From Corpus to Features, Part 1

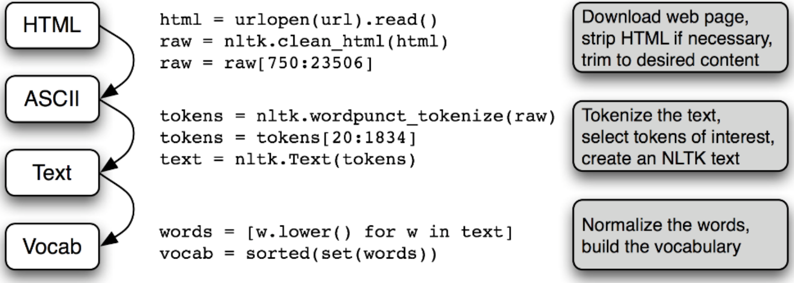
Elliott Ash

Bocconi 2018

Overview

- These slides describe the process of transforming a corpus into numerical data that can be used in statistical analysis.
- Input:
  - A set of documents (e.g. text files), D.
- Output:
  - A matrix, X, containing statistics about phrase frequencies in those documents.

The NLP Pipeline



Source: NLTK Book, Chapter 3.

Split into sentences

- Many tasks should be done on sentences, rather than corpora as a whole.
  - NLTK and spaCy do a good (but not perfect) job of splitting sentences, while accounting for periods on abbreviations, etc.

```
from nltk import sent_tokenize
sentences = sent_tokenize(text)
print(sentences)

import spacy
nlp = spacy.load('en')
doc = nlp(text)
sentences = list(doc.sents)
print(sentences)
```

Pre-processing

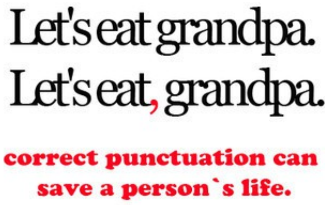
- As mentioned, an important part of the "art" of text analysis is deciding what data to throw out.
  - Uninformative data add noise and reduce precision of resulting estimates.
  - They are also computationally costly.
- In addition, pre-processing choices can affect down-stream results (as documented in Denny and Spirling 2017).

Capitalization

God or god?

```
text_lower = text.lower() # go to lower-case
```

Punctuation




```
from string import punctuation
translator = str.maketrans('','',punctuation)
text_nopunc = text_lower.translate(translator)
```

Tokens

- After removing punctuation, getting tokens is as simple as splitting on white space.

```
tokens = text_nopunc.split() # splits on spaces
```

Numbers

1871

1949

1990

```
# remove numbers (keep if not a digit)
nonumbers = [t for t in tokens if not t.isdigit()]
# keep if not a digit, else replace with "#"
norm_numbers = [t if not t.isdigit() else '#'
                                    for t in tokens ]
```

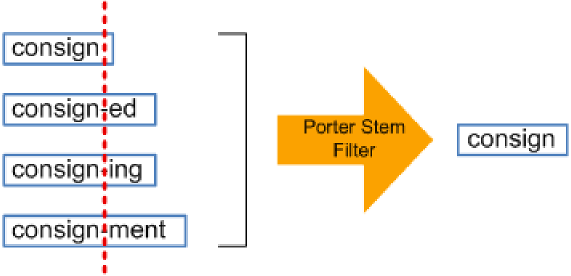
Stopwords

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

```python
from nltk.corpus import stopwords
stoplist = stopwords.words('english')
# keep if not a stopword
nostop = [t for t in tokens if t not in stoplist]
```

- But legal "memes" often contain stopwords:
  - "beyond a reasonable doubt"
  - "with all deliberate speed"
- An alternative is to filter out words and phrases using part-of-speech tags (tomorrow).

Stemming




```
from nltk.stem import SnowballStemmer
stemmer = SnowballStemmer('german')
print(stemmer.stem("Autobahnen"))
stemmer = SnowballStemmer('english')
# remake list of tokens with stemmed versions
tokens_stemmed = [stemmer.stem(t) for t in tokens]
print(tokens_stemmed)
```

Corpus length statistics

- Our raw document strings have now been transformed to a list of sentences, and a list of tokens.

```
num_sentences = len(sentences)
num_words = len(tokens)
words_per_sent = num_words / num_sentences
```

Bag-of-words representation

- Recall the goal of this lecture:
  - Convert a corpus D to a matrix X
- In the “bag-of-words” representation, a row of X is just the frequency distribution over words in the document corresponding to that row.

```
from collections import Counter
freqs = Counter(tokens)
freqs.most_common()
```

Measuring Judicial Output using Decisions Texts

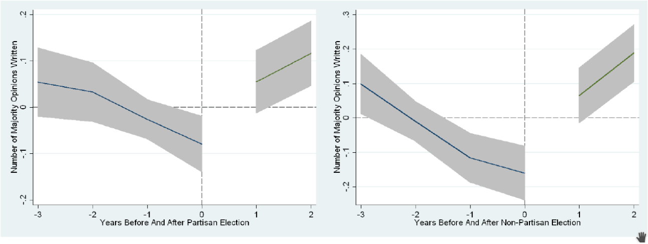
- The number of documents, and the length of those documents, already provides an interesting set of variables for analysis.
- For example:
  - How do electoral incentives affect judging effort?
  - How does the biological aging process affect effort and writing style?
- Appellate judges spend most of their time working on judicial opinions, so the combined length of those opinions provides some rough idea of how much work they are doing year-to-year.

Empirical Setting

- The setting for Ash and MacLeod (2015, 2016, 2017):
  - State supreme courts: the highest appellate court for each of the 50 states in the USA.
  - Data set has 1.1 million judicial opinions for 1947-1994
- States are nice place to look at natural experiments:
  - Unlike most jurisdictions, state judges are often elected, and the rules for election change over time.

Elections Reduce Number of Opinions Written

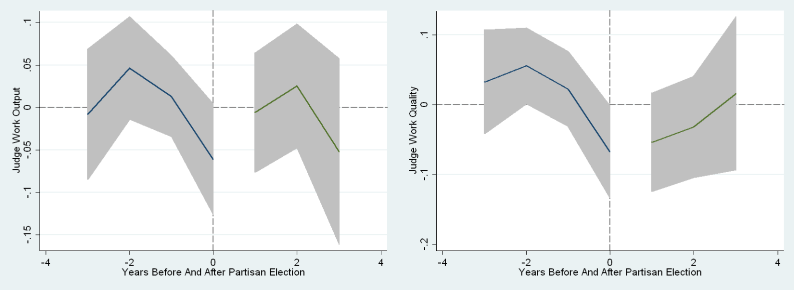
- Left panel: Partisan Elections, Right panel: Non-Partisan Elections



Fractional-polynomial prediction plots with y = outcomes and x = years before and after election year; outcomes residualized on judge and year fixed effects and standardized by judge; gray bars give 95% confidence intervals.

Effect of Partisan Election

- Respectively, effect on output (words written) and quality (citations per opinion):

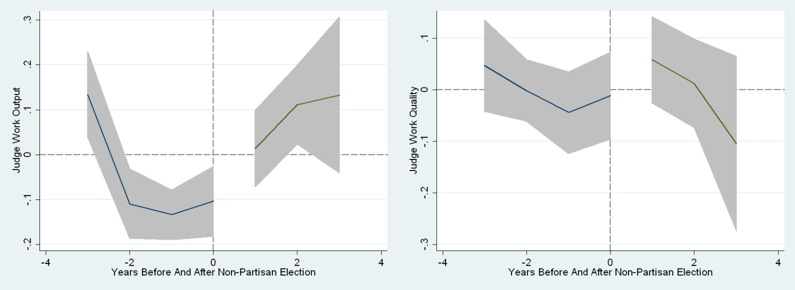


Fractional-polynomial prediction plots with y = outcomes and x = years before and after election year; outcomes residualized on judge and year fixed effects and standardized by judge; gray bars give 95% confidence intervals.

- Partisan Elections have negative effects on output and quality, but barely significant.

Effect of Non-Partisan Election

- Respectively, effect on output (words written) and quality (citations per opinion):

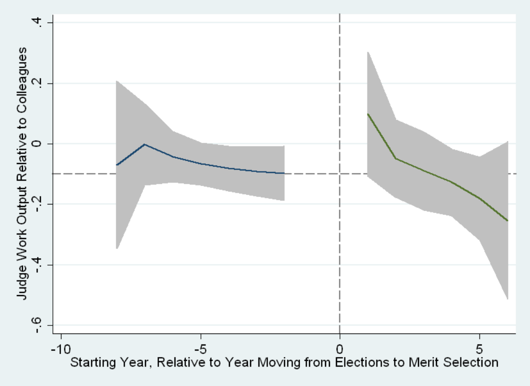


Fractional-polynomial prediction plots with y = outcomes and x = years before and after election year; outcomes residualized on judge and year fixed effects and standardized by judge; gray bars give 95% confidence intervals.

- Non-Partisan Elections have big negative effect on output, but not quality.
  - Consistent with motivation to reduce quantity and maintain quality.

Effect of Merit-Selection Reform on Work Output

- Judge work output, residualized on state-year fixed effects, plotted by starting year, relative to merit reform:

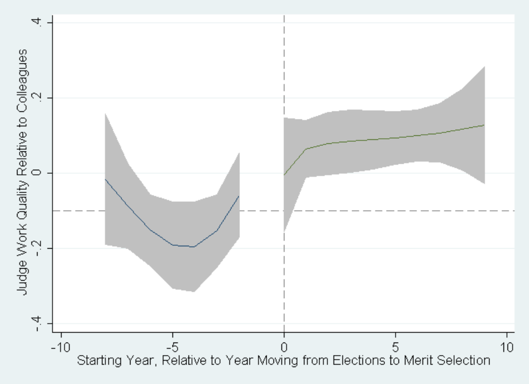


Fractional-polynomial prediction plots with y = judge output and x = judge starting year - reform year; outcomes residualized on state×year fixed effects and standardized by state×year; gray bars give 95% confidence intervals.

- Merit judges write about the same amount as elected judges.

Effect of Merit-Selection Reform on Work Quality

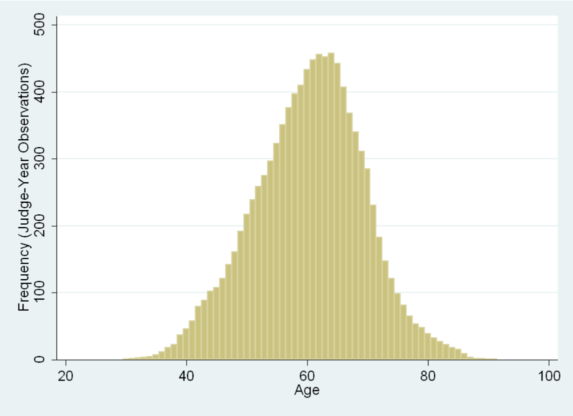
- Quality of judges, residualized on state-year fixed effects, plotted by starting year, relative to merit reform:



Fractional-polynomial prediction plots with y = judge quality and x = judge starting year - reform year; outcomes residualized on state×year fixed effects and standardized by state×year; gray bars give 95% confidence intervals.

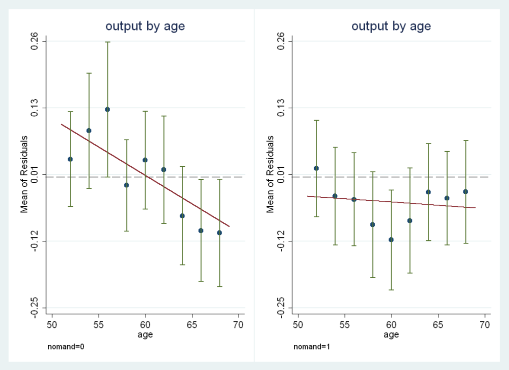
- Judges selected after the reform write higher-quality decisions than judges selected before the reform.

Judge Age Distribution



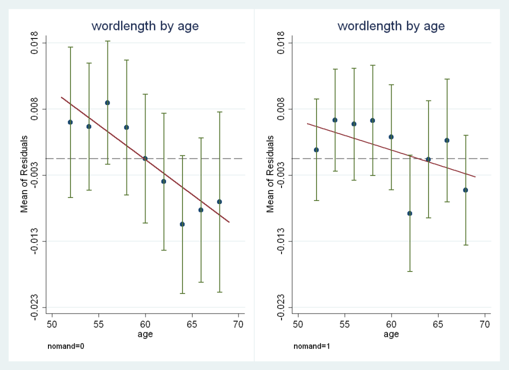
- State supreme court judges have a wide age range but all do the same work task.

Judge Age and Output



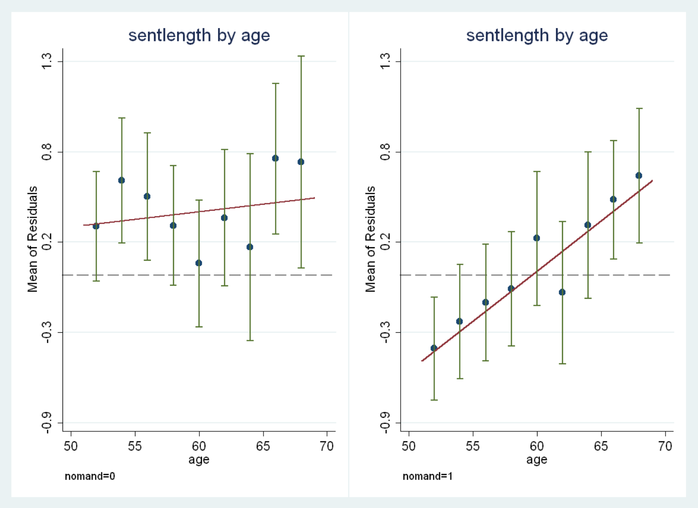
- Judge output decreases with age, but only under mandatory retirement (left panel).
  - Consistent with an incentive rather than physiological effect on productivity.

Judge Word Length and Age



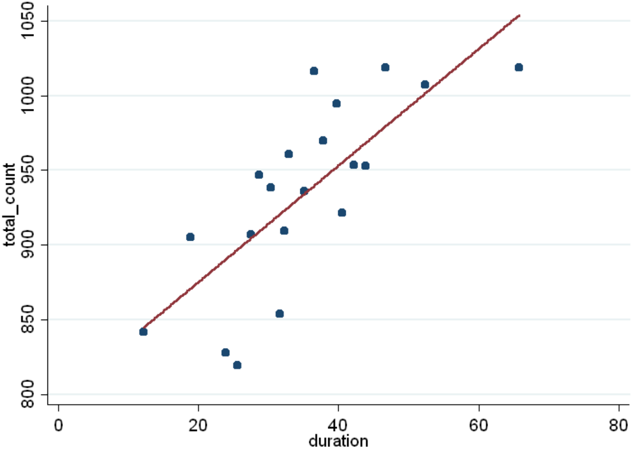
- Older judges use shorter words (fewer characters per word).

Judge Sentence Length and Age



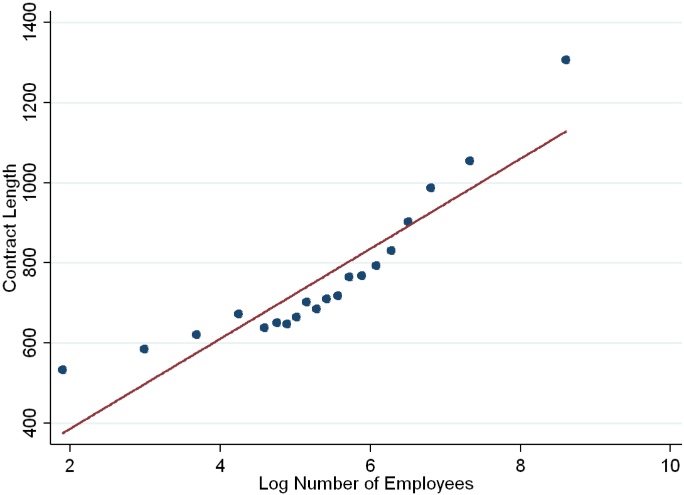
- Older judges use longer sentences (words per sentence)
  - Mandatory retirement incentives (left panel) weakens effect.

Longer-Duration Labor Union Contracts are More Detailed



- Vertical axis: number of clauses in contract.
- Horizontal axis: Duration of contract (in months)
- Source: Ash, MacLeod, and Naidu (2017)

Union Contract Length vs. Log Number of Employees



- Source: Ash, MacLeod, and Naidu (2017)

Up next

- We're already done with the basics of representing text as data.
- Next we will turn to the basics of machine learning.
- Tomorrow, we'll come back to richer text representation.