TEXT AS DATA: WEEK 8

MATTHIAS HABER

03 NOVEMBER 2021

GOALS FOR TODAY

GOALS

- Dictionary approaches
- Dictionaries in quanteda
- Sentiment analysis

DICTIONARY APPROACHES

DICTIONARY APPROACHES

- Dictionaries help classifying texts to categories or determine their content of a known concept
- They are a hybrid between qualitative and quantitative classification.
- Dictionary construction involves a lot of contextual interpretation and qualitative judgment.
  - Which text pertain to which categories?
  - Which texts contain how much of a concept?
- Dictionaries are perfectly reliable because there is no human decision making as part of the text analysis procedure

DICTIONARY APPROACHES

- Rather than count words that occur, pre-define words associated with specific meanings
- Dictionaries consist of keys and values, where the “key” is a category such as “positive” or “negative”, and the “values” consist of the patterns assigned to each key that will be counted as occurrences of those keys
- Dictionaries often require lemmatization rather than stemming

DICTIONARY APPROACHES

- A dictionary is basically just a list of words that is related to a common concept
- Applying them to a corpus of texts simply requires counting the number of times each word in the list occurs in each text and summing them

ADVANTAGES OF DICTIONARIES: IMPLEMENTATIONS

- There are many different implementations of dictionaries for all types of contexts
- Take for example the Linquistic Inquiry and Word Count project craeted by Pennebaker et al: http://www.liwc.net
- Consists of about 4,500 words and word stems, each defining one or more word categories or sub-dictionaries
- For example, the word *cried* is part of five word categories: sadness, negative emotion, overall affect, verb, and past tense verb
- Hierarchical: so “anger” are part of an emotion category and a negative emotion subcategory
- You can buy it here: https://liwcsoftware.onfastspring.com/

EXAMPLE 1: TERRORIST SPEECH

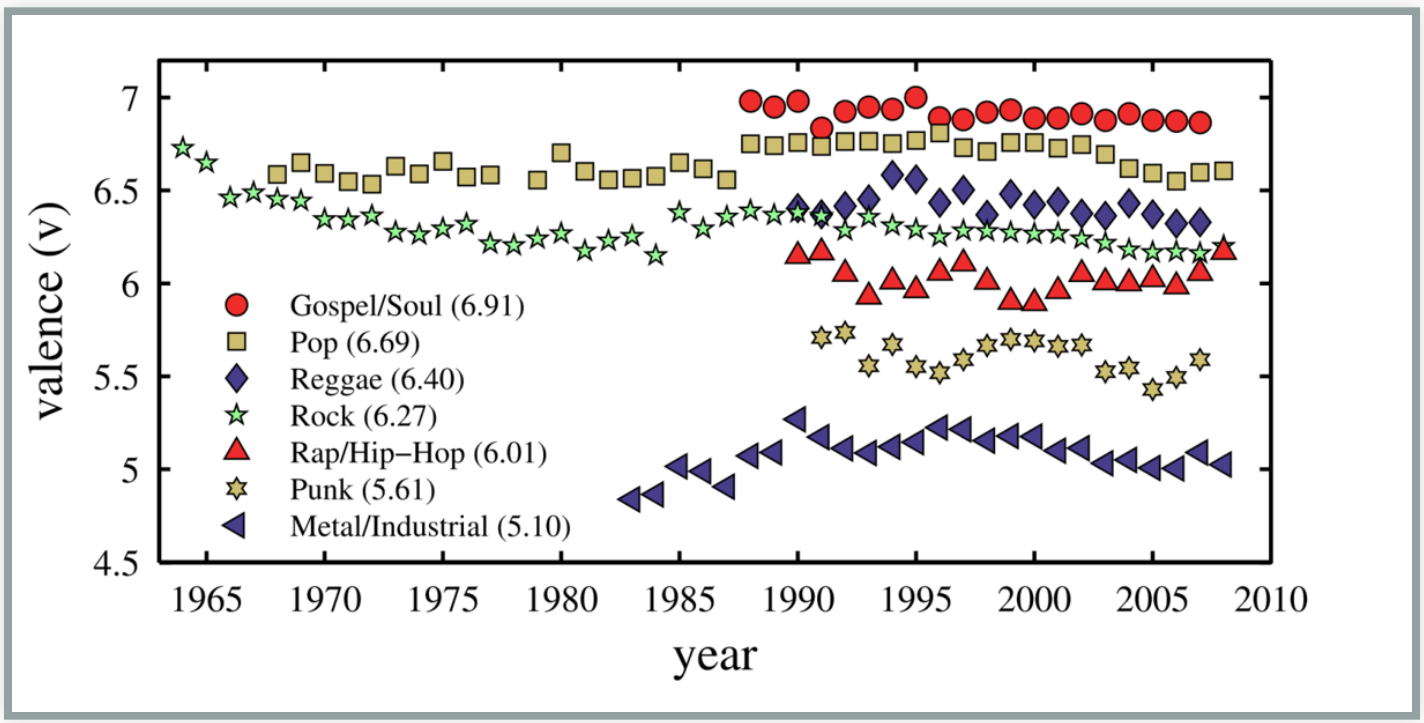
An analysis of terrorist speech (Pennebaker & Chung 2009)

	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 "Unknown" (n=2) were excluded due to their small sample sizes.

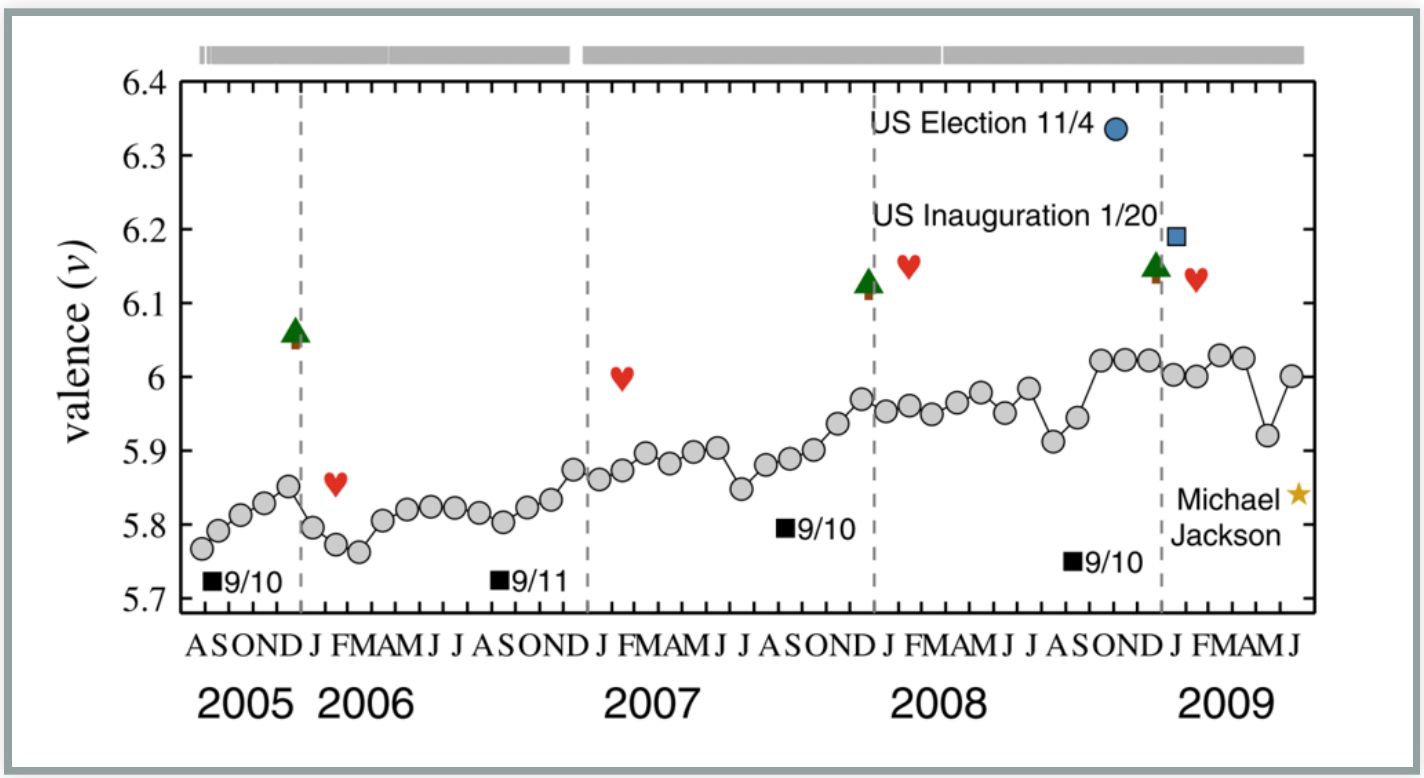
EXAMPLES 2: HAPPINESS IN SONG LYRICS

Valence time series for song titles broken down by representative genres (Dodds & Danforth 2009)



EXAMPLES 3: HAPPINESS IN BLOGS

Time series of average monthly valence for blog sentences starting with "I feel…" (Dodds & Danforth 2009)



ADVANTAGES OF DICTIONARIES: MULTI-LINGUAL

Dictionaries exist for lots of different languages.

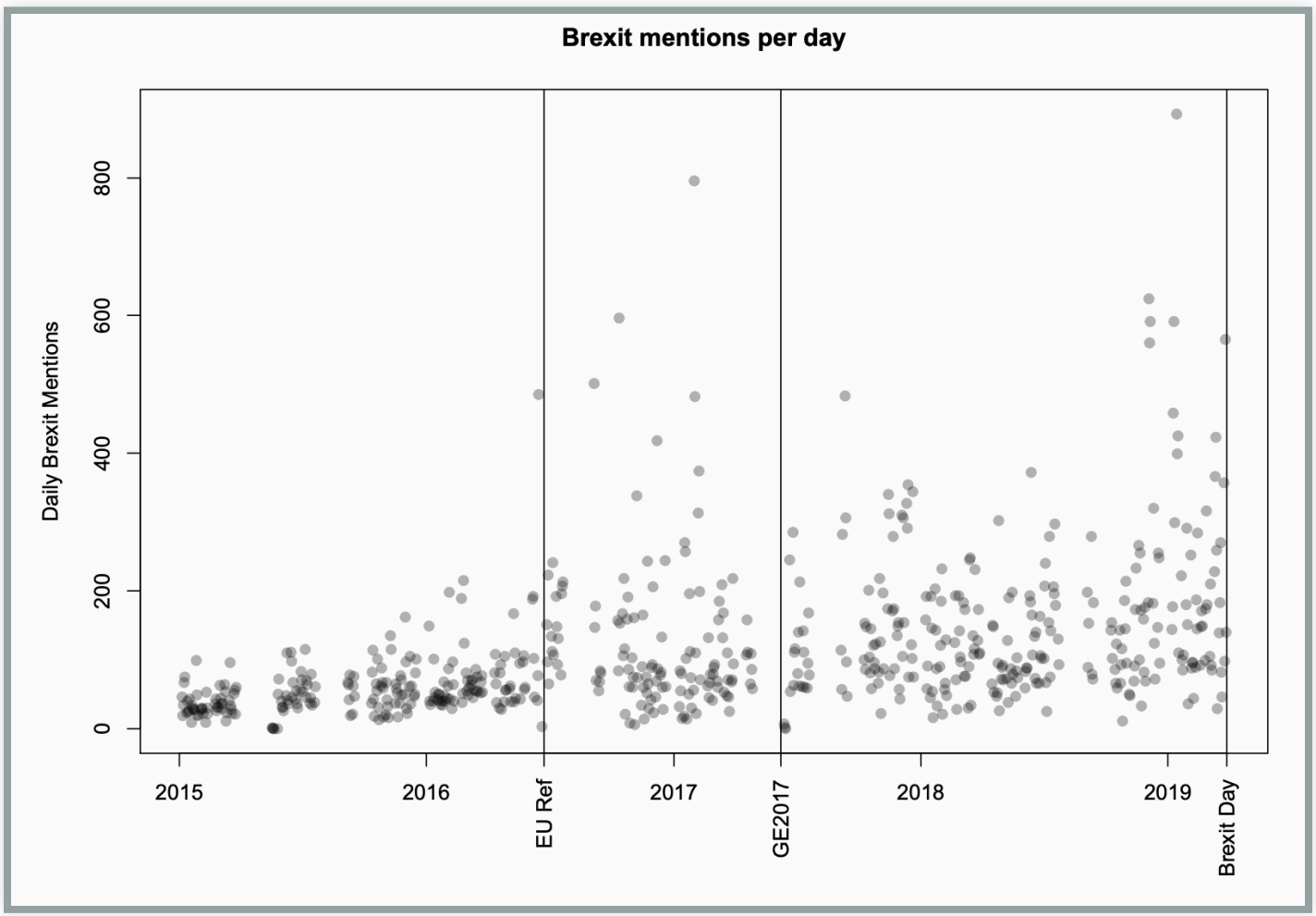
APPENDIX B
DICTIONARY OF THE COMPUTER-BASED CONTENT ANALYSIS

	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*

(from Rooduijn and Pauwels 2011)

ADVANTAGES OF DICTIONARIES: FAST TO APPLY

Dictionaries can by easily applied to thousands of texts in a matter of seconds allowing us to quickly analyze texts



DISADVANTAGE: HIGHLY SPECIFIC TO CONTEXT

- Dictionaries have problems with words that are polysemes i.e. words that have multiple meanings in different contexts
- Dictionaries might miss words that are important to the concept or the tone of a document
- They do not typically capture modifiers (e.g. “not good”)
- They often fail to capture synonyms

CREATING DICTIONARIES

- Collect the words that discriminate between categories/concepts, i.e. create a dictionary
  - Existing dictionaries
  - Creating a dictionary
- Quantify the occurrence of these words in texts
- Validate
- Be careful: Applying dictionaries outside the domain for which they were developed can lead to errors

DICTIONARIES IN QUANTEDA

CREATING A SIMPLE DICTIONARY

To create a simple dictionary of parts of speech, for instance we could define a dictionary consisting of articles and conjunctions, using:

```
pos_dict <- dictionary(list(articles = c("the", "a", "and"),
           conjunctions = c("and", "but", "or", "nor", "for", "yet",
         "so")))
```

CREATING A SIMPLE DICTIONARY

We can use this dictionary when we create a `dfm` to let this define a set of features:

```
pos_dfm <- data_corpus_inaugural %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  dfm() %>%
  dfm_lookup(pos_dict)
```

CREATING A SIMPLE DICTIONARY

Let's recreate the dfm but weight it by document length and compute the share of of articles and conjunctions in each speech:

```
pos_dfm_wgt <- data_corpus_inaugural %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  dfm() %>%
  dfm_weight(scheme = "prop") %>%
  dfm_lookup(pos_dict)
```

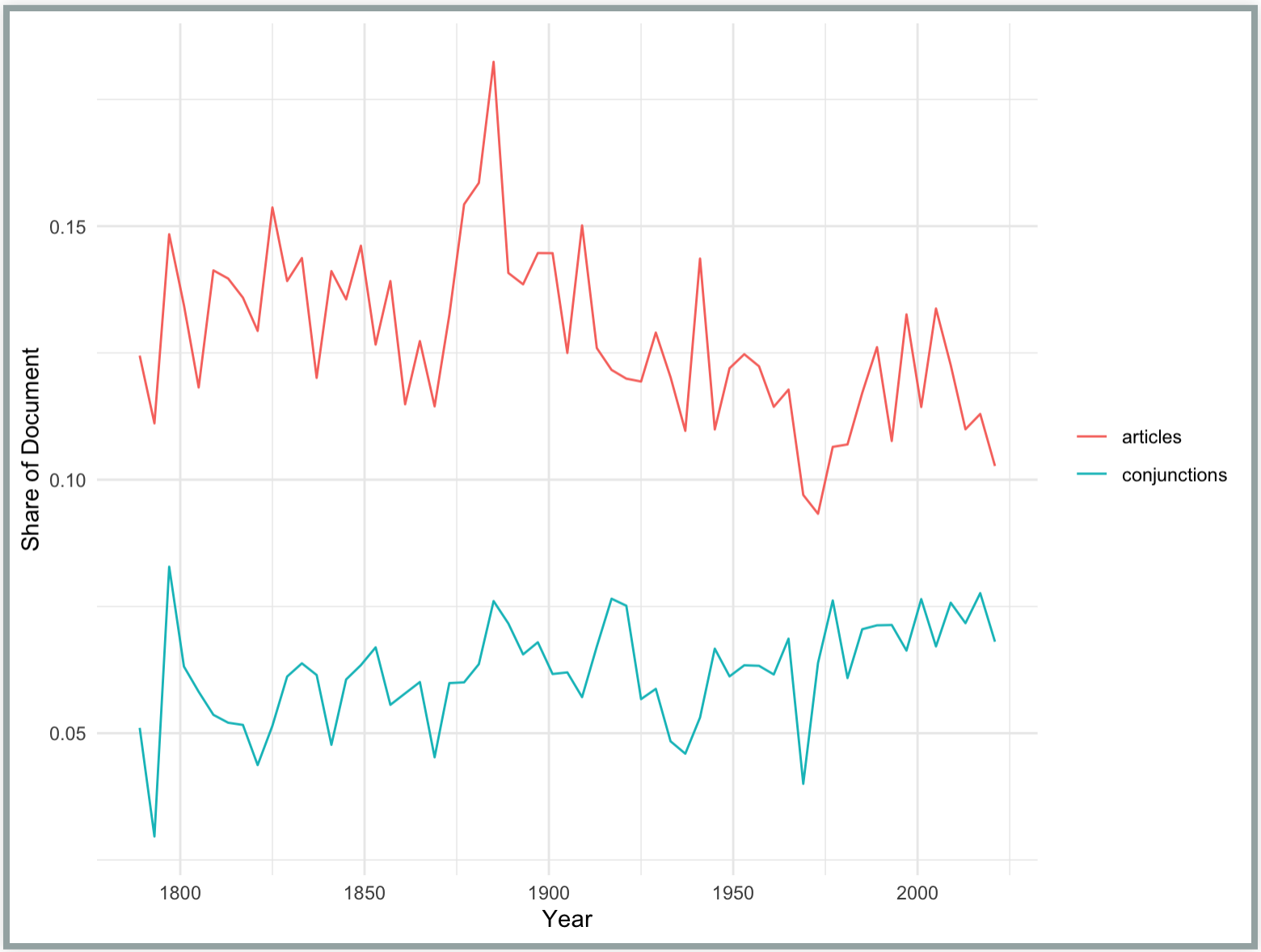
CREATING A SIMPLE DICTIONARY

Let's plot the trends over time. Before we can do that we need to convert the dfm to a data frame and tidy it up a bit.

```
pos_df_wgt <- pos_dfm_wgt %>%
  convert(to = "data.frame") %>%
  cbind(docvars(pos_dfm_wgt)) %>%
  tidyr::gather(pos, share, articles:conjunctions)

pos_df_wgt %>%
ggplot(aes(x = Year, y = share, color = pos)) +
  geom_line() +
  ylab("Share of Document") +
  theme(legend.title = element_blank())
```

CREATING A SIMPLE DICTIONARY



EXERCISE

Create a new dictionary with a concept of your own choosing and apply it to the inaugural speeches. Plot the prevalence of that concept in the speeches (e.g. over time or between different speakers). Paste your results into the chat.

Take about 10-15 minutes.

SENTIMENT ANALYSIS

SENTIMENT ANALYSIS WITH QUANTEDA

Perhaps the most popular type of dictionary are sentiment dictionaries used to assess the valence of a text by searching for words that describe opinions.

There is a new, still somewhat experimental, quanteda package called `quanteda.sentiment` that extends the quanteda package with functions for computing sentiment on text. You need to install it directly from github

```
#devtools::install_github("quanteda/quanteda.sentiment")
library(quanteda.sentiment)
```

QUANTEDA.SENTIMENT

`quanteda.sentiment` has two main functions: `textstat_polarity()` to compute *polarity-based sentiments* (i.e. polar opposites such as republican vs. democrat or negative vs. positive) and `textstat_valence()` to compute *valence-based sentiments* for continuous degrees of sentiments.

The package comes with the following built-in dictionaries:

Name	Description	Polarity	Valence
data_dictionary_AFINN	Nielsen's (2011) 'new ANEW' valenced word list		✔
data_dictionary_ANEW	Affective Norms for English Words (ANEW)		✔

Name	Description	Polarity	Valence
data_dictionary_geninqposneg	Augmented General Inquirer *Positiv* and *Negativ* dictionary	✔	
data_dictionary_HuLiu	Positive and negative words from Hu and Liu (2004)	✔	
data_dictionary_LoughranMcDonald	Loughran and McDonald Sentiment Word Lists	✔	
data_dictionary_LSD2015	Lexicoder Sentiment Dictionary (2015)	✔	
data_dictionary_NRC	NRC Word-Emotion Association Lexicon	✔	
data_dictionary_Rauh	Rauh's German Political Sentiment Dictionary	✔	
data_dictionary_sentiws	SentimentWortschatz (SentiWS)	✔	✔

EXPLORING BUILT-IN DICTIONARIES

You can view the content of those dictionaries by using the `print()` function:

```
print(data_dictionary_geninqposneg, max_nval = 5)
```

```
## Dictionary object with 2 key entries.
## Polarities: pos = "positive"; neg = "negative"
## - [positive]:
##   - abide, ability, able, abound, absolve [ ... and 1,648 more ]
## - [negative]:
##   - abandon, abandonment, abate, abdicate, abhor [ ... and 2,005 more
```

APPLYING A POLARITY-BASED SENTIMENT DICTIONARY

To compute the polarity-based sentiment scores for the most recent inaugural speeches we can simply apply the `textstat_polarity()` function to our corpus and specify the dictionary we wish to use.

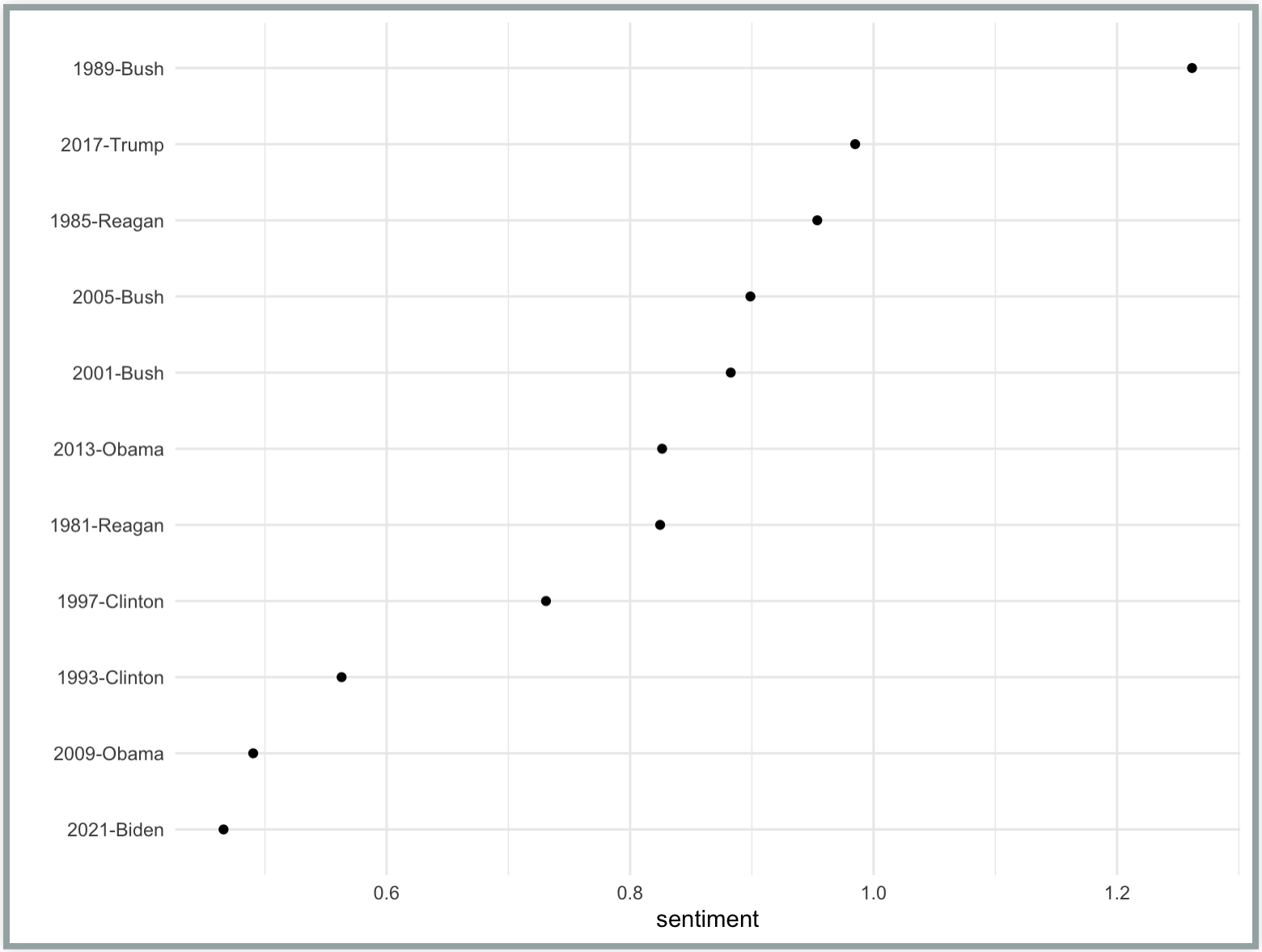
```
data(data_corpus_inaugural, package = "quanteda")
sent_pres <- data_corpus_inaugural %>%
  corpus_subset(Year > 1980) %>%
  textstat_polarity(dictionary = data_dictionary_LSD2015)
tail(sent_pres)
```

```
##         doc_id sentiment
## 6    2001-Bush 0.8826607
## 7    2005-Bush 0.8987976
## 8   2009-Obama 0.4903646
## 9   2013-Obama 0.8262646
## 10 2017-Trump 0.9848534
## 11 2021-Biden 0.4660002
```

APPLYING A POLARITY-BASED SENTIMENT DICTIONARY

Of course we could then also plot the results

```
sent_pres %>%
  ggplot(aes(x = sentiment, y = reorder(doc_id, sentiment))) +
    geom_point() +
    ylab("")
```


1989-Bush
2017-Trump
1985-Reagan
2005-Bush
2001-Bush
2013-Obama
1981-Reagan
1997-Clinton
1993-Clinton
2009-Obama
2021-Biden
0.6
0.8
1.0
1.2
sentiment

APPLYING A VALENCE-BASED SENTIMENT DICTIONARY

If we want to apply a valence-based dictionary instead we would use the `textstat_valence()` function. For example to compute the valence scores using Nielsen's (2011) 'new ANEW' valenced word list:

```
tail(data_corpus_inaugural) %>%
  textstat_valence(dictionary = data_dictionary_AFINN)
```

```
##       doc_id sentiment
## 1  2001-Bush 0.6578947
## 2  2005-Bush 0.7461140
## 3 2009-Obama 0.3586957
## 4 2013-Obama 0.5732484
## 5 2017-Trump 1.0341880
## 6 2021-Biden 0.4739884
```

EXERCISE

Install the package `quanteda.corpora` from github using the `install_github` function from the `devtools`or `remotes` package:

```
#remotes::install_github("quanteda/quanteda.corpora")
library(quanteda.corpora)
```

Download the corpus of 6,000 Guardian news articles using `download("data_corpus_guardian")` and create a plot showing how the sentiment score has changed over time. Paste your results into the chat.

Take about 10-15 Minutes.`

WRAPPING UP

QUESTIONS?

OUTLOOK FOR OUR NEXT SESSION

Next week we will continue our session on text classification by looking into scaling methods

THAT'S IT FOR TODAY

Thanks for your attention!