COGS138: Neural Data Science

Lecture 3

C. Alex Simpkins, PhD
UCSD Dept. of Cognitive Science, Spring 2023
http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23

rdprobotics@gmail.com | csimpkinsjr@ucsd.edu

(Based on a course created by Prof. Bradley Voytek)

Plan for today

- Announcements
- Review - Last time
- Neural data science data modalities - EEG, MEG, Text/speech, motion/behavior, others next time
- Tools for Neural data science:
  - MNE
  - NLTK
  - PyMO
- Motion capture technology discussion, relevance, issues to be aware of

Announcements

- FinAID survey
- A0 - due Friday
- A1 - due a week from release, which will be tonight or tomorrow
- Reading 1 - Released on canvas and in web site password protected area tonight, lecture quiz due next week

Last time

Course links

Website	http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23	Main face of the course and everything will be linked from here. Lectures, Readings, Handouts, Files, links
GitHub	https://github.com/drsimpkins-teaching	files/data, additional materials & final projects
datahub	https://datahub.ucsd.edu	assignment submission
Piazza	https://piazza.com/ucsd/spring2023/cogs138_sp23_a00/home (course code on canvas home page)	questions, discussion, and regrade requests
Canvas	https://canvas.ucsd.edu/courses/44897	grades, lecture videos
Anonymous Feedback	Will be able to submit via google form	if I ever offend you, use an example you are uncomfortable with, or to provide general feedback. Please remain constructive and polite

What is a program?

- Generally a **program** is a **set of instructions** the programmer defines for a device or entity (usually a computer but not always) to follow

- Regarding computers-> programmer writes a set of instructions (“program”) that tells the computer to perform a set of operations

- When the program is executed, the instructions are carried out

- Does a program have to run on a digital machine? What is a computer? “Multiple realizability”

Why write a program, what does it have to do with neuroscience?

- What do you think? Course discussion…
- Many reasons you may want to write a program
- This can be anything, i.e.:
  - Processing data - behavioral, neural, environmental, etc.
  - Making a robot walk
  - Computer/phone/tablet app for some function

Why python?

- It’s free
- Tremendous library support
- Easy interpreted language, quick for prototyping
- Highly optimized computational libraries
- Cross platform/portability
- Strong user community for answering questions/knowledgebase

When python?

- Web app development
- Data science
- Scripting
- Database programming
- Quick prototyping

Why Jupyter Notebooks

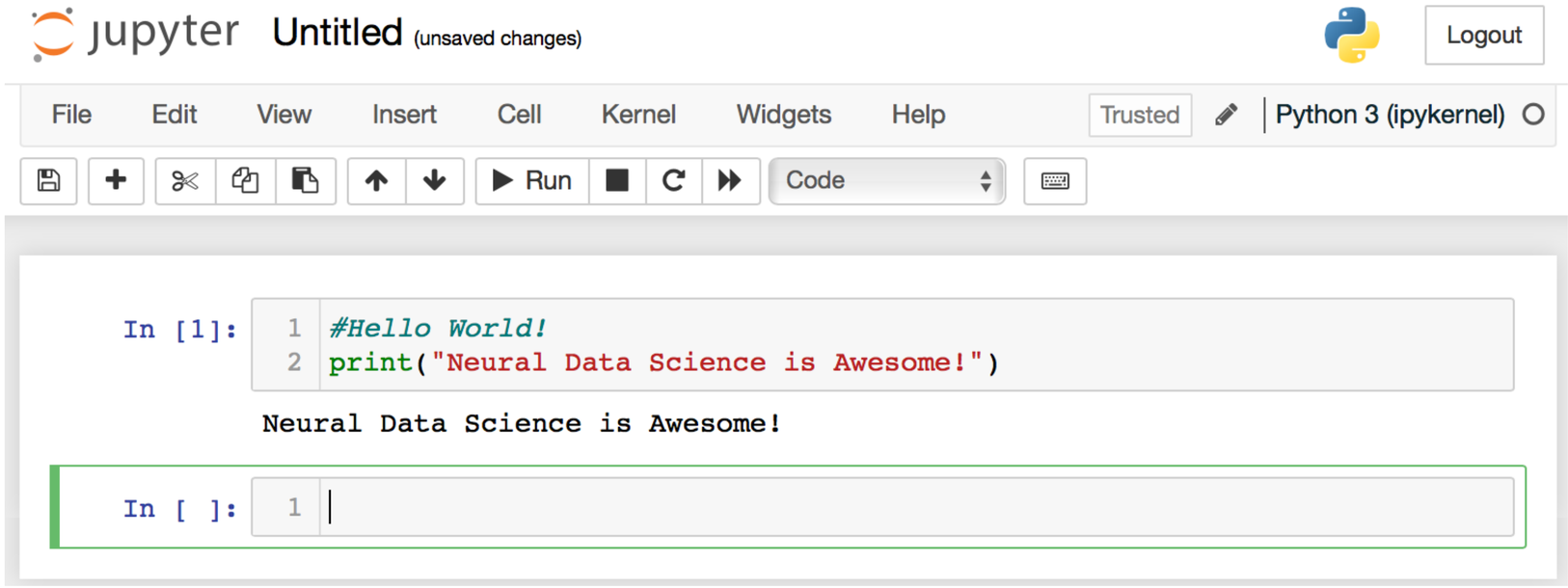
- Mixed media is excellent for data exploration and communication
- Don’t have to write a separate program from your notes, results, etc
- Easy to experiment in nonlinear and compartmentalized ways
- We’ll discuss the downsides later, but it’s not for all cases
  - It can be slow,
  - Version control can be difficult
  - Sometimes debugging is easier other times more difficult

JN use cases

- Prototyping
- Data ingestion
- Exploratory data analysis
- Feature engineering
- Model comparison
- Final model

Jupyter notebooks review

- https://jupyter.org/
- Installing anaconda
- https://github.com/COGS108/Tutorials
- https://github.com/NeuralDataScience/Tutorials
- Correcting common issues
- Up to students to correct and resubmit so grading can be timely



How do you write a program in Jupyer notebooks and python?

- datahub.ucsd.edu
- or your machine with anaconda
- The notebooks we will review are listed below and available in the lectures directory of the github and linked from the website and will be on canvas as well
  - 00-Introduction.ipynb
  - 01-Python.ipynb
  - 02-JupyterNotebooks.ipynb
  - 01_01_python-checkpoint.ipynb

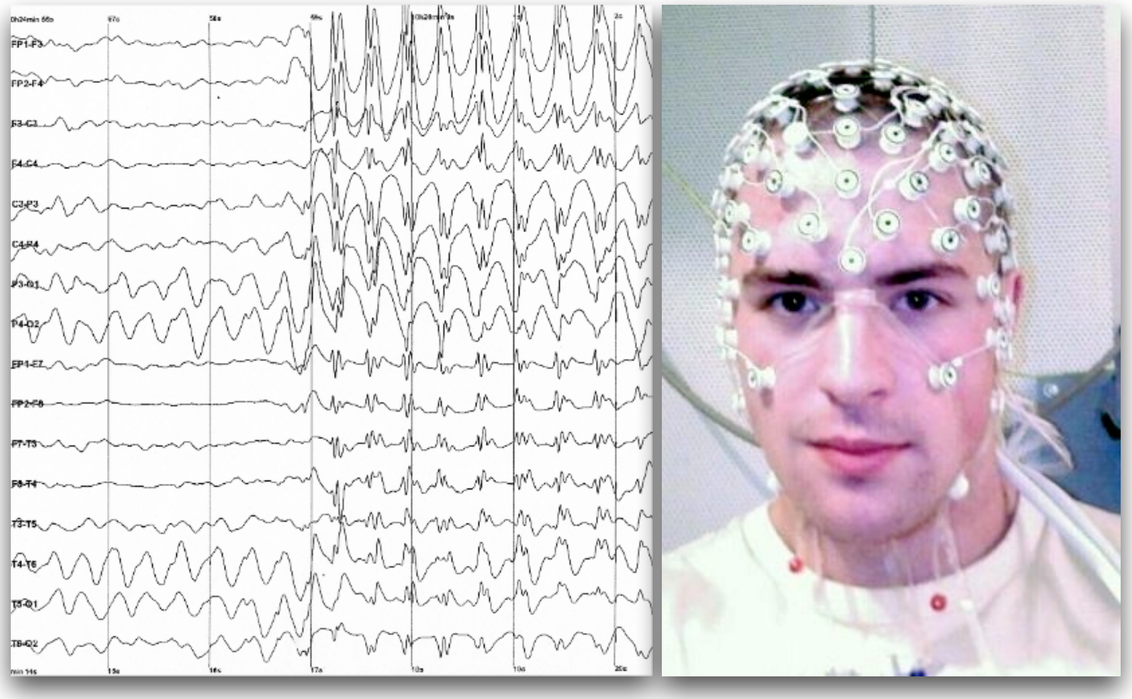
On to today…

Neural data science toolsets

- There are a variety of toolsets employed in neural data science
- Assist in processing data types, e.g.
  - **EEG and MEG analysis**
    - MNE -https://mne.tools/stable/index.html
  - **Linguistics**
    - NLTK- https://www.nltk.org
  - **Motion capture data** - kinematic/inverse kinematic and dynamic analysis

Why EEG and MEG analysis?

- **EEG** - **Electroencephalography**
  - Standard location patterns of sensors for recording (20-10, 10-10 systems)
  - EEG records electrical activity generated in your brain at the scalp
  - Global types of signals such as decision processes, spelling, gross body movement, etc
  - Why useful?



(Source: https://en.wikipedia.org/wiki/Electroencephalography)

Advantages of EEG

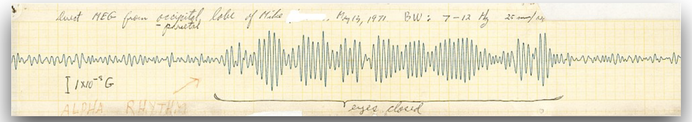
- Low cost
- Small, compact low complexity equipment vs. fMRI, PET, SPECT, MEG, MRS
- Simple data streams to process, signals can be used with minimal pre-processing
- Tolerant of subject movement, unlike many other methods
- Silent - fMRI anyone? Can have metal in nearby space
- No claustrophobia-inducing spaces
- No high intensity magnetic fields, no swallowing radioactive chemicals (PET)
- Better understanding of what signal is being measured than other technologies like fMRI (BOLD)

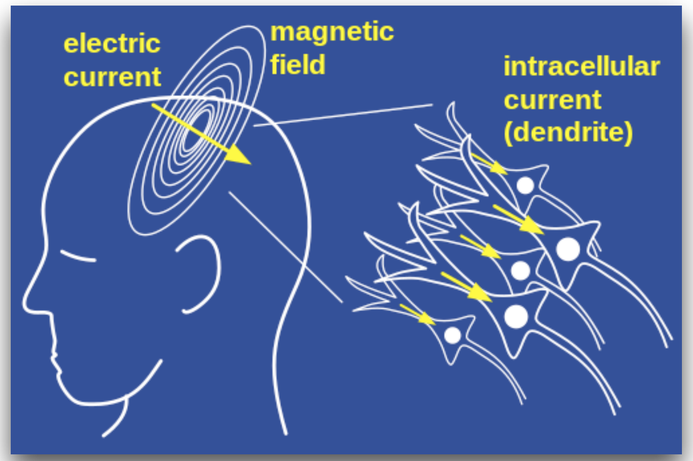
Disadvantages of EEG

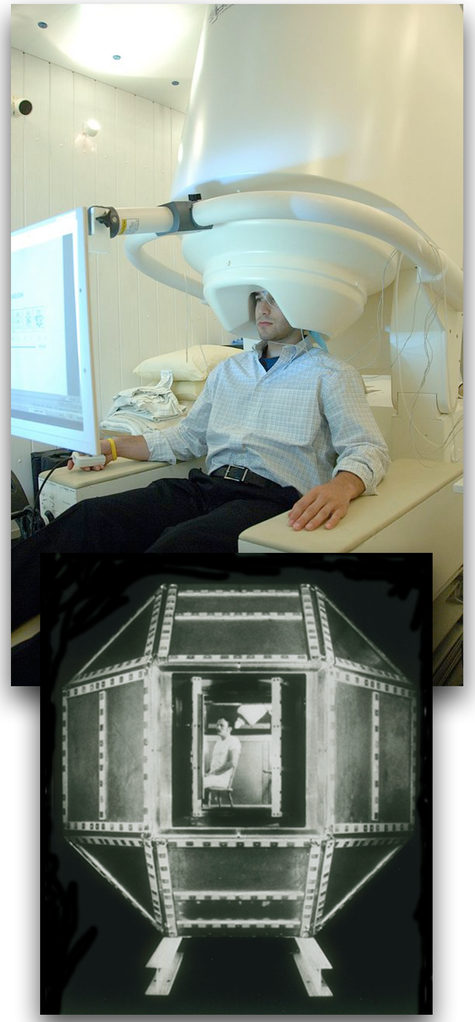
- Low spatial resolution and computational processing required to infer 3d spatial regions that are activated, must make assumptions about internal structure (can be based on scans)
- False localization is possible
- Cannot identify location in brain specific neurotransmitters found
- Slow to connect subjects vs fMRI, MEG, MRS, SPECT
- SNR poor/artifacts (eye movements, heart activity, blinks, facial movements, environment such as 60Hz noise, “Internal noise”)

MEG - Magnetoencephalography

- **MEG** - measurement of the magnetic field generated by electrical activity of neurons
- Mapped onto structural image from MRI
- **Advantages**
  - Provides a higher spatial (mm)/temporal (msec) resolution, no distortion through head
  - Decay relative to dist. is more pronounced than electrical fields thus useful for measuring superficial cortical activity
  - Shows absolute neuronal activity vs. fMRI shows relative activity (fMRI must always be compared to some reference neural activity)
  - Can be recorded for sleeping subjects, unconscious subjects other
  - Safe, no exposure to radiation/emf, noninvasive, easy to use



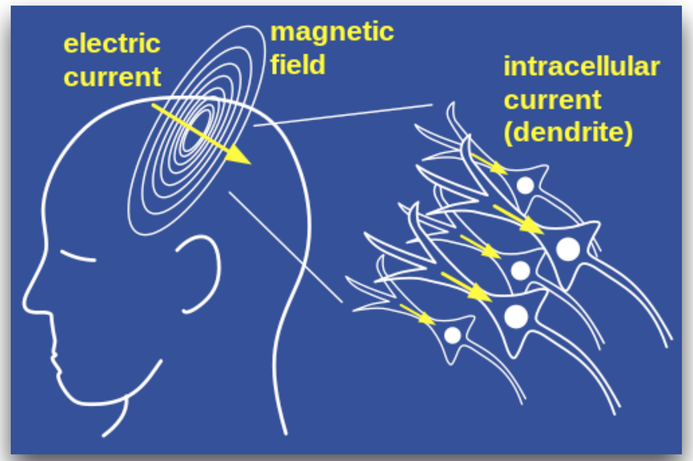


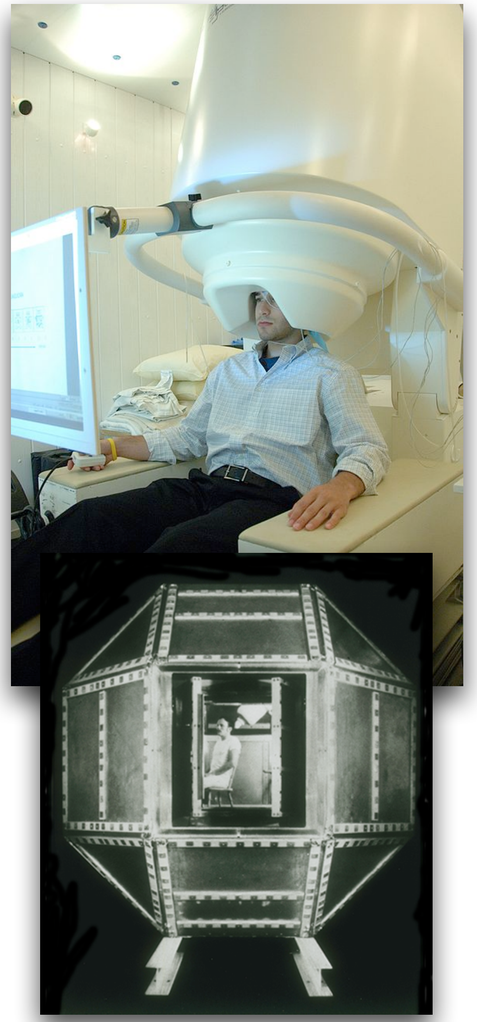


(Source: https://en.wikipedia.org/wiki/Magnetoencephalography)

MEG - Magnetoencephalography

- Disadvantages
  - Patients need to be fairly still
  - Pacemaker or VNS may not allow those patients
  - Possibly non-unique solution to localization problem
  - Sensitive instrumentation needed, subject to environmental noise



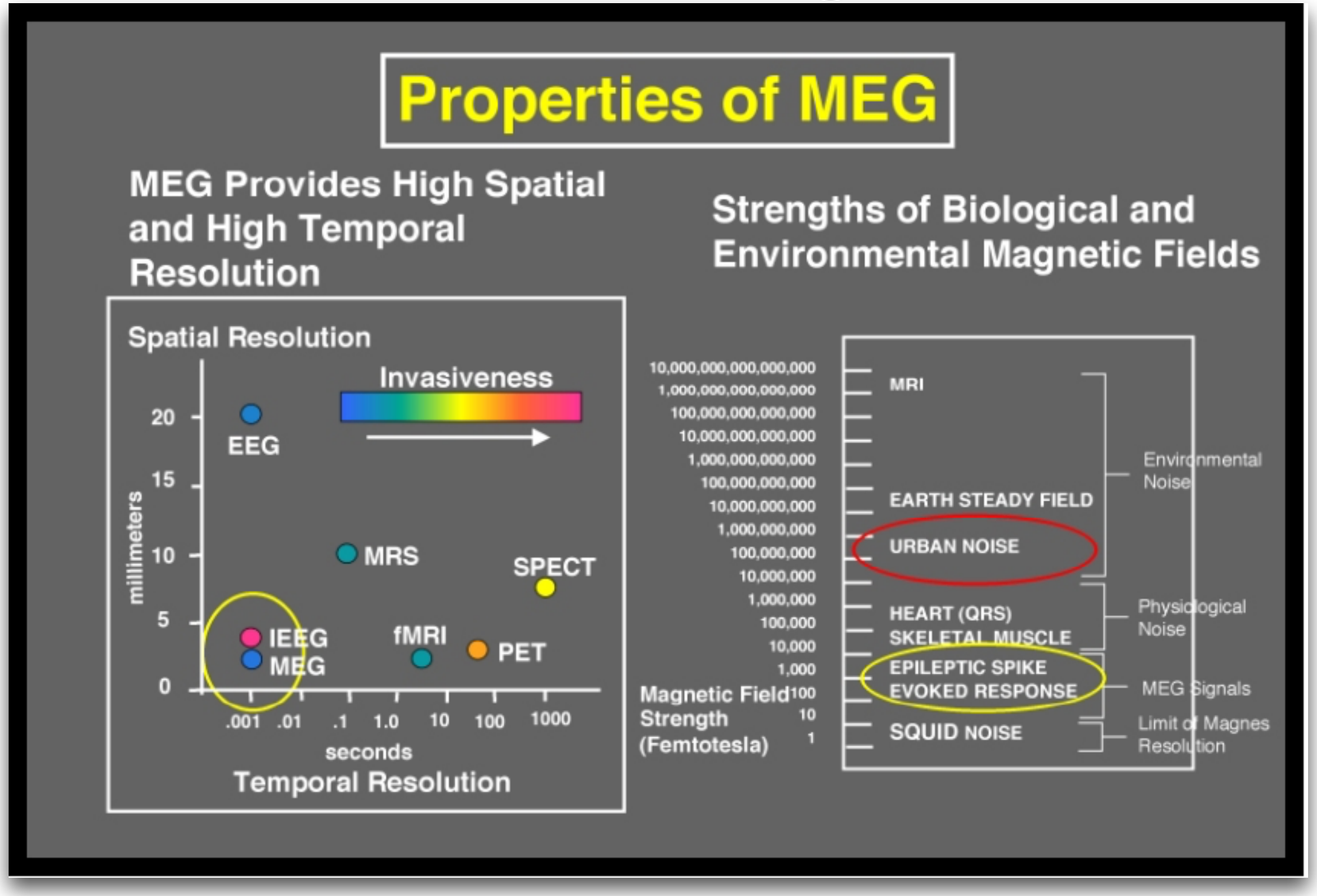


(Source: https://en.wikipedia.org/wiki/Magnetoencephalography)

Properties and challenges

Problem of biomagnetism:

- The brain's magnetic field, measuring at 10 femtotesla (fT) for cortical activity and 103 fT for the human alpha rhythm
- Ambient magnetic noise in an urban environment, which is on the order of 108 fT or 0.1 μT
- 50k Neurons for measurement
- Signals must be aligned->pyramidal cells (perp. to cortical surface)



(Source: http://web.mit.edu/kitmitmeg/MEG_Work_5.jpg)

Introduction to MNE

- https://mne.tools/stable/index.html
- https://mne.tools/stable/auto_tutorials/index.html

Natural Language Processing

- **NLTK** - natural language toolkit (python)
  - https://www.nltk.org/
- Easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet
- Libraries for easily performing - classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries
- Documentation and discussion forums

NLTK Simple Examples

- Adding to your notebook or script:

```
import nltk
```

- Checking installation:

```
import nltk
nltk.__path__
```

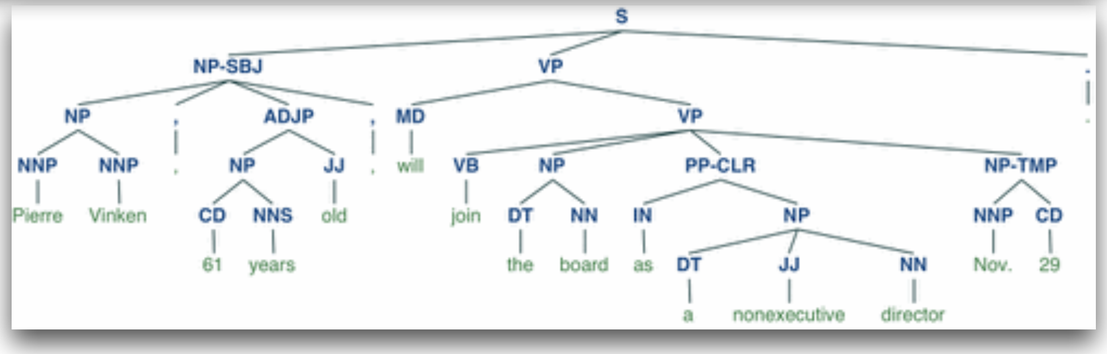
- Tokenize and tag text:

```
nltk.word_tokenize()
nltk.pos_tag()
```

- Display a parse tree:

```
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0001.mrg')[0]
t.draw()
```

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

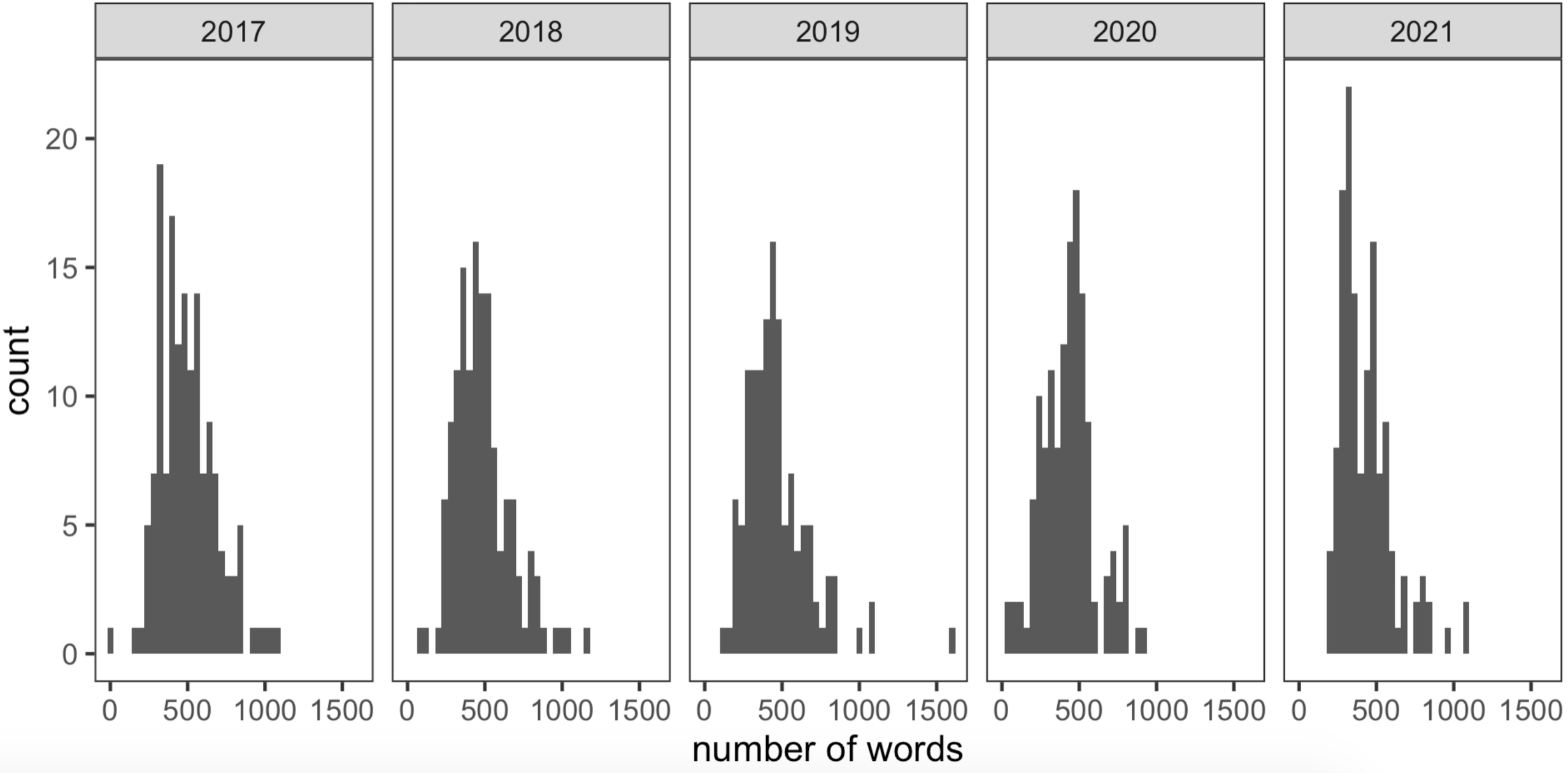


NLTK functions

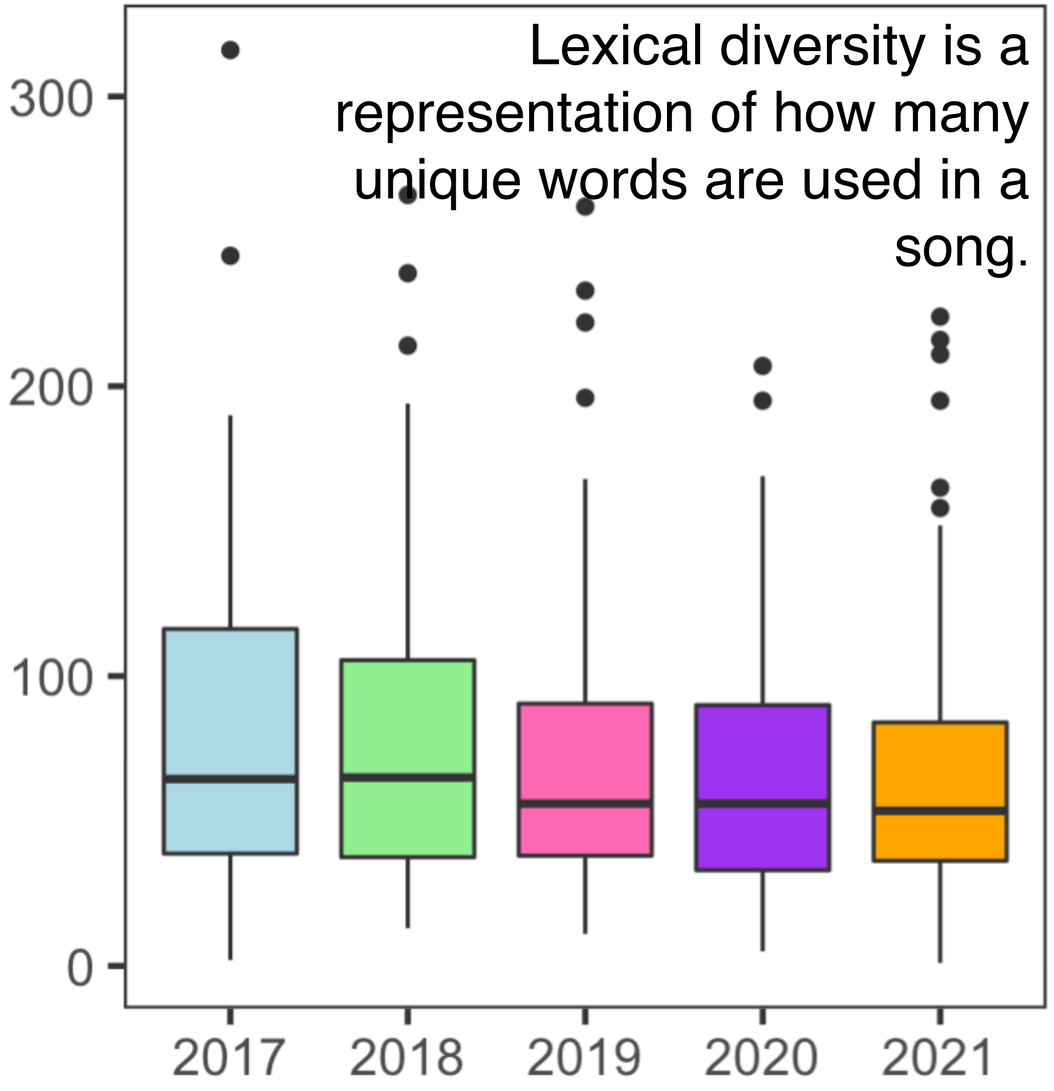
- To page…https://www.nltk.org/

Questions we can ask...

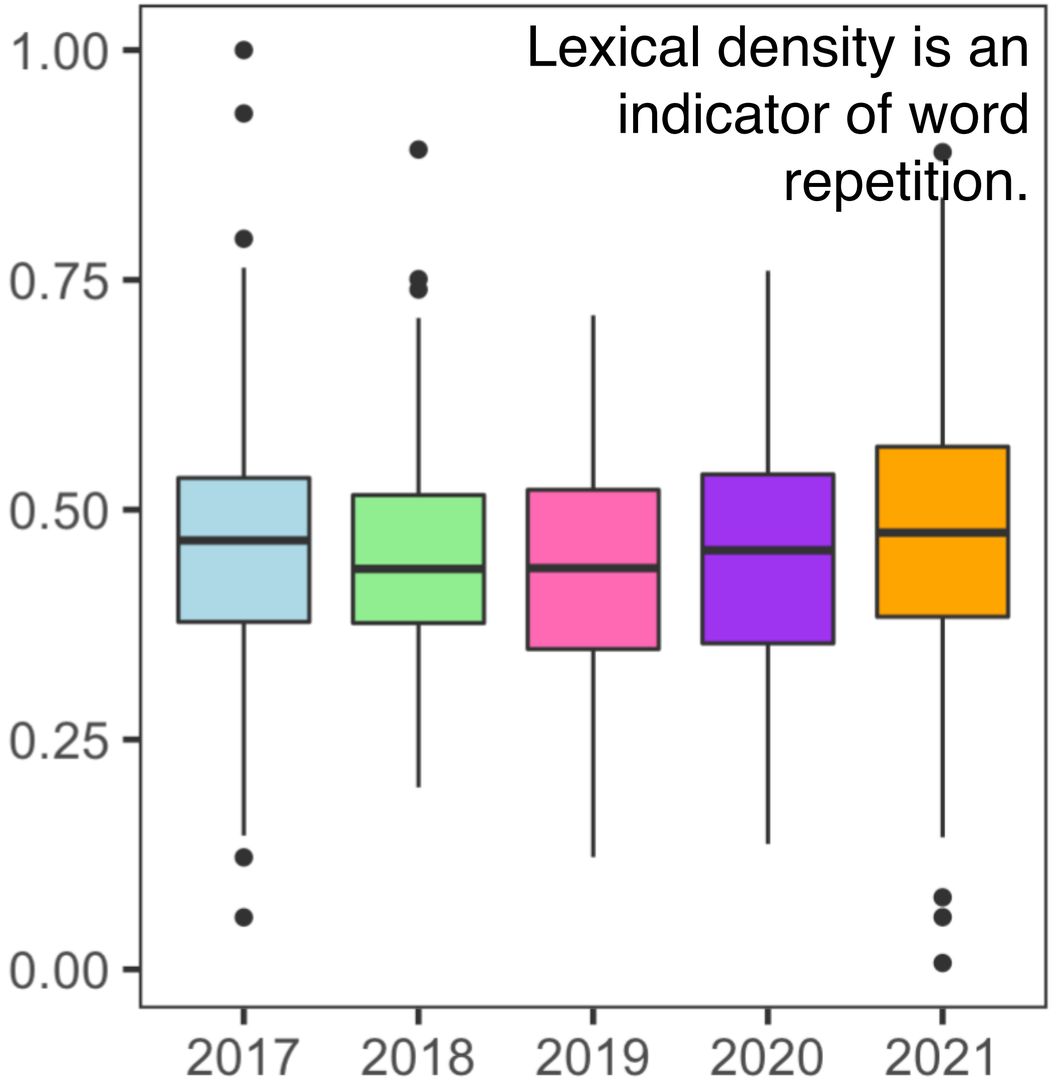
1. Does the total number of words change over time?
2. Does uniqueness change over time?
3. Does the diversity or density change?
4. What words are most common?
5. What words are most unique to each year?
6. What sentiment do songs convey most frequently?
7. Has sentiment changed over time?
8. What are 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



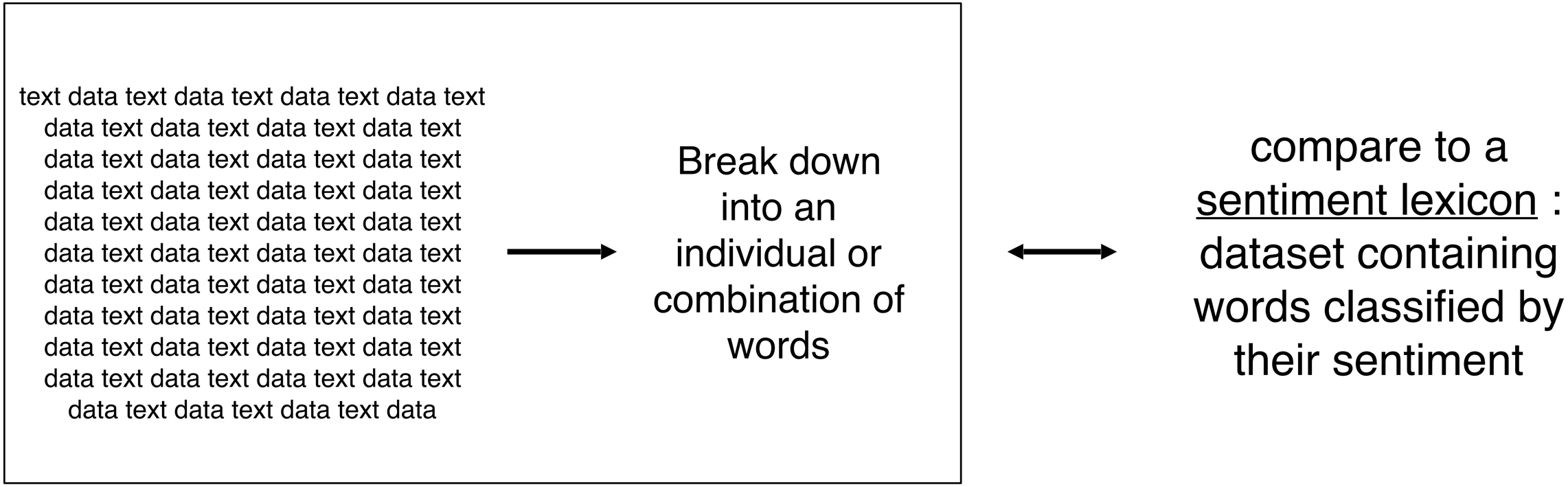
Lexical Density



Sentiment Analysis

Sentiment Analysis

Programmatically infer emotional content of text



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.)

stop words - words not helpful for analysis

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

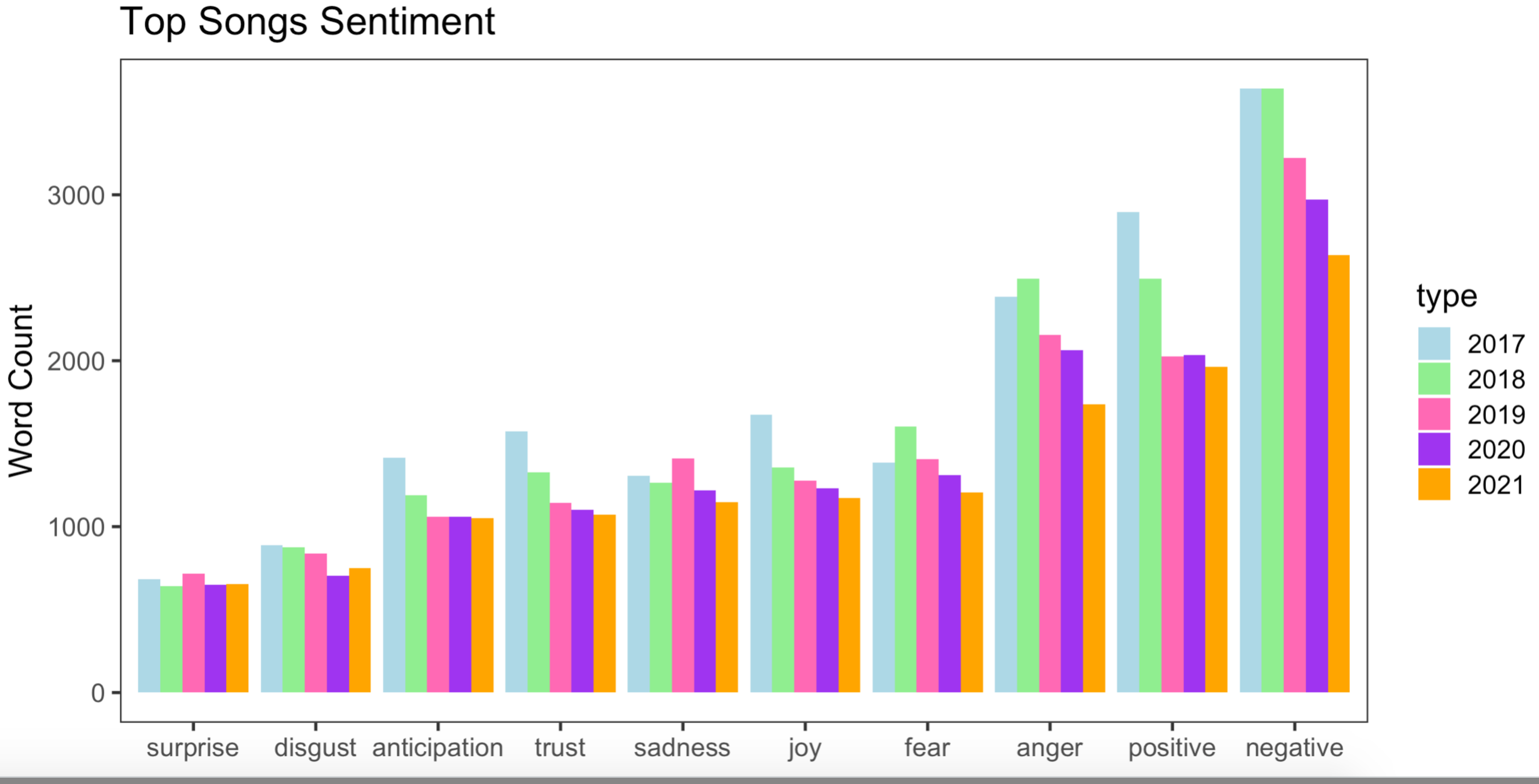
When doing sentiment analysis...

stemming - lexicon normalization

- Identifying the root for each token
- Jumping, jumped, jumps, jump all have the same root ‘jump’
- Where things get tricky: jumper???

In text analysis, your choices matter:

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


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

TF-IDF

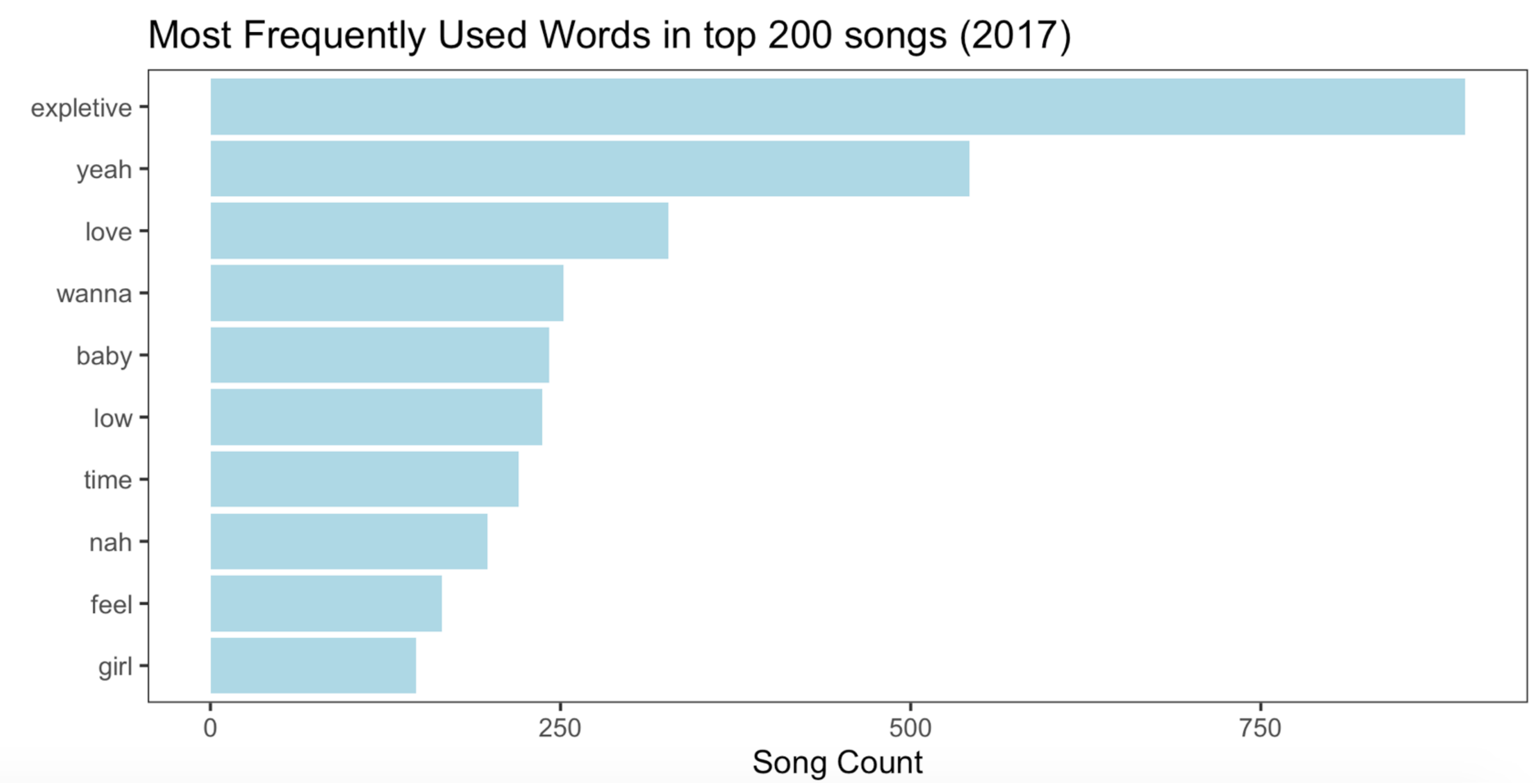
Term Frequency - Inverse Document Frequency

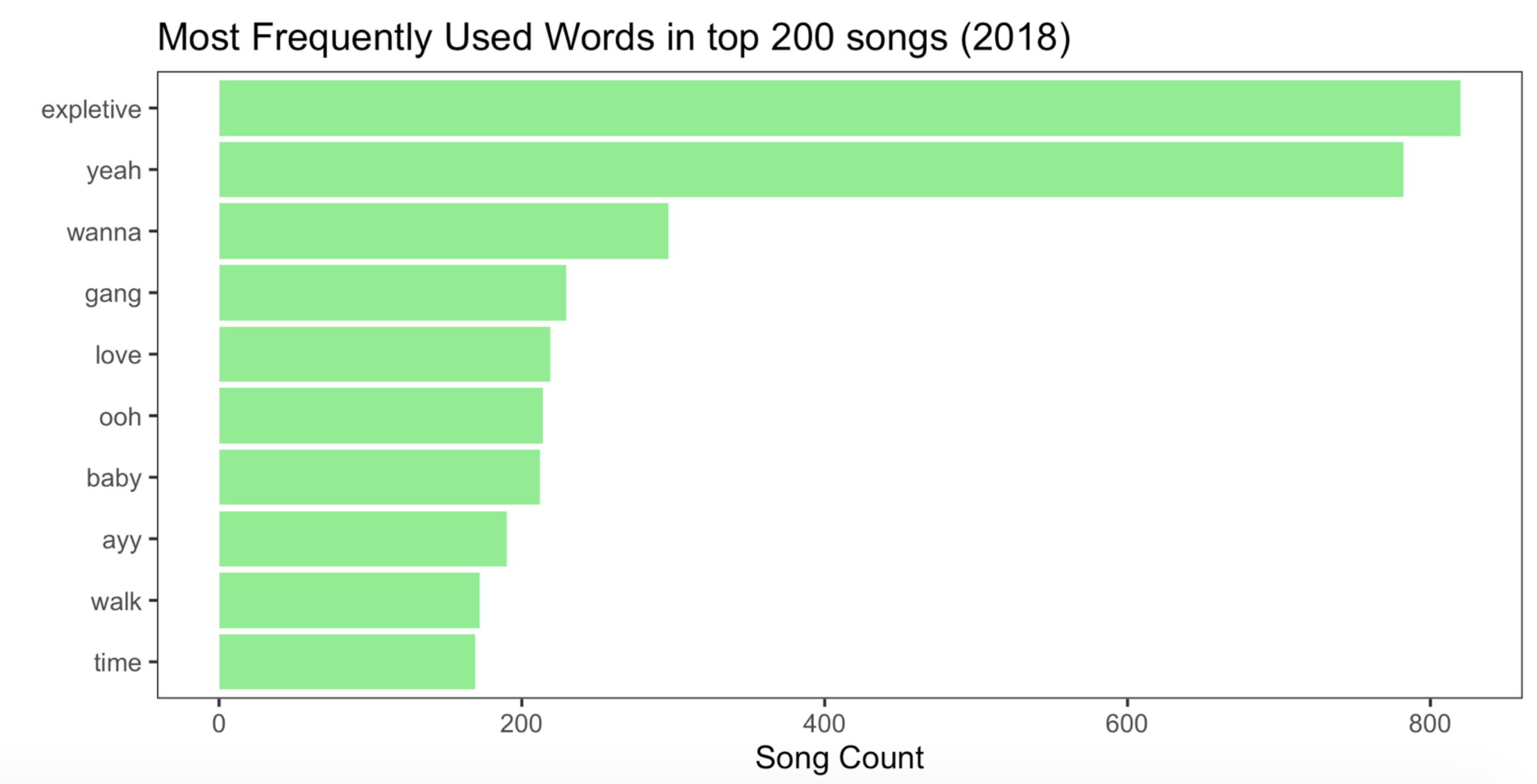
What words are the most unique to the lyrics of each year's top hits?

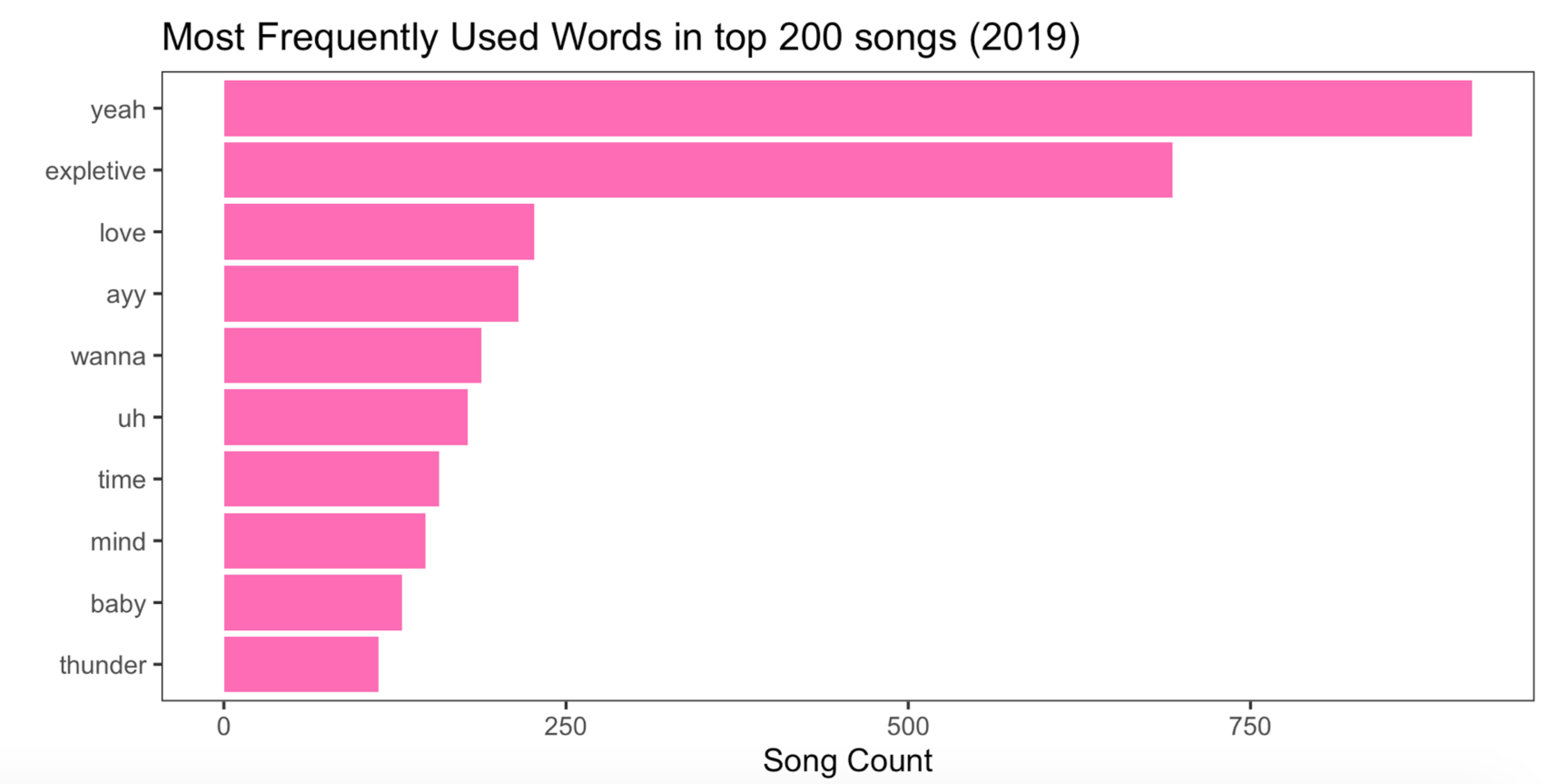
- Goal: to use TF-IDF to *find the important words* for the content of each document by decreasing the weight for commonly used words and increasing the weight for words that are not used very much in a collection or corpus of documents

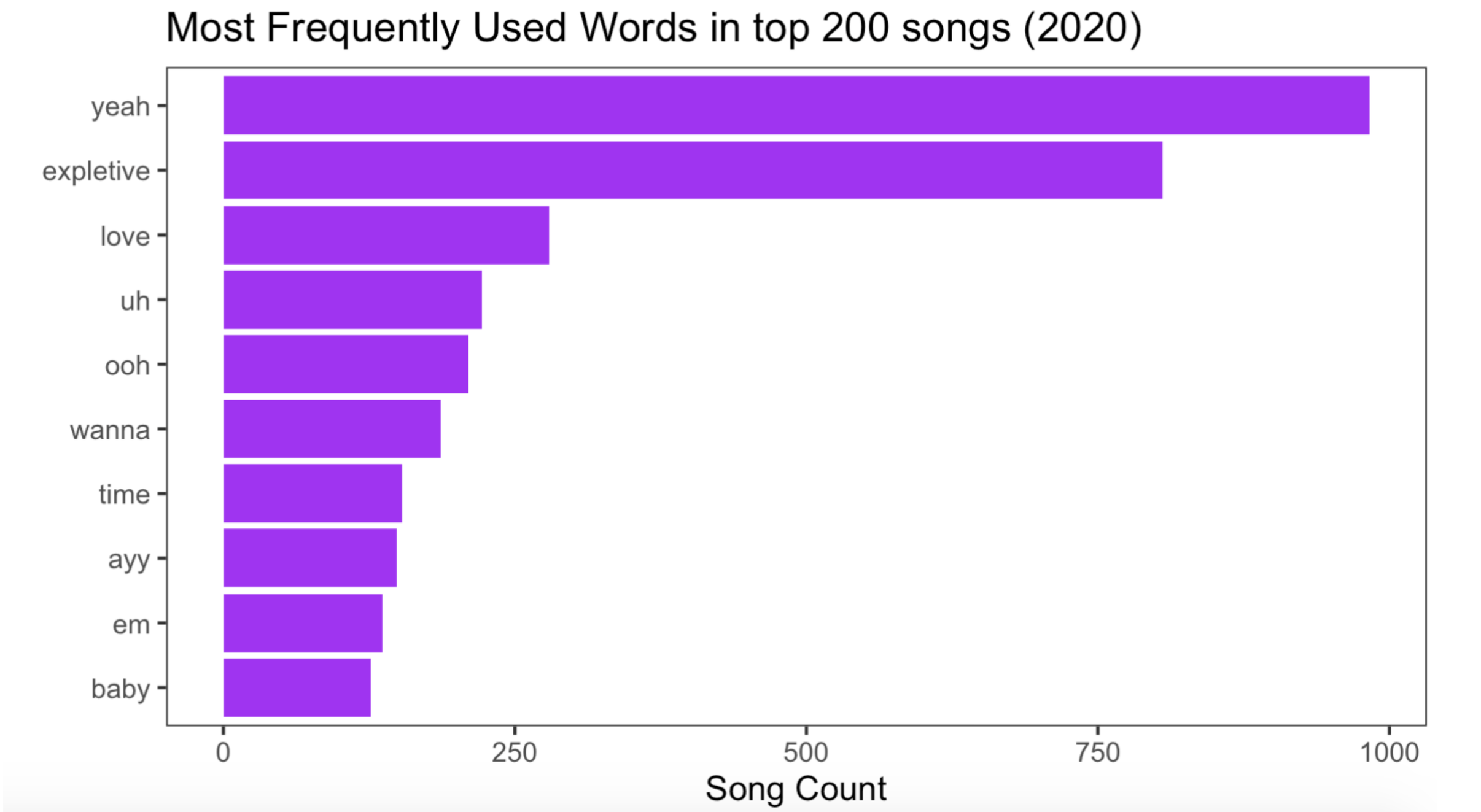
- Calculating TF-IDF attempts to find the words that are important (i.e., common) in a text, but not *too* common

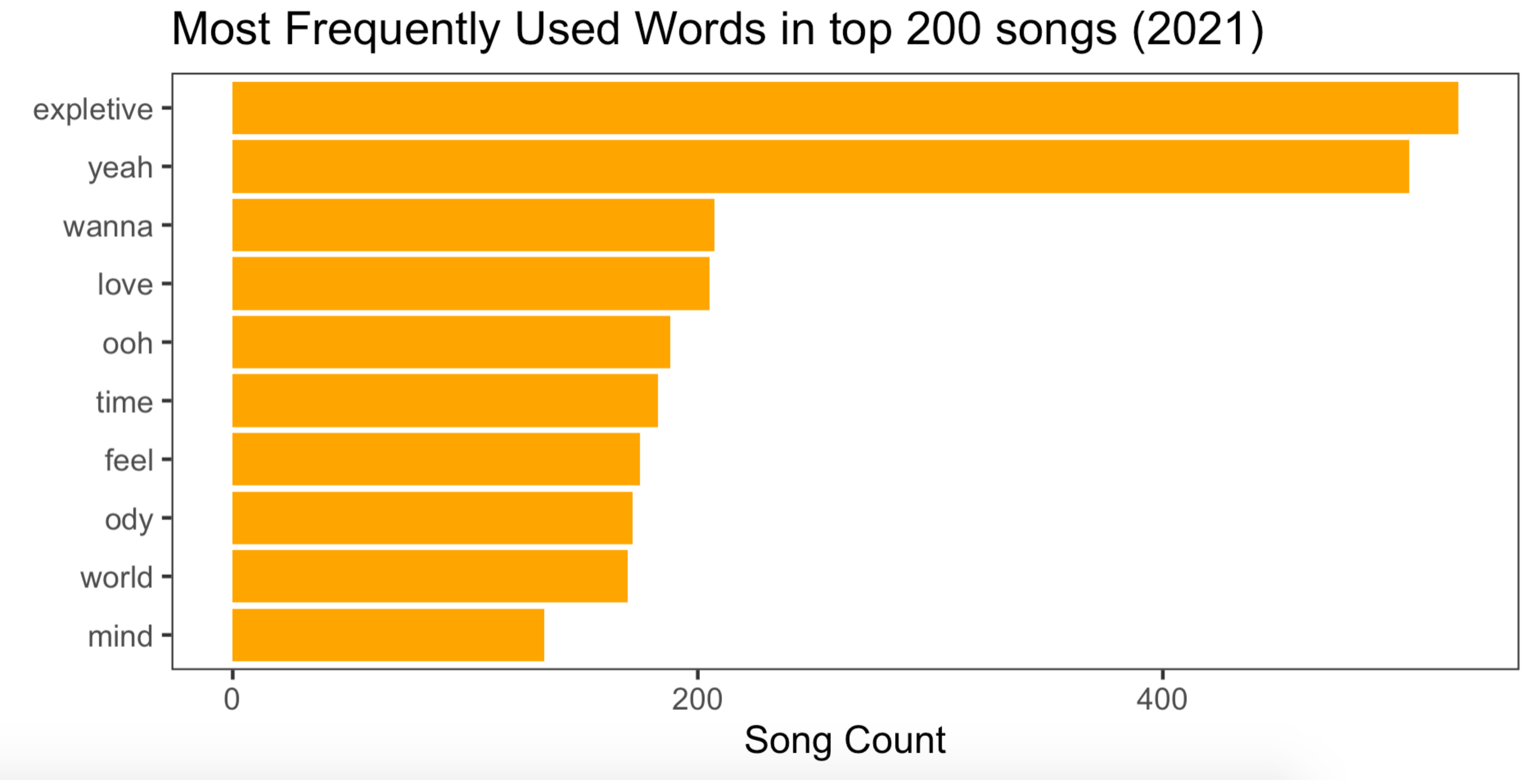
https://www.tidytextmining.com


Most Frequently Used Words in top 200 songs (2017)
expletive
yeah
love
wanna
baby
low
time
nah
feel
girl
0
250
500
750
Song Count

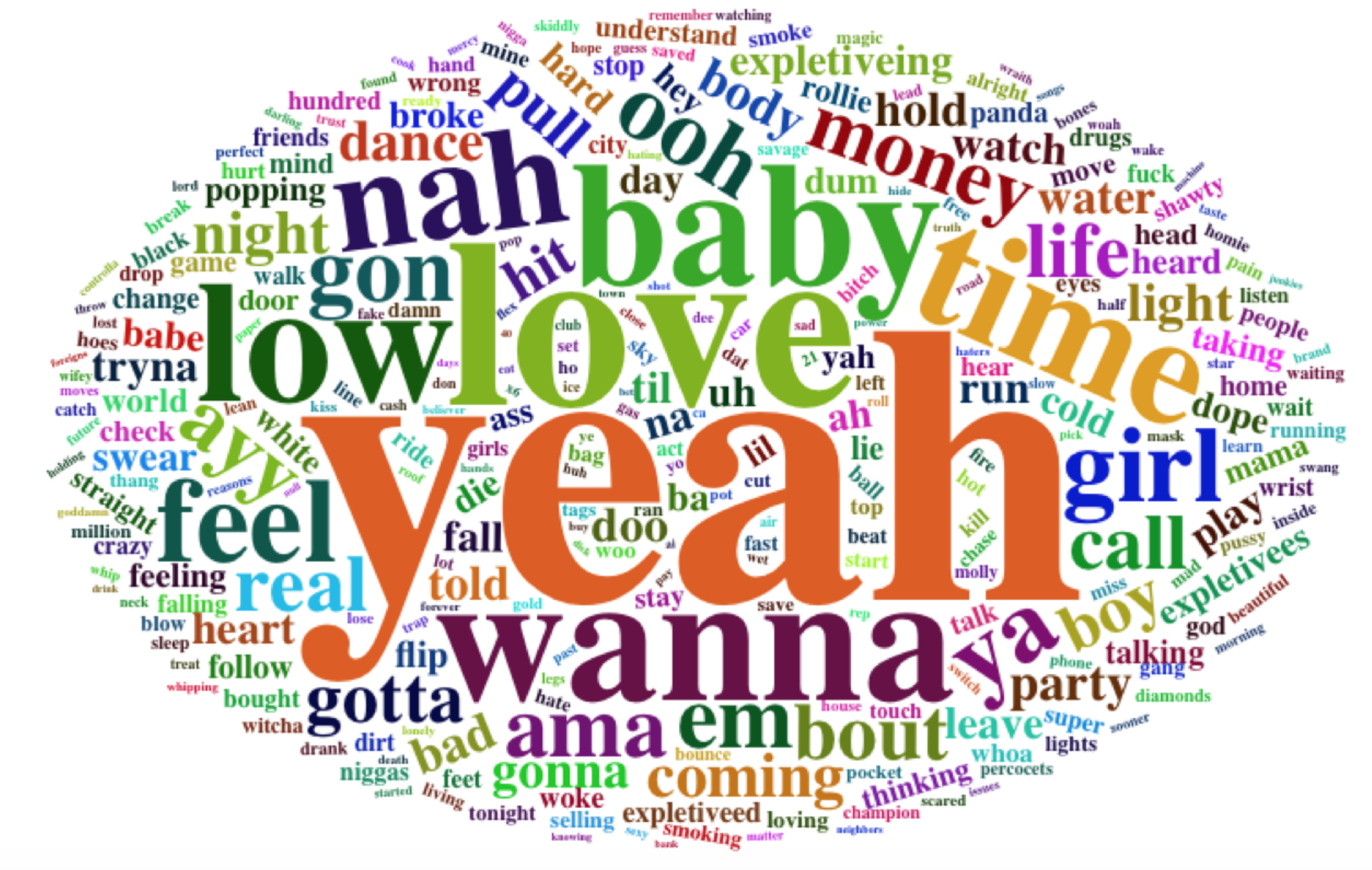

Most Frequently Used Words in top 200 songs (2018)
expletive
yeah
wanna
gang
love
ooh
baby
ayy
walk
time
0
200
400
600
800
Song Count


Most Frequently Used Words in top 200 songs (2019)
yeah
expletive
love
ayy
wanna
uh
time
mind
baby
thunder
0
250
500
750
Song Count


Most Frequently Used Words in top 200 songs (2020)
yeah
expletive
love
uh
ooh
wanna
time
ayy
em
baby
0
250
500
750
1000
Song Count

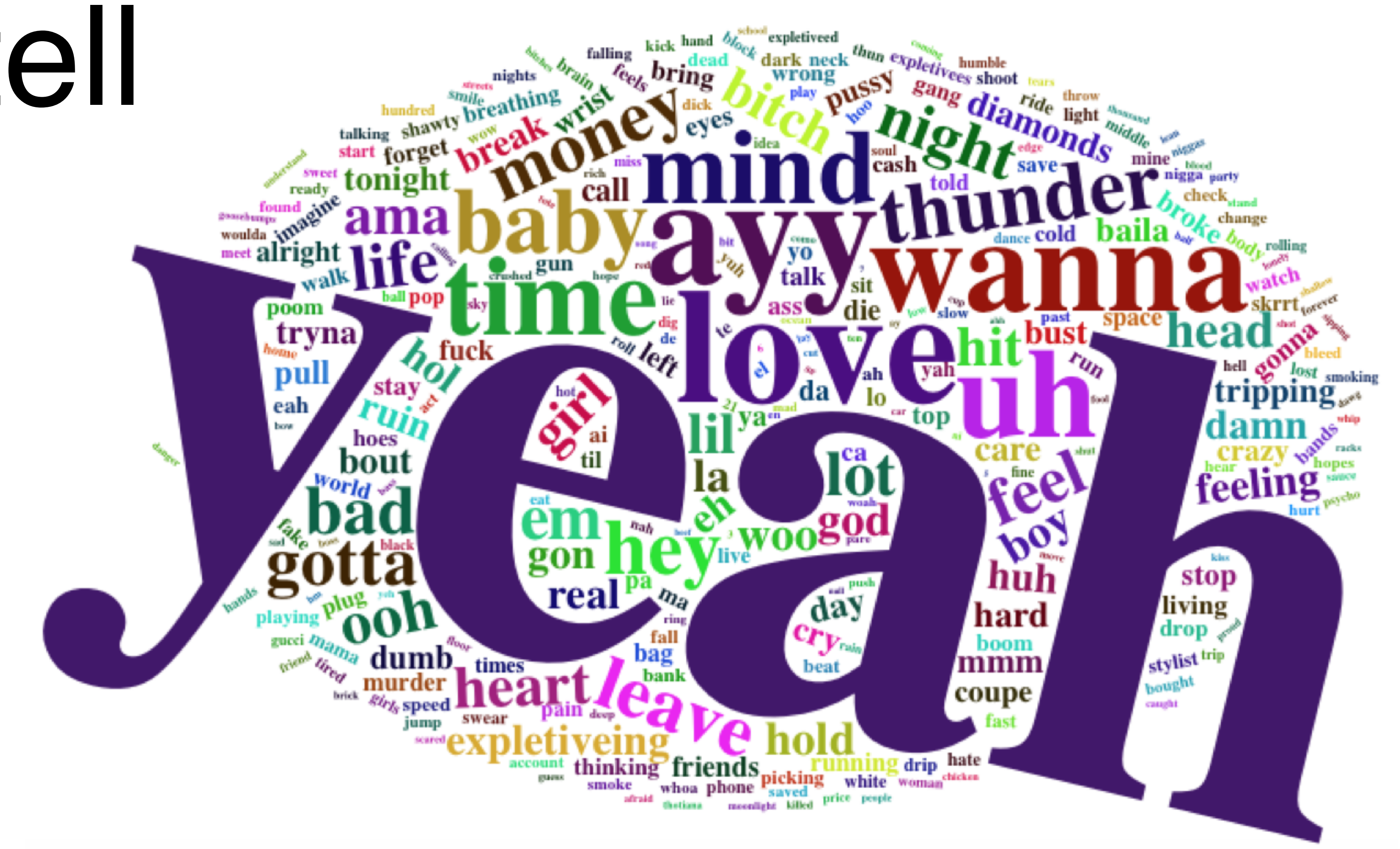

Most Frequently Used Words in top 200 songs (2021)
expletive
yeah
wanna
love
ooh
time
feel
ody
world
mind
0
200
400
Song Count

Term Frequency can only tell us so much….



2017

2018



2019

2020

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

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

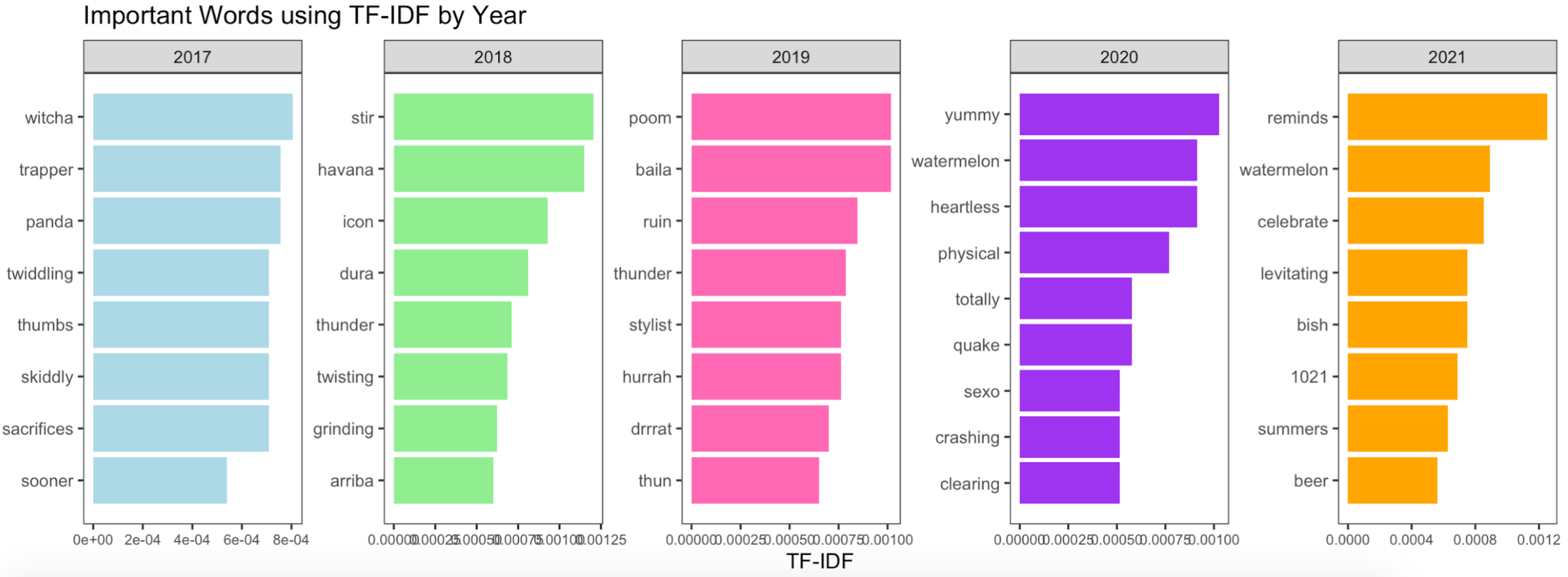
TF-IDF

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


Important Words using TF-IDF by Year
2017
witcha
trapper
panda
twiddling
thumbs
skiddly
sacrifices
sooner
0e+00
2e-04
4e-04
6e-04
8e-04
2018
stir
havana
icon
dura
thunder
twisting
grinding
arriba
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
2019
poom
baila
ruin
thunder
stylist
hurrah
drrrat
thun
0.00000
0.00025
0.00050
0.00075
0.00100
2020
yummy
watermelon
heartless
physical
totally
quake
sexo
crashing
clearing
0.00000
0.00025
0.00050
0.00075
0.00100
2021
reminds
watermelon
celebrate
levitating
bish
1021
summers
beer
0.0000
0.0004
0.0008
0.0012
TF-IDF

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?

EDA

4. What words are most common?
5. What words are most unique to each year?

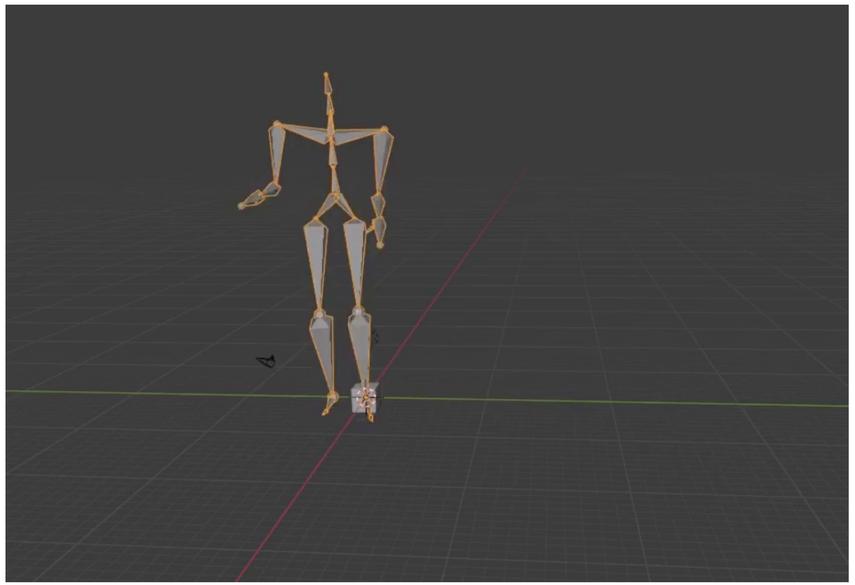
TF-IDF

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?

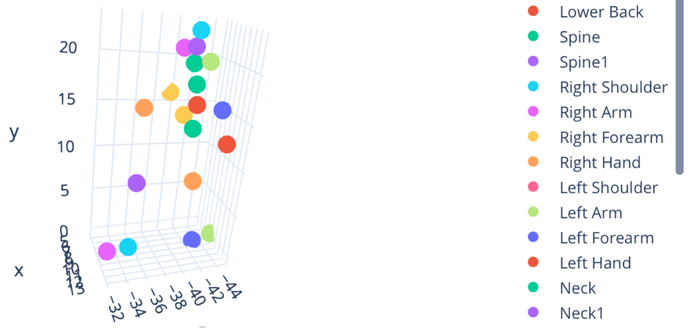
Sentiment Analysis

Motion capture data

- Recorded via
  - MoCap cameras - excellent, multiple types
  - Video - ok
  - IMUs - ok
- PyMO - https://omid.al/projects/pymo/
- pypi - https://pypi.org/project/mocaplib/
- Others
- Not well standardized yet

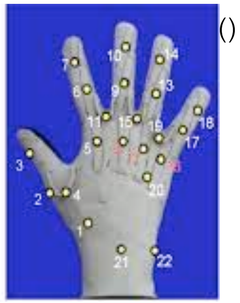


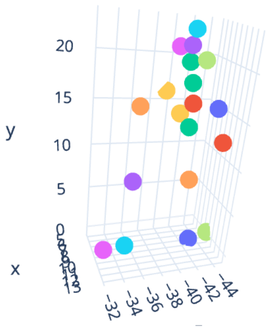
(Source: https://medium.com/swlh/movement-classification-b98614084ec6)



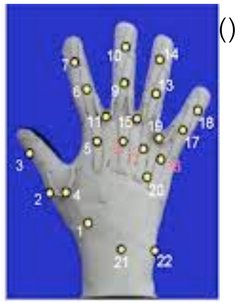
Motion capture data - challenges

- Hand manipulation involves many occlusions
  - Estimation
  - High camera density
  - Active markers
- Predictive estimation
- Marker occlusions generally, jumps and discontinuities
- Active systems require power, wires, may be delicate
- https://www.engadget.com/2018-05-25-motion-capture-history-video-vicon-siren.html





Motion capture systems

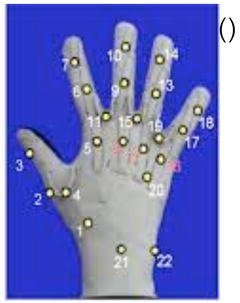


()

- Two main approaches
  - Active systems
  - Passive systems
- Combinations

Motion capture systems

- **VICON**: https://www.youtube.com/watch?v=HBD6vA0Xi6Y



()

- **PhaseSpace**:
  - https://www.youtube.com/watch?v=A1BrYmC1Vpo
  - https://www.youtube.com/watch?v=iklXUxpq-T4