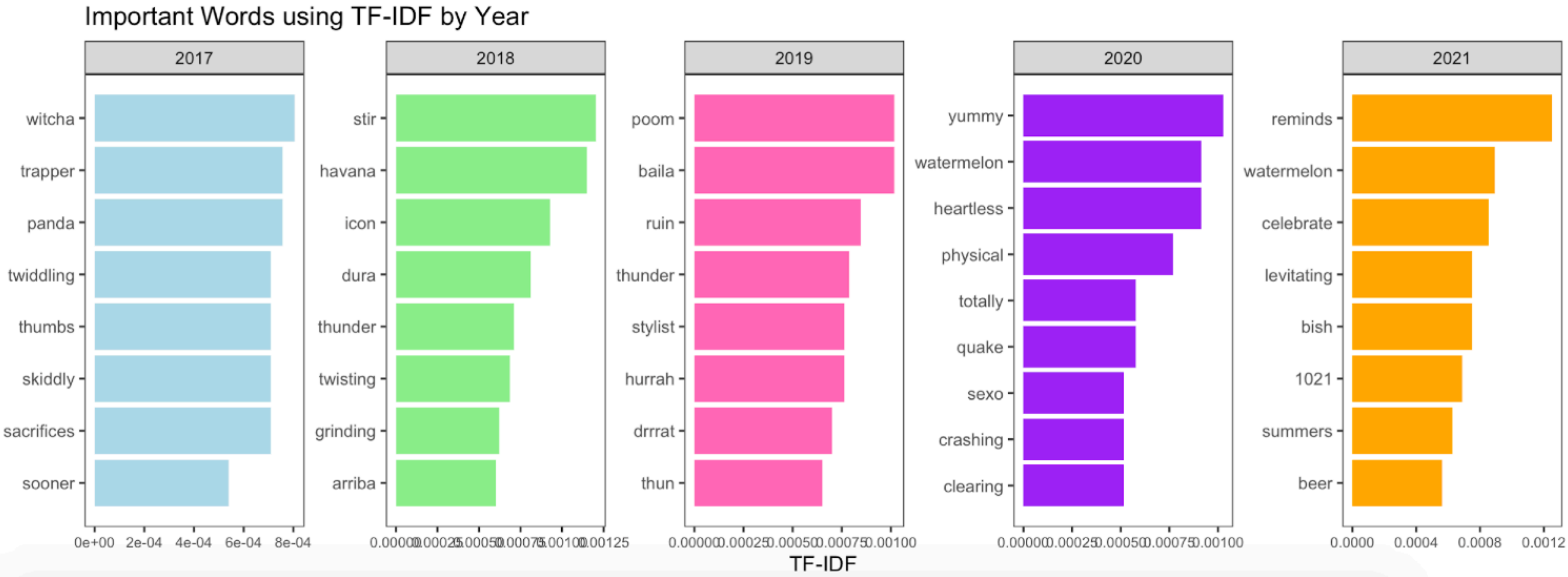
Term x within document y

$tf_{x,y}$ = frequency of x in y

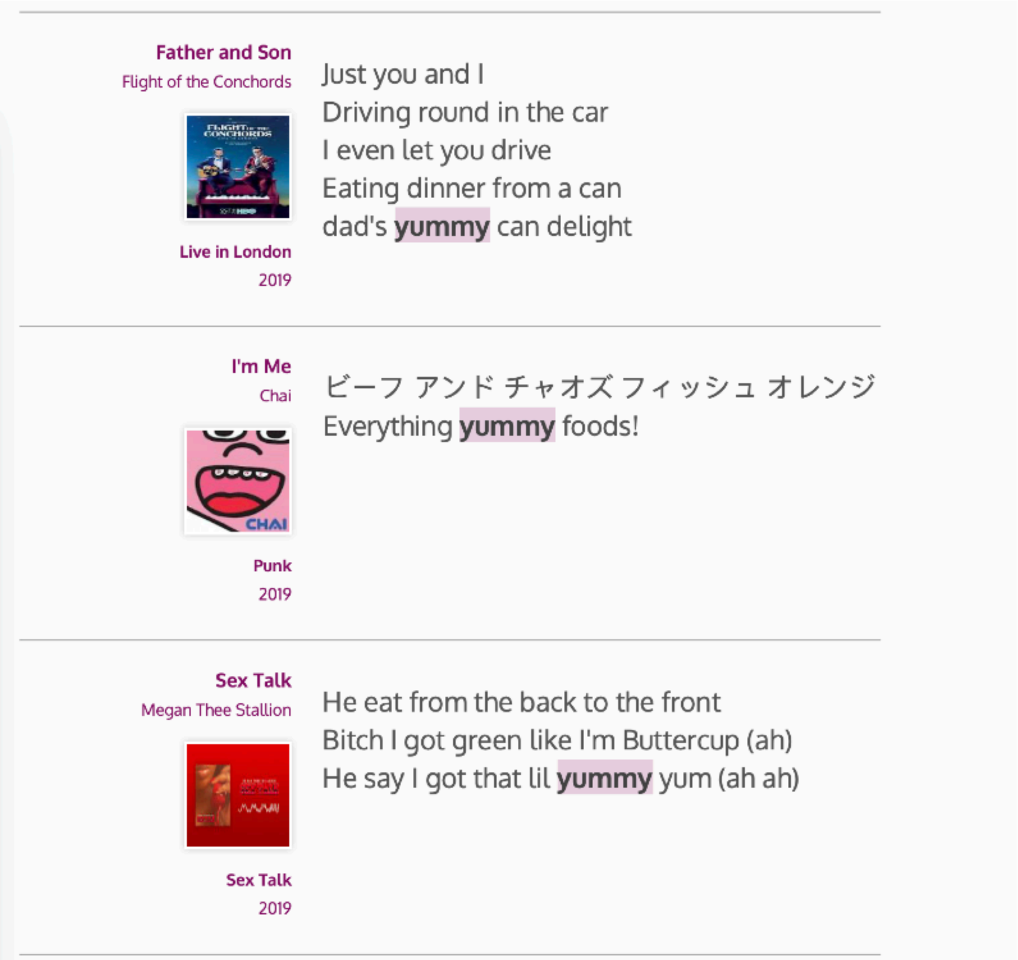
df_x = number of documents containing x

N = total number of documents

https://www.tidytextmining.com



Important Words using TF-IDF by Year



Father and Son
Flight of the Conchords

Live in London
2019

Just you and I
Driving round in the car
I even let you drive
Eating dinner from a can
dad's **yummy** can delight

I'm Me
Chai

Punk
2019

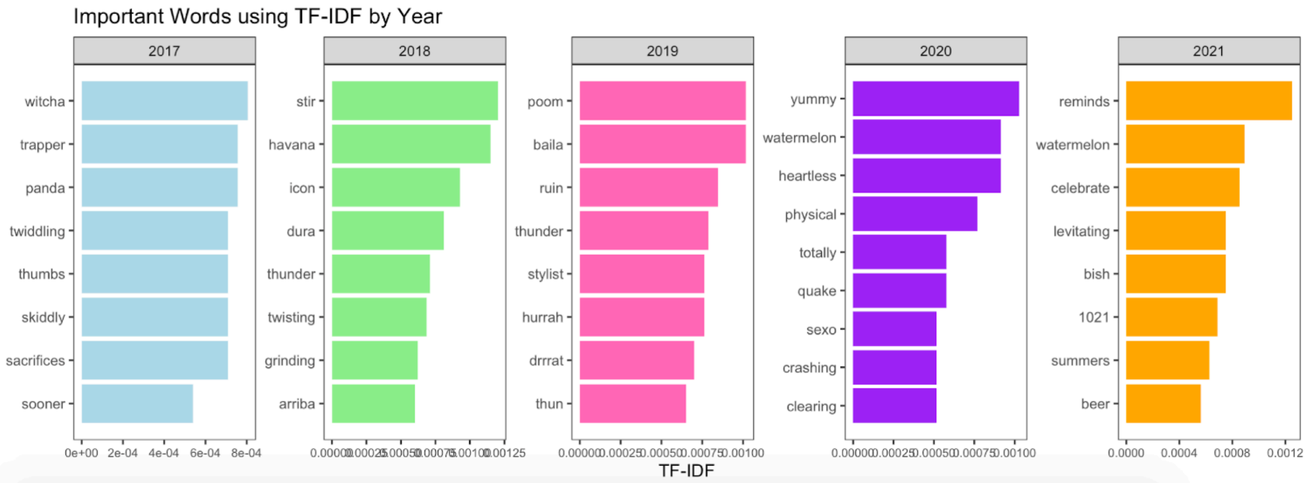
ビーフ アンド チャオズ フィッシュ オレンジ
Everything **yummy** foods!

Sex Talk
Megan Thee Stallion

Sex Talk
2019

He eat from the back to the front
Bitch I got green like I'm Buttercup (ah)
He say I got that lil **yummy** yum (ah ah)

What can you conclude from this TF-IDF plot?



A No words overlap across the years in these data

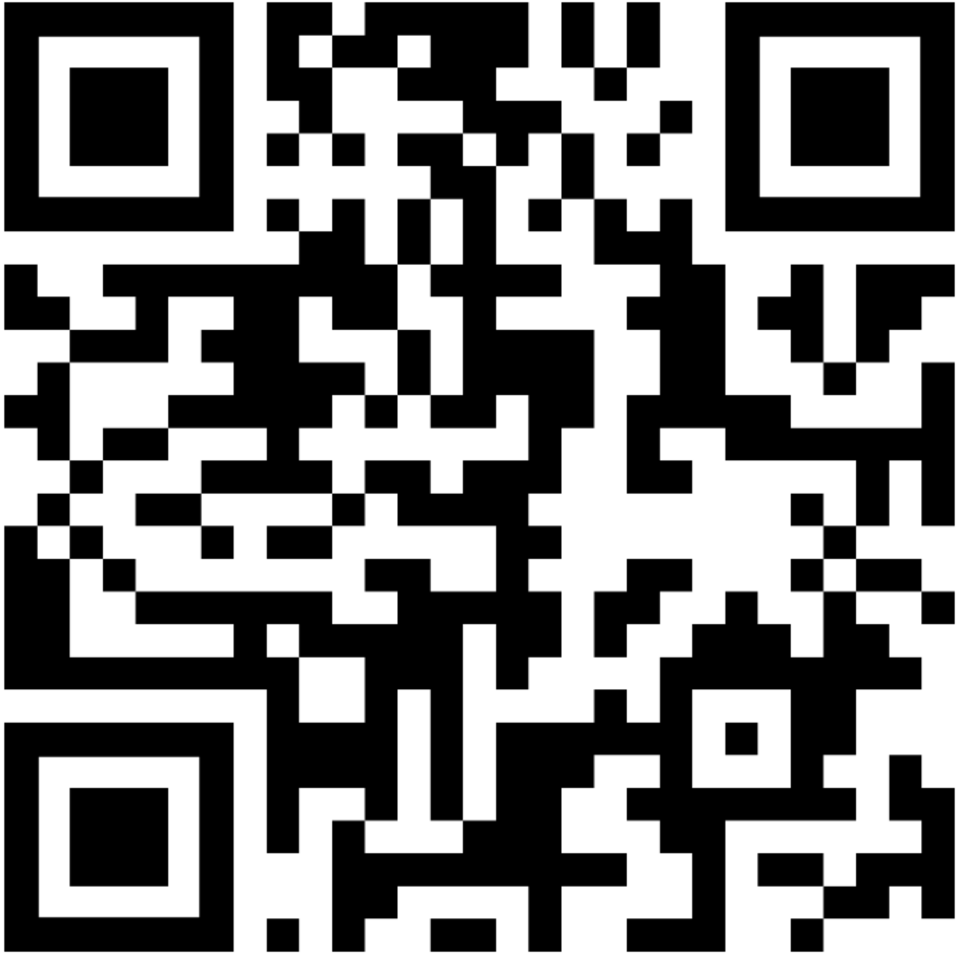
B ‘reminds’ and ‘watermelon’ are the most unique words to the 2021 data

C ‘watermelon’ is the most common word in this dataset

D A-C (all of the above)

E None of the above

https://forms.gle/uD8cVTUUzAsxcVvKA



Questions we can ask...

1. Does the total number of words change over time?
2. Does uniqueness change over time?
3. Does the diversity or density change?
4. What words are most common?
5. What words are most unique to each year?
6. What sentiment do songs convey most frequently?
7. Has sentiment changed over time?
8. What are the sentiment of the #1 songs?
9. What words contribute to the sentiment of these #1 songs?
10. ...what about bigrams? N-grams?

EDA

TF-IDF

Sentiment Analysis

Text analysis

The big picture

- Identify the problem you are trying to solve… what you want to answer suggests the approaches you will use below
- Clean and preprocess the data
- Linguistic transformations
    - Tokenization, lemmatization, stemming, remove stop words
    - [optional] part of speech tagging, named entity recognition
- EDA
- [optional] Calculate measurements and statistics
    - e.g., diversity, density, sentiment, TF-IDF
- [optional] Transform representation
    - Bag of words, vector embeddings
- [optional] Train a ML system or write an algorithm to accomplish a task given the representation
    - e.g., detect spam