# Day 9: Text Analysis

## ME314: Introduction to Data Science and Machine Learning

Jack Blumenau

25th July 2022

# Day 10 Outline

# Key Features of QTA

# Basic QTA workflow: Texts → Feature matrix → Analysis

An economic miracle is taking place in the United States, and the only thing that can stop it are foolish wars, politics, or ridiculous partisan investigations.

The United States of America right now has the strongest, most durable economy in the world. We're in the middle of the longest streak of private sector job creation in history.

We reinvented Government, transforming it into a catalyst for new ideas that stress both opportunity and responsibility and give our people the tools they need to solve their own problems.

To build a prosperc future, we must trus people with their own money and empower them to grow our economy.

*Source texts*

*Processed text as a document-feature matrix*

| documents | economy | united | wall | crime | climate |
|---|---|---|---|---|---|
| | features | | | | |
| Clinton-2000 | 10 | 4 | 1 | 5 | 1 |
| Bush-2008 | 6 | 4 | 0 | 0 | 1 |
| Obama-2016 | 16 | 4 | 1 | 0 | 4 |
| Trump-2019 | 5 | 19 | 6 | 2 | 0 |

*Quantitative analysis and inference*

Describing texts quantitatively or stylistically
Identifying keywords
Measuring ideology or sentiment in documents
Mapping semantic networks
Identifying topics and estimating their prevalence
Measuring document or term similarities
Classifying documents

# What role for “qualitative’ ’ analysis in QTA?

- Ultimately all reading of texts is qualitative, even when we count elements of the text or convert them into numbers
- QTA may involve human judgment in the construction of the feature-document matrix
- QTA may involve human judgment in the interpretation of the output of statistical models
- But QTA differs from more qualitiative approaches in that it:
  - Involves large-scale analysis of many texts, rather than close readings of few texts
  - Requires no *interpretation* of texts
- Uses a variety of statistical techniques to extract information from the document-feature matrix

# Key feature of quantitative text analysis

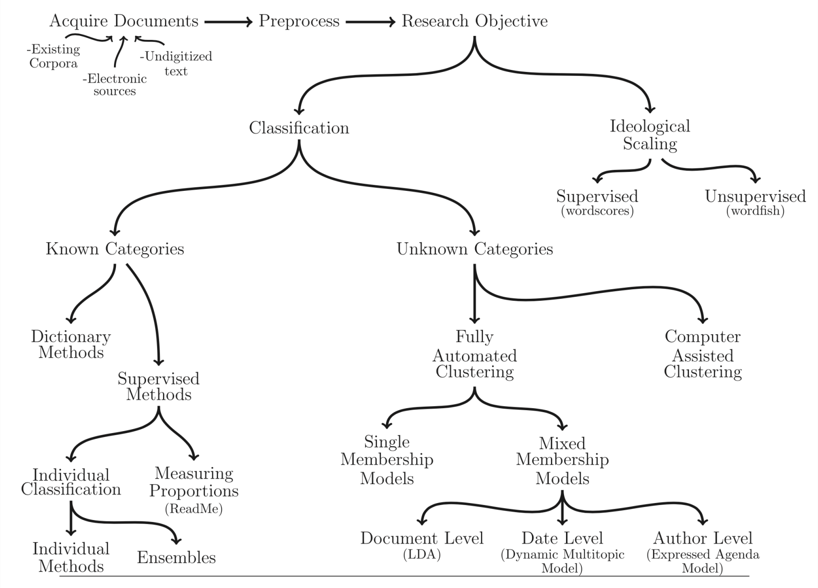
- Conversion of textual features into a quantitative matrix
- A quantitative or statistical procedure to extract information from the quantitative matrix
- Summary and interpretation of the quantitative results

# Key goals of quantitative text analysis

1. Prediction for 'downstream' tasks
    - Can we predict consumer behaviour from product reviews?
    - Can we predict football match outcomes using tweets?
2. Understanding of language use
    - Do men and women discuss political concepts differently?
    - How has the meaning of words changed over time?
3. Measurement of latent constructs
    - Can we infer student sophistication from the *complexity* of their writing?
    - Which set of *topics* characterises a corpus of texts?

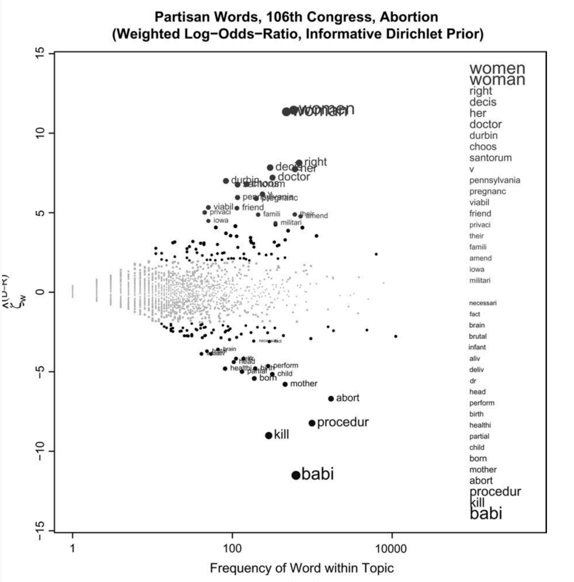
# 3 guiding priciples for QTA

1. All quantitative models for text are wrong, but some are useful
2. Quatitative models for text augment, but do not replace, humans
3. Validation is key


Acquire Documents
Preprocess
Research Objective
-Existing Corpora
-Electronic sources
-Undigitized text
Classification
Ideological Scaling
Supervised (wordscores)
Unsupervised (wordfish)
Known Categories
Unknown Categories
Dictionary Methods
Supervised Methods
Fully Automated Clustering
Computer Assisted Clustering
Individual Classification
Measuring Proportions (ReadMe)
Single Membership Models
Mixed Membership Models
Individual Methods
Ensembles
Document Level (LDA)
Date Level (Dynamic Multitopic Model)
Author Level (Expressed Agenda Model)

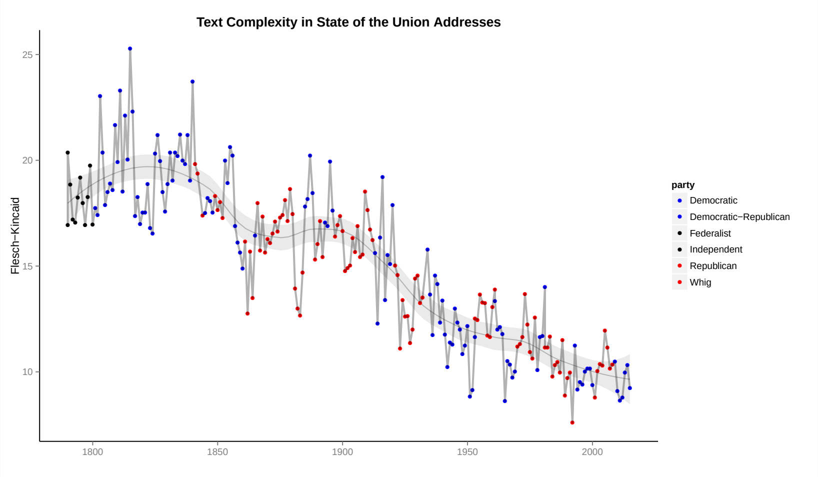
# Example: Wordclouds


(from Herzog and Benoit EPSA 2013)

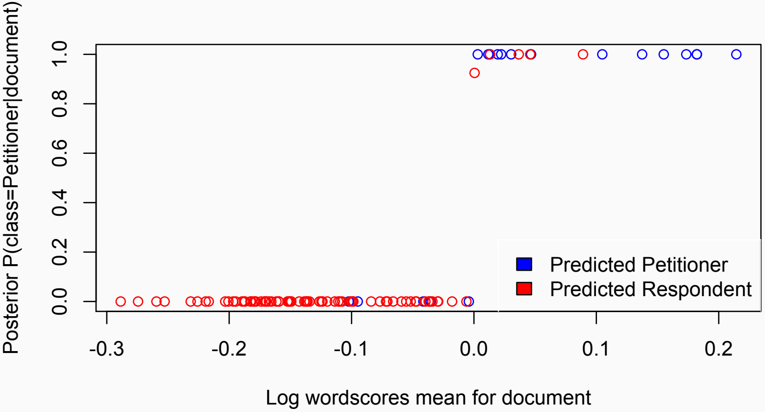



(from Munroe et al., 2009)

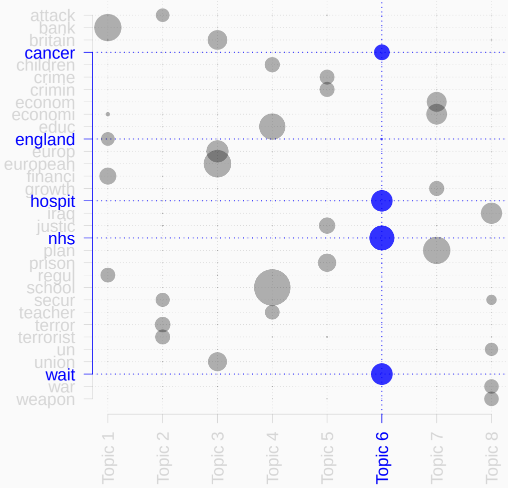
# Example: Text complexity



# Example: Document classification



# Example: Exploring the topics of a group of texts



# This requires assumptions

- That texts represent an observable implication of some underlying characteristic of interest (usually an attribute of the author)
- That texts can be represented through extracting their *features*
    - most common is the bag of words assumption
    - disregard grammar, disregard word order, just pay attention to word frequencies
    - many other possible definitions of “features’’
- A document-feature matrix can be analyzed using quantitative methods to produce meaningful and valid estimates of the underlying characteristic of interest

# Bag of words assumption

- Consider two sentences:

  1. Time flies like an arrow.
  2. Fruit flies like a banana.

- Convert these into a bag-of-words feature matrix:

| | time | flies | fruit | like | an | a | banana | arrow |
|---|---|---|---|---|---|---|---|---|
| Sentence 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| Sentence 2 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

- The dependency structure between words in each sentence is lost
- The word "flies" has a different meaning in the two sentences (metaphorical versus literal), but both sentences score a 1 here
- The "joke" is no longer funny

# Key features of quantitative text analysis

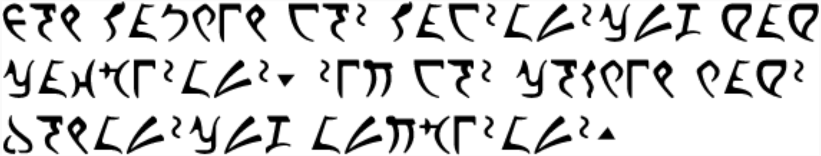
1. Selecting texts: Defining the *corpus*
2. Conversion of texts into a common electronic format
3. Defining documents: deciding what will be the unit of analysis (document, paragraph, sentence, etc)

# Key features of quantitative text analysis

4. Defining features. These can take a variety of forms, including tokens, equivalence classes of tokens (dictionaries), selected phrases, human-coded segments (of possibly variable length), linguistic features, and more.
5. Conversion of textual features into a quantitative matrix
6. A quantitative or statistical procedure to extract information from the quantitative matrix
7. Summary and interpretation of the quantitative results

# Extreme forms of QTA

- Fully automated technique with minimal human intervention or judgment calls – only with regard to reference text selection
- Methods can “discover” topics with little human supervision
- Language-blind: can scaling anything that occurs with regular patterns (even without knowing what these mean)
- Could potentially work on texts like this:



http://www.kli.org

# Some key basic concepts

**(text) corpus** a large and structured set of texts for analysis

**types** for our purposes, a unique word

**tokens** any word – so token count is total words

**stems** words with suffixes removed

**lemmas** canonical word form

***keys*** such as dictionary entries, where the user defines a set of equivalence classes that group different word types

# Some more key basic concepts

**"key" words** Words selected because of special attributes, meanings, or rates of occurrence

**stop words** Words that are designated for exclusion from any analysis of a text

**readability** provides estimates of the readability of a text based on word length, syllable length, etc.

**complexity** A word is considered "complex" if it contains three syllables or more

**diversity** (lexical diversity) A measure of how many types occur per fixed word rate (a normalized vocabulary measure)

# Documents and Features

# Strategies for selecting units of textual analysis

- Words
- $n$-word sequences
- pages
- paragraphs
- Natural units (a speech, a poem, a manifesto)
- Key: depends on the research design

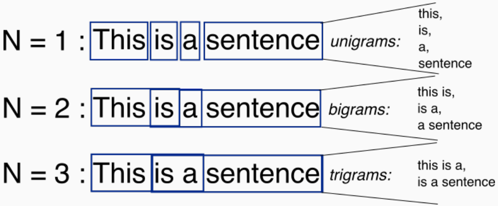
# Defining Features

- words
- word stems or lemmas: this is a form of defining *equivalence classes* for word features
- word segments, especially for languages using compound words, such as German, e.g.

  *Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz*

(the law concerning the delegation of duties for the supervision of cattle marking and the labelling of beef)

- word sequences/n-grams: contiguous sequence of words from document (1-gram, unigram; 2-gram, bigram, etc)



# Defining Features

- (if qualitative coding is used) coded or annotated text segments
- linguistic features: parts of speech

# Parts of speech

- the Penn "Treebank" is the standard scheme for tagging POS

| Number | Tag | Description |
|---|---|---|
| 1. | CC | Coordinating conjunction |
| 2. | CD | Cardinal number |
| 3. | DT | Determiner |
| 4. | EX | Existential *there* |
| 5. | FW | Foreign word |
| 6. | IN | Preposition or subordinating conjunction |
| 7. | JJ | Adjective |
| 8. | JJR | Adjective, comparative |
| 9. | JJS | Adjective, superlative |
| 10. | LS | List item marker |
| 11. | MD | Modal |
| 12. | NN | Noun, singular or mass |
| 13. | NNS | Noun, plural |
| 14. | NNP | Proper noun, singular |
| 15. | NNPS | Proper noun, plural |
| 16. | PDT | Predeterminer |
| 17. | POS | Possessive ending |
| 18. | PRP | Personal pronoun |
| 19. | PRP$ | Possessive pronoun |
| 20. | RB | Adverb |

# POS tagging in R

```
library(spacyr)
text <- "Harry Potter is a boy wizard at
        Hogwarts School of Witchcraft and Wizardry."
spacy_parse(text)
```

```
##          token      lemma   pos   entity
## 1        Harry      Harry PROPN PERSON_B
## 2       Potter     Potter PROPN PERSON_I
## 3           is         be   AUX
## 4            a          a   DET
## 5          boy        boy  NOUN
## 6       wizard     wizard  VERB
## 7           at         at   ADP
## 8     Hogwarts   Hogwarts PROPN    ORG_B
## 9       School     School PROPN    ORG_I
## 10          of         of   ADP    ORG_I
## 11 Witchcraft Witchcraft PROPN    ORG_I
## 12         and        and CCONJ    ORG_I
## 13    Wizardry   Wizardry PROPN    ORG_B
## 14           .          . PUNCT
```

# Strategies for feature selection

- This can lead to a lot of features!
- An example (small) corpus:
  - 17,129 speeches made in the final month of 2016 in the House of Commons
  - $\approx$ 3 million total words
  - 46998 unique words
  - 468244 unique 1-gram and 2-gram sequences

# Strategies for feature selection

- document frequency How many documents in which a term appears
- term frequency How many times does the term appear in the corpus
- deliberate disregard Use of “stop words’ ’: words excluded because they represent linguistic connectors of no substantive content
- purposive selection Use of a *dictionary* of words or phrases

# Common English stop words

```r
library("quanteda")
cat(paste0(stopwords("en"), collapse = "; "))
```

i; me; my; myself; we; our; ours; ourselves; you; your; yours; yourself; yourselves; he; him; his; himself; she; her; hers; herself; it; its; itself; they; them; their; theirs; themselves; what; which; who; whom; this; that; these; those; am; is; are; was; were; be; been; being; have; has; had; having; do; does; did; doing; would; should; could; ought; i'm; you're; he's; she's; it's; we're; they're; i've; you've; we've; they've; i'd; you'd; he'd; she'd; we'd; they'd; i'll; you'll; he'll; she'll; we'll; they'll; isn't; aren't; wasn't; weren't; hasn't; haven't; hadn't; doesn't; don't; didn't; won't; wouldn't; shan't; shouldn't; can't; cannot; couldn't; mustn't; let's; that's; who's; what's; here's; there's; when's; where's; why's; how's; a; an; the; and; but; if; or; because; as; until; while; of; at; by; for; with; about; against; between; into; through; during; before; after; above; below; to; from; up; down; in; out; on; off; over; under; again; further; then; once; here; there; when; where; why; how; all; any; both; each; few; more; most; other; some; such; no; nor; not; only; own; same; so; than; too; very; will

- But no list should be considered universal...

# Common English stop words

```r
cat(paste0(stopwords("smart"), collapse = "; "))
```

a; a's; able; about; above; according; accordingly; across; actually; after; afterwards; again; against; ain't; all; allow; allows; almost; alone; along; already; also; although; always; am; among; amongst; an; and; another; any; anybody; anyhow; anyone; anything; anyway; anyways; anywhere; apart; appear; appreciate; appropriate; are; aren't; around; as; aside; ask; asking; associated; at; available; away; awfully; b; be; became; because; become; becomes; becoming; been; before; beforehand; behind; being; believe; below; beside; besides; best; better; between; beyond; both; brief; but; by; c; c'mon; c's; came; can; can't; cannot; cant; cause; causes; certain; certainly; changes; clearly; co; com; come; comes; concerning; consequently; consider; considering; contain; containing; contains; corresponding; could; couldn't; course; currently; d; definitely; described; despite; did; didn't; different; do; does; doesn't; doing; don't; done; down; downwards; during; e; each; edu; eg; eight; either; else; elsewhere; enough; entirely; especially; et; etc; even; ever; every; everybody; everyone; everything; everywhere; ex; exactly; example; except; f; far; few; fifth; first; five; followed; following; follows; for; former; formerly; forth; four; from; further; furthermore; g; get; gets; getting; given; gives; go; goes; going; gone; got; gotten; greetings; h; had; hadn't; happens; hardly; has; hasn't; have; haven't; having; he; he's; hello; help; hence; her; here; here's; hereafter; hereby; herein; hereupon; hers; herself; hi; him; himself; his; hither; hopefully; how; howbeit; however; i; i'd; i'll; i'm; i've; ie; if; ignored; immediate; in; inasmuch; inc; indeed; indicate; indicated; indicates; inner; insofar; instead; into; inward; is; isn't; it; it'd; it'll; it's; its; itself; j; just; k; keep; keeps; kept; know; knows; known; l; last; lately; later; latter; latterly; least; less; lest; let; let's; like; liked; likely; little; look; looking; looks; ltd; m; mainly; many; may; maybe; me; mean; meanwhile; merely; might; more; moreover; most; mostly; much; must; my; myself; n; name; namely; nd; near; nearly; necessary; need; needs; neither; never; nevertheless; new; next; nine; no; nobody; non; none; noone; nor; normally; not; nothing; novel; now; nowhere; o; obviously; of; off; often; oh; ok; okay; old; on; once; one; ones; only; onto; or; other; others; otherwise; ought; our; ours; ourselves; out; outside; over; overall; own; p; particular; particularly; per; perhaps; placed; please; plus; possible; presumably; probably; provides; q; que; quite; qv; r; rather; rd; re; really; reasonably; regarding; regardless; regards; relatively; respectively; right; s; said; same; saw; say; saying; says; second; secondly; see; seeing; seem; seemed; seeming; seems; seen; self; selves; sensible; sent; serious; seriously; seven; several; shall; she; should; shouldn't;

# Stemming words

**Lemmatization** refers to the algorithmic process of converting words to their lemma forms.

**stemming** the process for reducing inflected (or sometimes derived) words to their stem, base or root form. Different from *lemmatization* in that stemmers operate on single words without knowledge of the context.

**both** convert the morphological variants into stem or root terms

**example:** produc from
`production, producer, produce, produces, produced`

**example II:** saw
Lemmatization may covert to either see or saw depending on whether usage was as a noun or a verb

# Feature selection in practice

`debates18` includes 89416 speeches made in 2018 in the House of Commons

```r
# Construct DFM
debate_dfm <- dfm(debates18$texts)

# Stopwords
debate_dfm_stop <- dfm_remove(debate_dfm, pattern = stopwords("en"))

# Stem
debate_dfm_stem <- dfm_wordstem(debate_dfm)

# Trim (word frequency)
debate_dfm_trim1 <- dfm_trim(debate_dfm, min_termfreq = 5)

# Trim (document frequency)
debate_dfm_trim2 <- dfm_trim(debate_dfm, min_docfreq = 0.001,
                             docfreq_type = "prop")
```

# Feature selection in practice

- 72404 unique words
  - After stopwords: 72232
  - ... and stemming: 49108
  - ... and removing features that appear fewer than 5 times: 29202
  - ... and removing features in fewer than 0.001 documents: 6482
- Feature selection matters! See Denny and Spirling, 2017
  - Just seven (binary) preprocessing decisions leads to a total of $2^7 = 128$ possible feature matrices
  - These selection decisions can have substantive implications for the inferences we draw from QTA

# Descriptive Text Analysis

# Basic descriptive summaries of text

**Length** in characters, words, unique words, lines, sentences, paragraphs, pages, sections, chapters, etc.

**Key words in context** provide how words or phrases are used in a corpus.

**Readability statistics** Use a combination of syllables and sentence length to indicate “readability” in terms of complexity

**Vocabulary diversity** At its simplest involves measuring a *type-to-token ratio* (TTR) where unique words are types and the total words are tokens

**Word (relative) frequency** Measures how often some word occurs relative to some other word

# Describe your text data!

| Speaker | Party | Tokens | Types |
|---|---|---|---|
| Brian Cowen | FF | 5,842 | 1,466 |
| Brian Lenihan | FF | 7,737 | 1,644 |
| Ciaran Cuffe | Green | 1,141 | 421 |
| John Gormley (Edited) | Green | 919 | 361 |
| John Gormley (Full) | Green | 2,998 | 868 |
| Eamon Ryan | Green | 1,513 | 481 |
| Richard Bruton | FG | 4,043 | 947 |
| Enda Kenny | FG | 3,863 | 1,055 |
| Kieran ODonnell | FG | 2,054 | 609 |
| Joan Burton | LAB | 5,728 | 1,471 |
| Eamon Gilmore | LAB | 3,780 | 1,082 |
| Michael Higgins | LAB | 1,139 | 437 |
| Ruairi Quinn | LAB | 1,182 | 413 |
| Arthur Morgan | SF | 6,448 | 1,452 |
| Caoimhghin O'Caolain | SF | 3,629 | 1,035 |
| All Texts | | 49,019 | 4,840 |
| *Min* | | 919 | 361 |
| *Max* | | 7,737 | 1,644 |
| *Median* | | 3,704 | 991 |
| *Hapaxes with Gormley Edited* | | 67 | |
| *Hapaxes with Gormley Full Speech* | | 69 | |

# Key Words in Context

**KWIC** *Key words in context*: A KWIC shows how a word or phrase is used across various texts in the corpus

```
debate_corpus <- corpus(debates18, text_field = "texts")
head(kwic(debate_corpus, "European"))
```

```
## Keyword-in-context with 6 matches.
##  [text3, 102]  still in negotiations with the | European | Union in terms of delivering
##   [text5, 44]     day of consideration of the | European | Union Bill by the Committee
##   [text7, 55]           remain a party to the | European | convention on human rights after
##   [text7, 73]        is also reflected in the | European | Union Act 2018, which
##  [text15, 18] constituency voted to leave the | European | Union in the referendum.
##   [text37, 2]                             The | European | Union's negotiating position on the
```

The idea of a local “context” is central to more advanced QTA analyses such as word-embeddings.

# Lexical Diversity

- Basic measure is the TTR: Type-to-Token ratio

$$TTR = \frac{\text{Number of Types}(V)}{\text{Number of Tokens}(N)}$$

- Problem 1: Very sensitive to overall document length, as shorter texts may exhibit fewer word repetitions
- Problem 2: length may relate to the introdution of additional subjects, which will also increase richness

# Lexical diversity and corpus length

- In natural language text, the rate at which new types appear is very high at first, but diminishes with added tokens

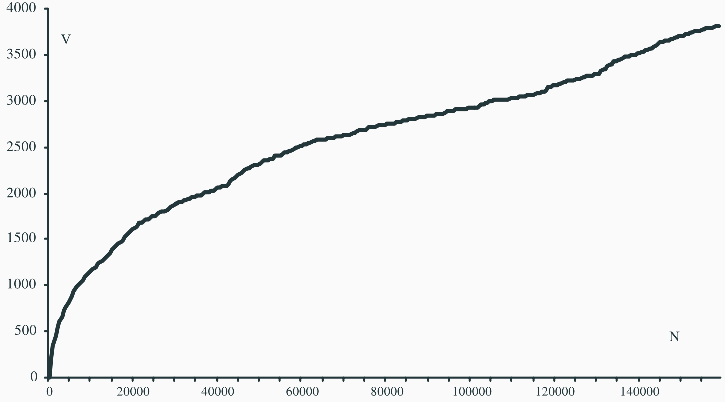



Fig. 1. Chart of vocabulary growth in the tragedies of Racine (chronological order, 500 token intervals).

# Lexical Diversity Example

- Variations use automated segmentation – here approximately 500 words in a corpus of serialized, concatenated weekly addresses by de Gaulle (from Labb'e et. al. 2004)
- While most were written, during the period of December 1965 these were more spontaneous press conferences

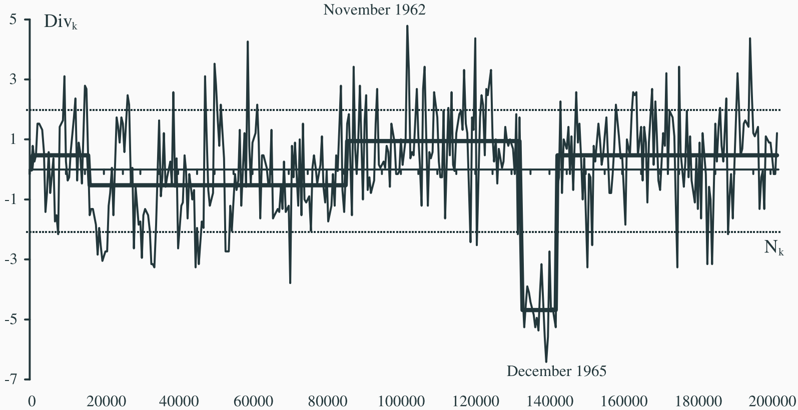
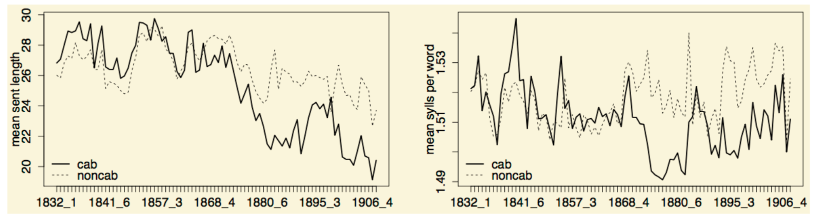



Fig. 8. Evolution of vocabulary diversity in General de Gaulle's broadcast speeches (June 1958–April 1969).

# Readability Example (Spirling, 2015)

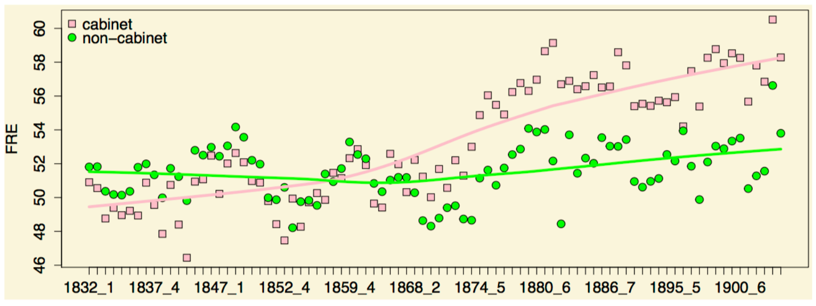
- Most commonly used readability scores focus on a combination of syllables and sentence length
  - Shorter sentences = more readable
  - Fewer syllables = more readable
- Research question: Do Members of Parliament use less complex language when appealing to a more diverse electorate?
- Context: Parliamentary speeches before and after the Great Reform Act (1867)



# Readability Example (Spirling, 2015)

Flesch score:

$$206.835 - 1.105\left(\frac{\text{total number of words}}{\text{total number of sentences}}\right) - 84.6\left(\frac{\text{total number of syllables}}{\text{total number of words}}\right)$$



# Readability Example (Benoit, Spirling, and Munger (2019))

Are these simple measures really sufficient? What might be missing?

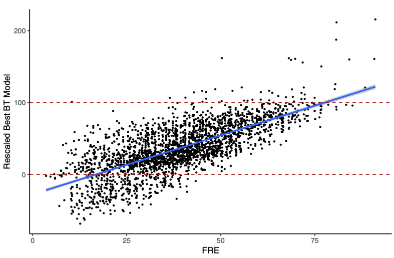
1. **Other features of complexity/readability** (word rarity; Syntactic and grammatical structure)
    - Use relative frequency of terms compared to “the” in google books (dynamic over time)
    - Use number of clauses; proportion of nouns/verbs/adjectives/adverbs
2. **In-domain validation** (are the predictors of “complexity” the same in politics and education?)
    - Crowdsource comparison task of pairs of political sentences (SOTU addresses)
3. **Uncertainty estimates** (is a text with FRE = 50 really more readable than one with FRE = 55?)
    - Bradley-Terry model for paired comparisons to provide probabilistic statements of relative complexity

# Readability Example (Benoit, Spirling, and Munger (2019))

Findings:

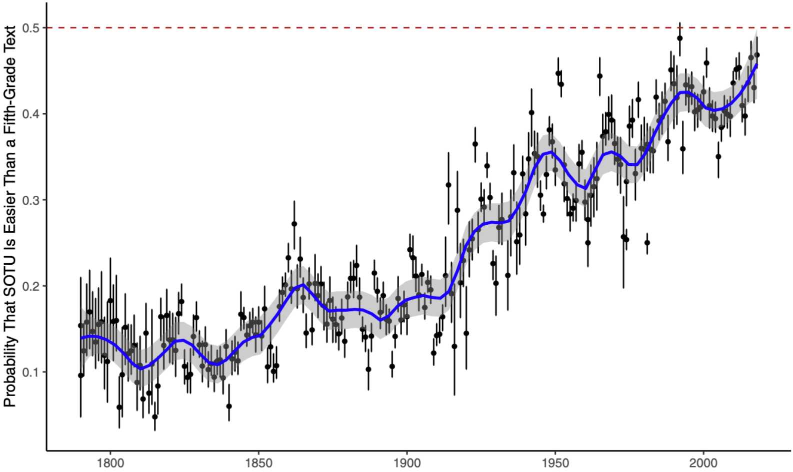
1. **Most important predictors** are sentence length, the proportion of nouns, word rarity, word length
    - Sound familiar?
2. **Modest improvement** over FRE score (3 percentage point improvement over 70% baseline)
3. **Very high correlation** with basic Flesch measure

FIGURE 1 Comparing Our Rescaled Measure to the FRE of the Snippets



# Readability Example (Benoit, Spirling, and Munger (2019))

**FIGURE 2 Probability That a State of the Union Address Is Easier to Understand Than a Fifth Grade Text Baseline, Compared to FRE**



# Text summaries in practice

Thankfully, `quanteda` makes it trivial to calculate many of these statistics...

```r
debate_toks <- tokens(debate_corpus)

# Number of tokens
debate_tokens <- ntoken(debate_toks)
head(debate_tokens)
```

```
## text1 text2 text3 text4 text5 text6
##    91    83   126    77   142   240
```

```r
# Number of types
debate_types <- ntype(debate_toks)
head(debate_types)
```

```
## text1 text2 text3 text4 text5 text6
##    69    53    79    54    92   120
```

# Text summaries in practice

```
# Token-type ratio
library("quanteda.textstats")
debate_ttr <- textstat_lexdiv(debate_toks, "TTR")
head(debate_ttr, n = 3)
```

```
##   document       TTR
## 1    text1 0.7619048
## 2    text2 0.6315789
## 3    text3 0.6428571
```

```
# Readability
debate_read <- textstat_readability(debate_corpus,
                                    measure = "Flesch")
head(debate_read, n = 3)
```

```
##   document   Flesch
## 1    text1 50.23882
## 2    text2 32.45974
## 3    text3 63.40500
```

# Break

Go here: https://jblumenau.shinyapps.io/validate/

# Content Analysis

## Hand-coding: “Classic” content analysis

- Key feature: use of “human” coders to implement a pre-defined coding scheme, by reading and coding texts
- Human decision-making is the central feature of coding decisions, not a computer or other mechanized tool
- Example: hand-coding sentences into pre-defined categories
- Alternative 1: dictionary-based approaches (somewhat more automated)
    - More on this in about 2 minutes
- Alternative 2: inductive scaling or clustering of texts from the quantitative matrix (entirely automated)
    - More on this tomorrow

# Hand-coding: “Classic” content analysis

- Validity is usually the objective, rather than reliability
    - Validity: am I measuring what I am claiming to measure?
    - Reliability: am I able to reliably replicate my coding?
- Another motivating factor could be ease of use, or the difficulty of implementing an automated procedure
- May be *computer-assisted*, especially for unitization
- Many common “CATA” tools exist – e.g. QDAMiner

# Components of classical content analysis designs

**Unitizing** The systematic distinguishing of segments of text that are of interest to the analysis.

**Sampling** Choice (and justification of the choice) of text units to sample, from population of possible text units.

**Coding** Classifying each coded unit of text from the sample according to the pre-defined category scheme.

**Summarizing** Reducing the coded data to summary quantities of interest.

**Inference and reporting** The final steps wherein the analyzed results are used to generalize about social world, and communicating these results to others.

# Dictionary Analysis

# Motivation

**Are female politicians less aggressive than male politicians?**

A repeated claim in the qualitative literature on gender and politics is that female politicians have a distinct style from male politicians. Many political observers argue that women are less aggressive in political debate than their male colleagues. Most of the evidence for these claims is taken from small-N classical content analysis studies. We will review this question by applying an existing sentiment dictionary to a large-N corpus of parliamentary texts.

# Bridging qualitative and quantitative text analysis

- A hybrid procedure between qualitative and quantitative classification the fully automated end of the text analysis spectrum
- "Qualitiative" since it involves identification of the concepts and associated keys/categories, and the textual features associated with each key/category
- Dictionary construction involves a lot of contextual interpretation and qualitative judgment
- Perfect reliability because there is no human decision making as part of the text analysis procedure

# Rationale for dictionaries

- Rather than count words that occur, pre-define words associated with specific meanings
- Two components:
    1. key: the label for the equivalence class for the concept or canonical term e.g. "dog"
    2. values: (multiple) terms or patterns that are declared equivalent occurences of the key class e.g. "Dalmatian", "Labrador", "Poodle"
- Frequently involves lemmatization: transformation of all inflected word forms to their "dictionary look-up form'' – more powerful than stemming

At its simplest, a dictionary is just a list of words ($m = 1, ..., M$) that is related to a common concept.

| Aggression |
|---|
| stupid |
| dishonest |
| lier |
| idiot |
| ignorant |
| hate |
| fight |
| battle |

# Counting words

Applying a dictionary to a corpus of texts $(i = 1, ..., N)$ simply requires counting the number of times each word occurs in each text and summing them.

If $W_{im}$ is a vector measuring 1 if word $m$ appears in text $i$ and 0 otherwise, then the dictionary score for document $i$ is:

$$t_i = \frac{\sum_{m=1}^{M} W_{im}}{N_i}$$

Or, the proportion of words in document $i$ that appear in the dictionary.

# Counting words

*"That statement is as barbaric as it is downright stupid; it is nothing more than an ignorant, cruel and deliberate misconception to hide behind."*

$$t_i = \frac{\sum_{m=1}^{M} W_{im}}{N_i} = \frac{1+1}{14} = 0.14$$

# Counting *weighted* words

A slight development on this would be to assign each word in the dictionary a weight which reflects something about the importance of the word to the concept

| Aggression | Weight |
|---|---|
| stupid | .6 |
| dishonest | .2 |
| lie | .5 |
| idiot | .7 |
| ignorant | .3 |
| brutal | .4 |
| violence | .5 |

Note that weights are implicit in *all* dictionary approaches. Typically, all words are counted equally which implies a score of 1 for all words. This is not necessarily correct!

# Counting *weighted* words

We can adjust the previous formula to incorporate the weights ($s_m$):

$$t_i = \frac{\sum_{m=1}^{M} s_m W_{im}}{N_i}$$

Why normalise by $N_i$? Some texts will be longer than others and we do not want these texts to mechanically be assigned higher scores.

# Counting *weighted* words

*"That statement is as barbaric as it is downright stupid; it is nothing more than an ignorant, cruel and deliberate misconception to hide behind."*

$$t_i = \frac{\sum_{m=1}^{M} s_m W_{im}}{N_i} = \frac{(1 \cdot 0.6) + (1 \cdot 0.3)}{14} = 0.06$$

# Weights or no weights?

Most applications of dictionary methods in the social science and industry applications use unweighted dictionary approaches.

Why learn this then?

1. The equal weighting assumption is not necessarily reasonable or effective
2. The idea of assigning weights to words is something that will come up in the context of supervised learning and topic models

# Advantages of dictionaries: Many existing implementations

Linquistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match 82 language dimensions
- $\approx$ 4,500 words and word stems, each defining one or more word categories
- For example, the word *cried* is part of five word categories: sadness, negative emotion, overall affect, verb, and past tense verb.
- Hierarchical: so "anger" is part of an *emotion* category and a *negative emotion* subcategory
- You can buy it here: http://www.liwc.net/descriptiontable1.php

# Example: Terrorist speech (Pennebaker and Chung, 2009)

| | Bin Ladin (1988 to 2006) N = 28 | Zawahiri (2003 to 2006) N = 15 | Controls N = 17 | p (two-tailed) |
|---|---|---|---|---|
| Word Count | 2511.5 | 1996.4 | 4767.5 | |
| Big words (greater than 6 letters) | 21.2a | 23.6b | 21.1a | .05 |
| Pronouns | 9.15ab | 9.83b | 8.16a | .09 |
| I (e.g. I, me, my) | 0.61 | 0.90 | 0.83 | |
| We (e.g. we, our, us) | 1.94 | 1.79 | 1.95 | |
| You (e.g. you, your, yours) | 1.73 | 1.69 | 0.87 | |
| He/she (e.g. he, hers, they) | 1.42 | 1.42 | 1.37 | |
| They (e.g., they, them) | 2.17a | 2.29a | 1.43b | .03 |
| Prepositions | 14.8 | 14.7 | 15.0 | |
| Articles (e.g. a, an, the) | 9.07 | 8.53 | 9.19 | |
| Exclusive Words (but, exclude) | 2.72 | 2.62 | 3.17 | |
| Affect | 5.13a | 5.12a | 3.91b | .01 |
| Positive emotion (happy, joy, love) | 2.57a | 2.83a | 2.03b | .01 |
| Negative emotion (awful, cry, hate) | 2.52a | 2.28ab | 1.87b | .03 |
| Anger words (hate, kill) | 1.49a | 1.32a | 0.89b | .01 |
| Cognitive Mechanisms | 4.43 | 4.56 | 4.86 | |
| Time (clock, hour) | 2.40b | 1.89a | 2.69b | .01 |
| Past tense verbs | 2.21a | 1.63a | 2.94b | .01 |
| Social Processes | 11.4a | 10.7ab | 9.29b | .04 |
| Humans (e.g. child, people, selves) | 0.95ab | 0.52a | 1.12b | .05 |
| Family (mother, father) | 0.46ab | 0.52a | 0.25b | .08 |
| Content | | | | |
| Death (e.g. dead, killing, murder) | 0.55 | 0.47 | 0.64 | |
| Achievement | 0.94 | 0.89 | 0.81 | |
| Money (e.g. buy, economy, wealth) | 0.34 | 0.38 | 0.58 | |
| Religion (e.g. faith, Jew, sacred) | 2.41 | 1.84 | 1.89 | |

Note. Numbers are mean percentages of total words per text file. Statistical tests are between Bin Ladin, Zawahiri, and Controls. Documents whose source indicates "Both" (n=3) or

# Example: Terrorist speech (Pennebaker and Chung, 2009)

> *The analysis of the al-Zawahiri and bin Laden files suggest somewhat different speaking and, by extension, thinking styles*

Maybe, but this requires us believing that the number of big words, pronouns, and references to affect and social processes reflects underlying charactersitics of the authors!

# Advantages of dictionaries: Multi-lingual

APPENDIX B
DICTIONARY OF THE COMPUTER-BASED CONTENT ANALYSIS

| | NL | UK | GE | IT |
|---|---|---|---|---|
| **Core** | elit* | elit* | elit* | elit* |
| | consensus* | consensus* | konsens* | consens* |
| | ondemocratisch* | undemocratic* | undemokratisch* | antidemocratic* |
| | ondemokratisch* | | | |
| | referend* | referend* | referend* | referend* |
| | corrupt* | corrupt* | korrupt* | corrot* |
| | propagand* | propagand* | propagand* | propagand* |
| | politici* | politici* | politiker* | politici* |
| | *bedrog* | *deceit* | täusch* | ingann* |
| | *bedrieg* | *deceiv* | betrüg* | |
| | | | betrug* | |
| | *verraa* | *betray* | *verrat* | tradi* |
| | *verrad* | | | |
| | schaam* | shame* | scham* | vergogn* |
| | | | schäm* | |
| | schand* | scandal* | skandal* | scandal* |
| | waarheid* | truth* | wahrheit* | verità |
| | oneerlijk* | dishonest* | unfair* | disonest* |
| | | | unehrlich* | |
| **Context** | establishm* | establishm* | establishm* | partitocrazia |
| | heersend* | ruling* | *herrsch* | |
| | capitul* | | | |
| | kapitul* | | | |
| | kaste* | | | |
| | leugen* | | lüge* | menzogn* |
| | lieg* | | | mentir* |

# Advantages of dictionaries: Fast and easy to apply

Here, `debates` is a `data.frame` of parliamentary debates, which contains about a million speeches.

```
brexit_dict <- dictionary(list(brexit = c("brexit",
                                           "leave",
                                           "remain",
                                           "sovereignty",
                                           "control")))

dictionary_dfm <- tokens(debates18$texts) %>%
  tokens_lookup(dictionary = brexit_dict) %>%
  dfm()
```

# Advantages of dictionaries: Fast and easy to apply

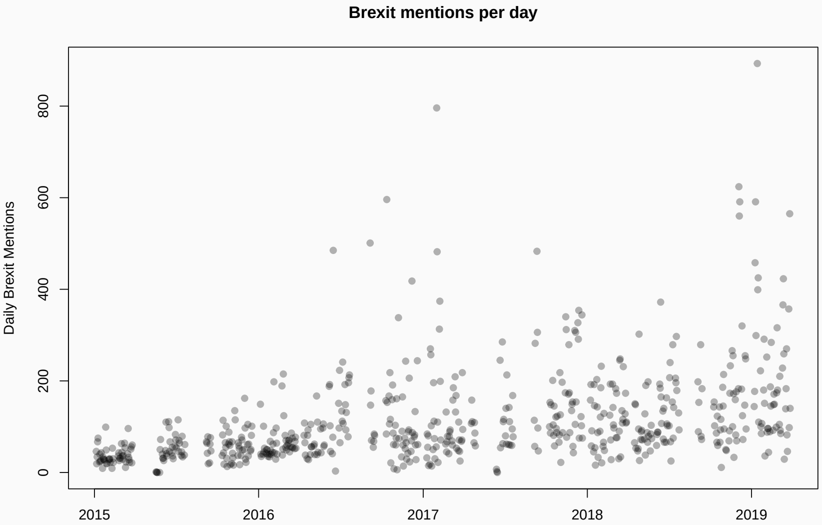
```
dfm_subset(dictionary_dfm, ntoken(dictionary_dfm) > 0)
```

```
## Document-feature matrix of: 1,278 documents, 1 feature (0.00% sparse) and 0 docvars.
##          features
## docs      brexit
##   text7        1
##   text9        1
##   text15       2
##   text18       1
##   text21       1
##   text24       1
## [ reached max_ndoc ... 1,272 more documents ]
```

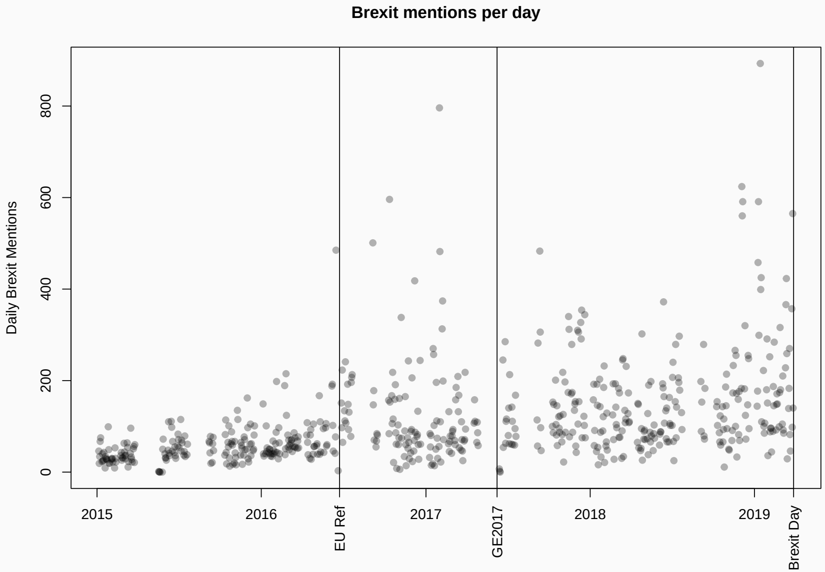
In contrast to some of the other methods we will study, dictionaries can be easily applied to thousands of texts in a matter of seconds.

The code above runs in about a minute.

# Advantages of dictionaries: Fast and easy to apply



# Advantages of dictionaries: Fast and easy to apply



# Disdvantages of dictionaries

- Problem 1: polysemes – words that have multiple meanings
  - Loughran and McDonald used the Harvard-IV-4 TagNeg (H4N) file to classify sentiment for a corpus of 50,115 firm-year 10-K filings from 1994–2008
  - Almost three-fourths of the “negative” words of H4N were typically not negative in a financial context:

    e.g. *mine* or *cancer*, or *tax*, *cost*, *capital*, *board*, *liability*, *foreign*, and *vice*
- Problem 2: Dictionaries often lack important negative financial words, for example; *felony*, *litigation*, *restated*, *misstatement*, and *unanticipated*
- Problem 3: Some dictionaries might do more to pick up the *topic* of a document than the *tone* of a document

# Disdvantages of dictionaries

*"That statement is as barbaric as it is downright stupid; it is nothing more than an ignorant, cruel and deliberate misconception to hide behind."*

*"Terrible acts of brutality and violence have been carried out against the Rohingya people."*

- Dictionaries may miss words that are important to the concept
  - "barbaric" is probably an aggressive word in this context
- Dictionaries do not typically capture modifiers
  - "downright" is an intensifier (also: negators like "not good")
- Dictionaries often fail to capture all synonyms
  - "deliberate misconception" is parliamentary language for "lie"
- Dictionaries may not capture the relevant concept
  - brutality/violence: descriptions, rather than *expressions*, of aggression

# Validation

# What kind of validation might we use here?

Applying dictionaries outside the domain for which they were developed can lead to errors.

One way of assessing the seriousness of these errors is to conduct **validation tests**

Main idea: are the texts that are flagged by the dictionary more representative of the relevant concept than other texts?

# Applying dictionaries in quanteda

```r
library(quanteda)
aggression_words <- read.csv("aggression_words.csv")[,1]
aggression_texts <- read.csv("aggression_texts.csv")[,1]
```

1. `aggression_words` is a vector of 222 words from the an existing "Aggression" dictionary
2. `aggression_texts` is a vector of 10937 sentences from parliamentary speeches

Our goal is to use `aggression_words` to score the texts in `aggression_texts`.

# Applying dictionaries in quanteda

First we convert the texts to a `corpus` object:

```
aggression_corpus <- corpus(aggression_texts)
```

Then we extract the `tokens()` and create a `dfm()`:

```
aggression_tokens <- tokens(aggression_corpus)
aggression_dfm <- dfm(aggression_tokens)
```

And the words to a `dictionary` object:

```
aggression_dictionary <- dictionary(list(aggression = aggression_words))
```

Finally, we “apply” the dictionary to the dfm using the `dfm_lookup` function:

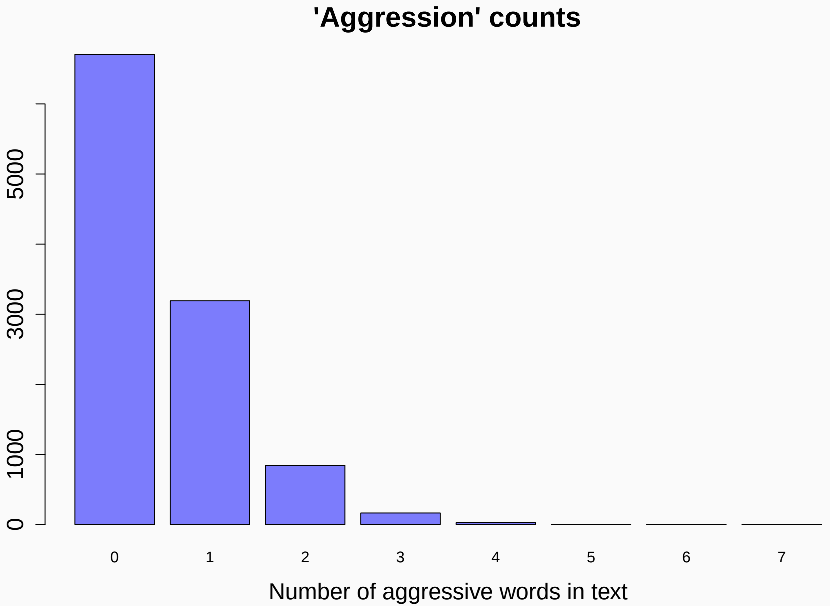
```
aggression_dfm_dictionary <- dfm_lookup(aggression_dfm,
                                        dictionary = aggression_dictionary)
```

# Applying dictionaries in quanteda

```r
print(aggression_dfm_dictionary)
```

```
## Document-feature matrix of: 10,937 documents, 1 feature (79.05% sparse) and 0 docvar
##         features
## docs    aggression
##   text1          0
##   text2          0
##   text3          0
##   text4          0
##   text5          1
##   text6          1
## [ reached max_ndoc ... 10,931 more documents ]
```

`aggression_dfm` is a document-feature matrix, where the only "feature" is the dictionary counts


'Aggression' counts
0
1000
3000
5000
0
1
2
3
4
5
6
7
Number of aggressive words in text

# Applying dictionaries in quanteda

Finally, we can calculate the score by dividing the dictionary counts by the number of words in each text:

```
aggression_proportions <- as.numeric(aggression_dfm_dictionary[,1])
  ntoken(aggression_corpus)

summary(aggression_proportions)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.008109 0.000000 0.190476
```

# Face validity (1)

**Intuition:** Does the measure vary in sensible ways?

In this case, one obvious test is whether MPs speeches are more aggressive during Prime Minister's Questions (PMQs).

# Face validity (1)

```r
coef(summary(lm(aggression_proportions ~ pmq_dummy)))
```

```
##                Estimate   Std. Error  t value     Pr(>|t|)
## (Intercept) 0.008109493 0.0001847772 43.88796 0.000000e+00
## pmq_dummy   0.008363489 0.0004699483 17.79661 5.119374e-70
```

There is clear evidence that PMQ debates tend to have higher levels of aggressive language than other debates.

# Face validity (2)

How does this approach perform? Let's look at the top sentences:

| | score | text |
|---|---|---|
| text3998 | 0.19 | I fully appreciate that it is the Opposition's job to oppose, but there are times when opposition is destructive. |
| text7416 | 0.18 | We unequivocally condemn Hamas's dreadful and murderous rocket attacks and defend Israel's right to defend itself. |
| text2941 | 0.14 | They were asking ridiculous prices, because they had the sole remedy for a complaint, so could exploit that situation. |
| text106 | 0.13 | Terrible acts of brutality and violence have been carried out against the Rohingya people. |
| text144 | 0.13 | The motion condemns the early release scheme for those who have assaulted police officers. |

While some seem reasonable, others indicate that we are picking up topic rather than tone.

# Human validation as a gold standard

What is the “gold standard” for judging whether our dictionary works?

Typically, we compare the performance of our method to human judgements of our concept of interest.

In essence, we can ask people to rate sentences according to their “aggressiveness” and see whether this correlates with our measure.

**Key assumption:** Human coders can accurately and reliably recognise instances of aggression in text.

# Which of these questions is easier?

1. On a scale from 0 to 100, how aggressive is this sentence?
    - "I regard it as an essential weapon in the armoury of the fight against terrorism."
2. Which of these sentences is more aggressive?
    - "I regard it as an essential weapon in the armoury of the fight against terrorism."
    - "I also welcome the fact that the Bill will encourage more young people to take advantage of the programme."

Paired comparisons tend to give more useful and reliable information than single ratings.

# Set-up

1. Apply 7 basic QTA measures (including 6 dictionaries) to 8 million sentences
    - Aggression
    - Positive Emotion
    - Negative Emotion
    - Fact
    - Anecdote
    - Complexity
    - Repetition
2. Score each sentence using uniform word weights
3. Present pairs of sentences to human coders and ask them to select which sentence is most representative of a certain concept

# Validation app

Go here: https://jblumenau.shinyapps.io/validate/

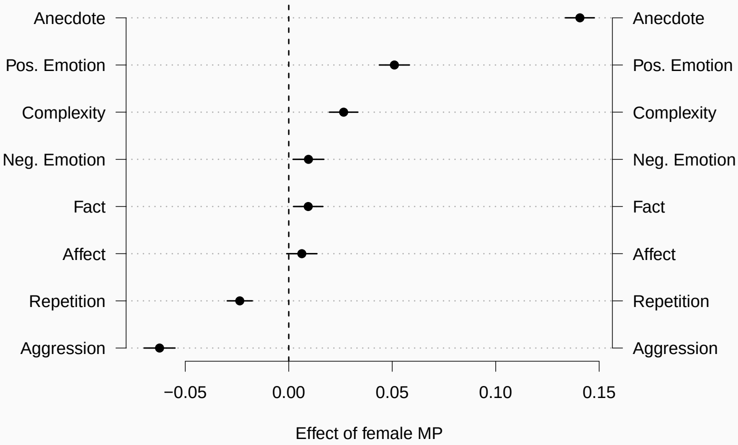
# Validation measure

Does the difference in sentence-level dictionary scores predict human judgements?

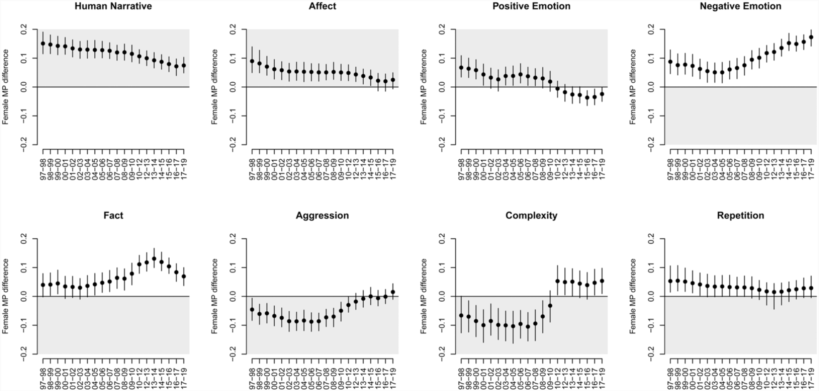
- Sample pairs of sentences from the corpus
    - Score each pair as $\text{Diff}_i = \text{Style Score}_{1i} - \text{Style Score}_{2i}$
- Randomly present to human coders, code $(Y_i)$ whether:
    - Sentence one is more <style> (1)
    - About the same (0)
    - Sentence two is more <style> (-1)
- Calculate the relationship between human coding and dictionaries by:
    - $Y_i = \alpha + \beta \text{Diff}_i$
    - $Cor(Y_i, \text{Diff}_i)$
- Repeat for each dictionary

# Are women less aggressive?

Let's believe for a second that our validation strategy worked.



# Have male/female political styles changed over time aggressive?



# Conclusion

# Conclusion

- QTA allows us to draw inferences from very large collections of text without (too much) human interpretation
- All quantitative models of text are wrong, but some are useful
- Simple quantitative metrics of text can be very revealing
- `quanteda` is awesome
- Validation is very important!

# Road map

For the rest of the week, we will build upon the tools we covered today

- Tuesday: Supervised learning with text, and text scaling models
- Wednesday: Unsupervised text models (topic models)
- Thursday: Data from the web