Digging Deeper, Reaching Further

Module 4.2: Performing Text Analysis

Basic Approaches with Python

In this module we'll…

- Review text analysis strategies for advanced researchers

  → *Make skill-appropriate recommendations*

- Explore text analysis methods in-depth

  → *Understand the kinds of research available in the field*

- Use an HTRC dataset to conduct exploratory data analysis

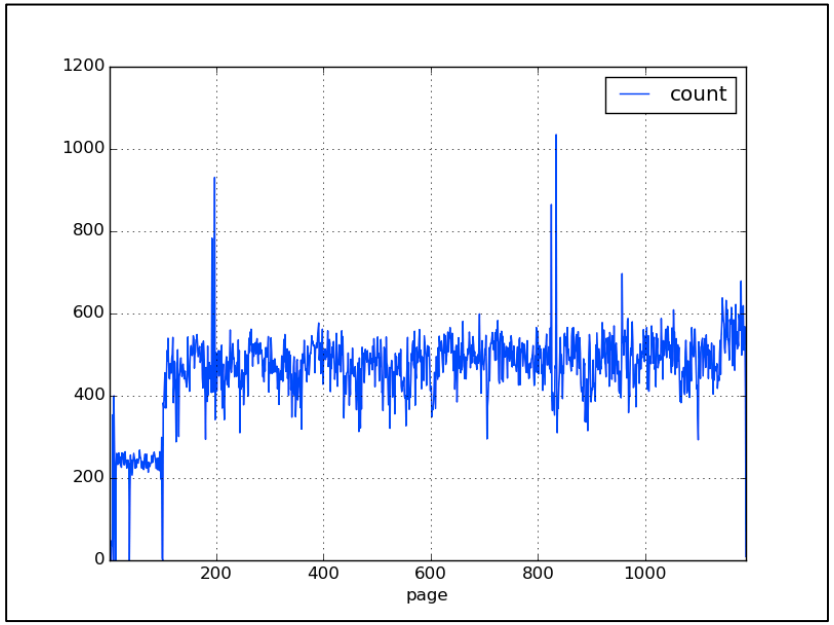
  → *Practice programming skills for data-driven research*

- See how Sam analyzed his *Creativity Corpus*

  → *Learn how a researcher used text analysis methods*

Where we'll end up

```
13:45 ~ $ python top_adjectives.py 1970
                count
token
_               21248
other           17824
own             11552
new             10400
good             9984
great            9440
American         7936
many             7840
major            6480
last             6288
public           6224
important        6096
first            5808
such             5760
economic         5504
human            5424
international    5216
national         4864
same             4688
next             3920
nuclear          3840
local            3744
foreign          3696
political        3648
comprehensive    3488
few              3472
sure             3440
possible         3408
```



Use a Python library to create a list of top adjectives in a volume and a graph that visualizes word count across a volume.

Key approaches to text analysis

Broad Area: Natural Language Processing (NLP)

Using computers to understand the meaning, relationships, and semantics within human-language text

- **Specific Methods:**
  - **Named entity extraction:** what names of people, places, and organizations are in the text?
  - **Sentiment analysis**: what emotions are present in the text?
  - **Stylometry**: what can we learn from measuring features of style?

Key approaches to text analysis

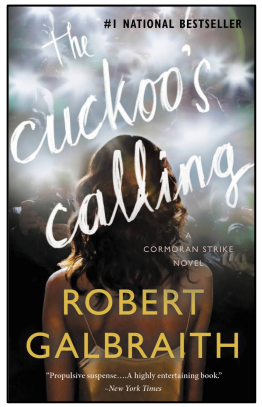
Broad Area: Machine Learning

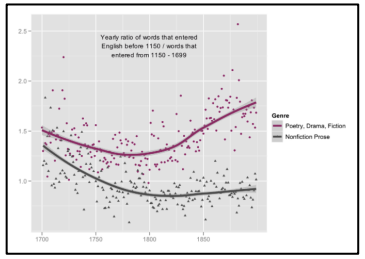
Training computers to recognize patterns.

- **Specific Methods**
  - **Topic modeling** – What thematic topics are present in the text?
  - **Naïve Bayes classification –** Which of the categories that I have named does the text belong to?

Activity: Identify the method

☞ *See Handout p. 1*



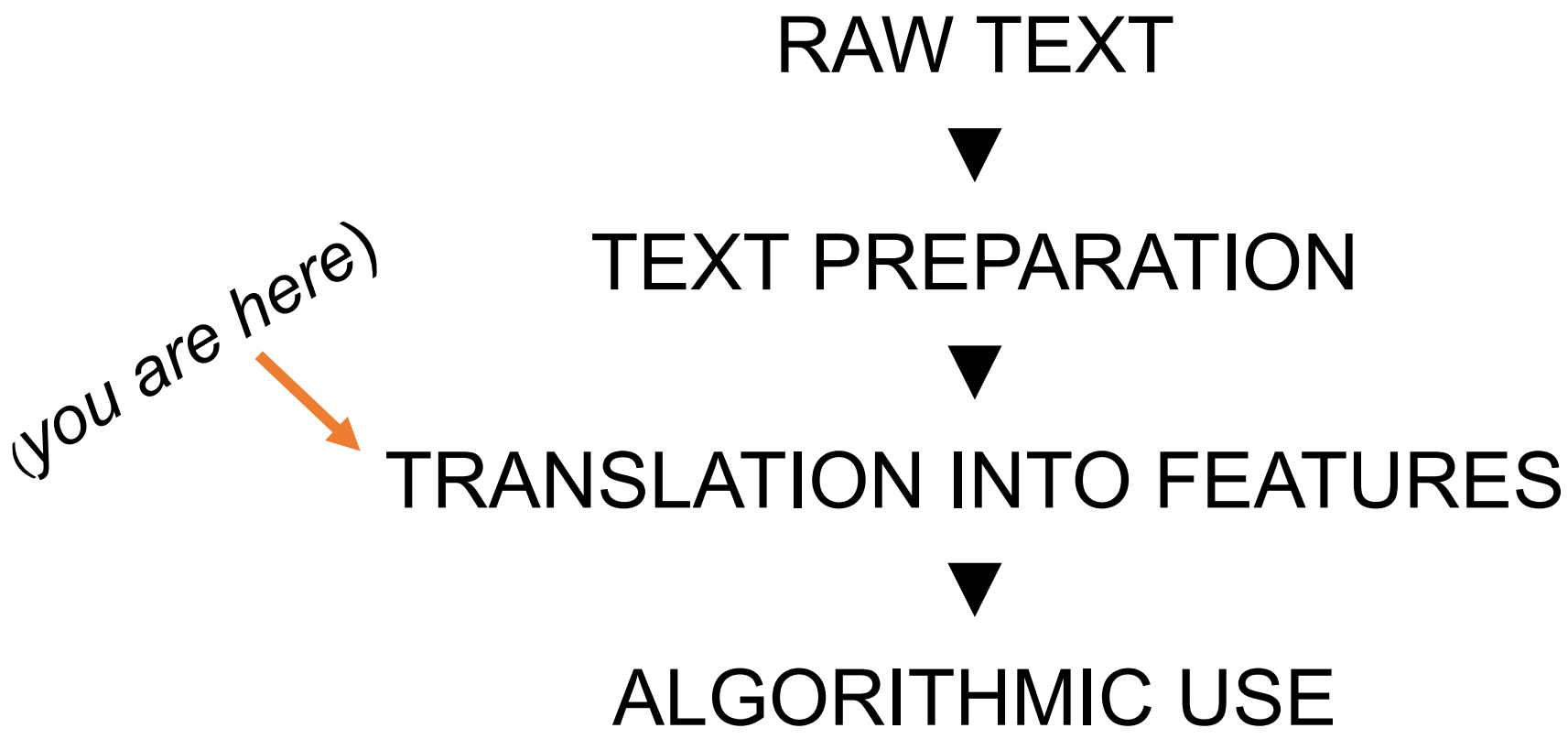


	Broad area	Specific method
'Rowling and "Galbraith"': an authorial analysis		
Significant Themes in 19th Century Literature		
The Emergence of Literary Diction		

Note:
Broad areas/specific methods are those defined in the previous two slides

Link to project summaries: http://go.illinois.edu/ddrf-research-examples

Text analysis workflow



Features in the HTRC

- HTRC Extracted Features dataset
- Downloadable
- Structured data consisting of features
- 5 billion pages, in 13.6 million volumes

https://analytics.hathitrust.org/datasets#ef

HTRC Extracted Features (EF)

- The features are
  - Selected data and metadata
  - Extracted from raw text
- Position the researcher to begin analysis
  - Some of the preprocessing is already done
- Form of non-consumptive access

Per-volume features

- Pulled from bibliographic metadata
- Title
- Author
- Language
- Identifiers

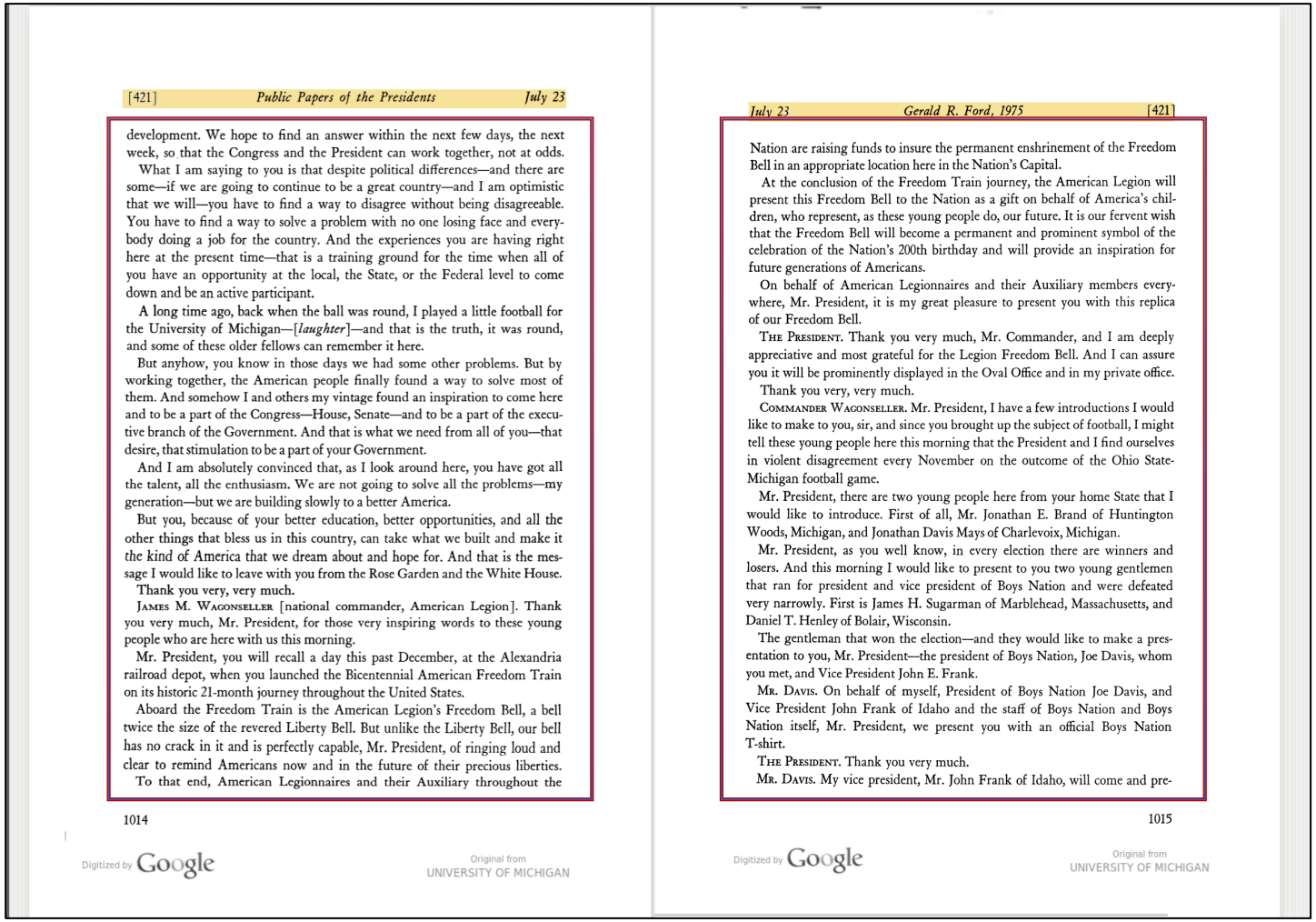

```
{
   "id":"uc1.b3419888",
   "metadata":{
      "schemaVersion":"1.2",
      "dateCreated":"2015-02-12T13:30",
      "title":"Zoonomia = or The laws of organic life / by Erasmus Darwin.",
      "pubDate":"1809",
      "language":"eng",
      "htBibUrl":"http://catalog.hathitrust.org/api/volumes/full/htid/uc1.b3419888.json",
      "handleUrl":"http://hdl.handle.net/2027/uc1.b3419888",
      "oclc":"3679915",
      "imprint":"Thomas and Andrews, 1809."
   },
   "features":{
      "schemaVersion":"2.0",
      "dateCreated":"2015-02-20T23:58",
      "pageCount":616,
      "pages":[
```

Per-page features

- Page sequence
- Computationally-inferred metadata
  - Word, line, and sentence counts
  - Empty line count
  - Language


```
{
    "seq":"00000035",
    "tokenCount":507,
    "lineCount":44,
    "emptyLineCount":0,
    "sentenceCount":14,
    "languages":[
    {
        "en":"1.00"}],
```

Page section features



[421] *Public Papers of the Presidents* *July 23*

development. We hope to find an answer within the next few days, the next week, so that the Congress and the President can work together, not at odds.

What I am saying to you is that despite political differences—and there are some—if we are going to continue to be a great country—and I am optimistic that we will—you have to find a way to disagree without being disagreeable. You have to find a way to solve a problem with no one losing face and everybody doing a job for the country. And the experiences you are having right here at the present time—that is a training ground for the time when all of you have an opportunity at the local, the State, or the Federal level to come down and be an active participant.

A long time ago, back when the ball was round, I played a little football for the University of Michigan—[*laughter*]—and that is the truth, it was round, and some of these older fellows can remember it here.

But anyhow, you know in those days we had some other problems. But by working together, the American people finally found a way to solve most of them. And somehow I and others my vintage found an inspiration to come here and to be a part of the Congress—House, Senate—and to be a part of the executive branch of the Government. And that is what we need from all of you—that desire, that stimulation to be a part of your Government.

And I am absolutely convinced that, as I look around here, you have got all the talent, all the enthusiasm. We are not going to solve all the problems—my generation—but we are building slowly to a better America.

But you, because of your better education, better opportunities, and all the other things that bless us in this country, can take what we built and make it the kind of America that we dream about and hope for. And that is the message I would like to leave with you from the Rose Garden and the White House.

Thank you very, very much.

JAMES M. WAGONSELLER [national commander, American Legion]. Thank you very much, Mr. President, for those very inspiring words to these young people who are here with us this morning.

Mr. President, you will recall a day this past December, at the Alexandria railroad depot, when you launched the Bicentennial American Freedom Train on its historic 21-month journey throughout the United States.

Aboard the Freedom Train is the American Legion's Freedom Bell, a bell twice the size of the revered Liberty Bell. But unlike the Liberty Bell, our bell has no crack in it and is perfectly capable, Mr. President, of ringing loud and clear to remind Americans now and in the future of their precious liberties.

To that end, American Legionnaires and their Auxiliary throughout the

1014

July 23 *Gerald R. Ford, 1975* [421]

Nation are raising funds to insure the permanent enshrinement of the Freedom Bell in an appropriate location here in the Nation's Capital.

At the conclusion of the Freedom Train journey, the American Legion will present this Freedom Bell to the Nation as a gift on behalf of America's children, who represent, as these young people do, our future. It is our fervent wish that the Freedom Bell will become a permanent and prominent symbol of the celebration of the Nation's 200th birthday and will provide an inspiration for future generations of Americans.

On behalf of American Legionnaires and their Auxiliary members everywhere, Mr. President, it is my great pleasure to present you with this replica of our Freedom Bell.

THE PRESIDENT. Thank you very much, Mr. Commander, and I am deeply appreciative and most grateful for the Legion Freedom Bell. And I can assure you it will be prominently displayed in the Oval Office and in my private office.

Thank you very, very much.

COMMANDER WAGONSELLER. Mr. President, I have a few introductions I would like to make to you, sir, and since you brought up the subject of football, I might tell these young people here this morning that the President and I find ourselves in violent disagreement every November on the outcome of the Ohio State-Michigan football game.

Mr. President, there are two young people here from your home State that I would like to introduce. First of all, Mr. Jonathan E. Brand of Huntington Woods, Michigan, and Jonathan Davis Mays of Charlevoix, Michigan.

Mr. President, as you well know, in every election there are winners and losers. And this morning I would like to present to you two young gentlemen that ran for president and vice president of Boys Nation and were defeated very narrowly. First is James H. Sugarman of Marblehead, Massachusetts, and Daniel T. Henley of Bolair, Wisconsin.

The gentleman that won the election—and they would like to make a presentation to you, Mr. President—the president of Boys Nation, Joe Davis, whom you met, and Vice President John E. Frank.

MR. DAVIS. On behalf of myself, President of Boys Nation Joe Davis, and Vice President John Frank of Idaho and the staff of Boys Nation and Boys Nation itself, Mr. President, we present you with an official Boys Nation T-shirt.

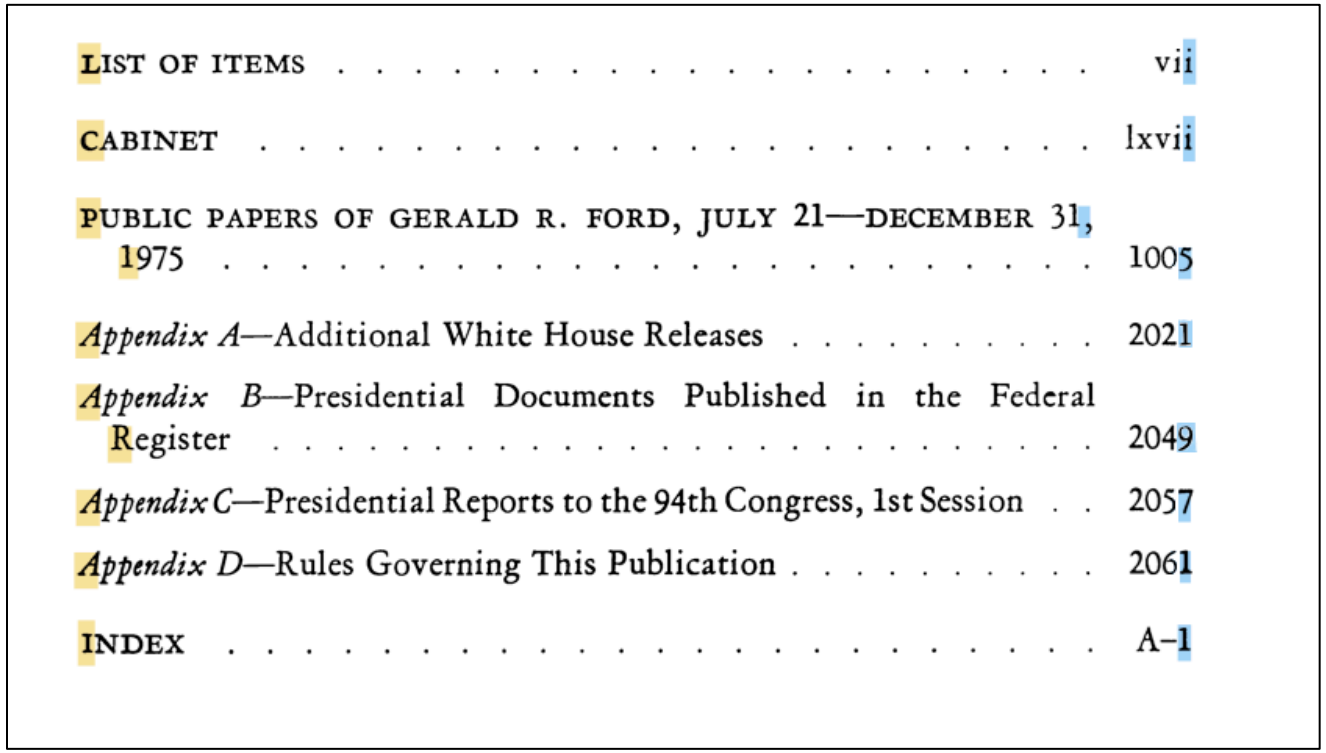
THE PRESIDENT. Thank you very much.

MR. DAVIS. My vice president, Mr. John Frank of Idaho, will come and pre-

1015

Public papers of the presidents of the United States (Gerald R. Ford, book 2)

Page section features



Public papers of the presidents of the United States (Gerald R. Ford, book 2)

Page section features

Header, body, footer

- Line, empty line, and sentence count
- Counts of beginning- and end-line characters
- Token counts
  - Homonyms counted separately
  - Part-of-speech codes are from the Penn Tree Bank

```
40 "body":{
41     "tokenCount":504,
42     "lineCount":43,”
43     "emptyLineCount":0,”
44     "sentenceCount":12,
45     "tokenPosCount":{
46 "fynthefis":{"NNP":1},
47 "Laws":{"NNP":1},
48 "beautiful":{"JJ":1},
49 "philofopher":{"NN":1},
50                  "uponthe":{"IN":1},
51 "for":{"IN":1},
```

Using HTRC Extracted Features

- Identify parts of a book
  - From descriptive metadata
- Perform any method that works with bags-of-words
  - Topic modeling
  - Dunning's log-likelihood
- Classify volumes
  - Compare with bibliographic metadata

Do-it-yourself text analysis

- Some researchers won't use off-the-shelf tools
  - Want more control over processes
  - Set their own parameters
- Mix-and-match approaches to create "toolkit" of strategies

The toolkit

- Researcher-dependent
- Requires understanding of statistics
- Often draws on expert collaborators
- Consists of command line tools and programming languages

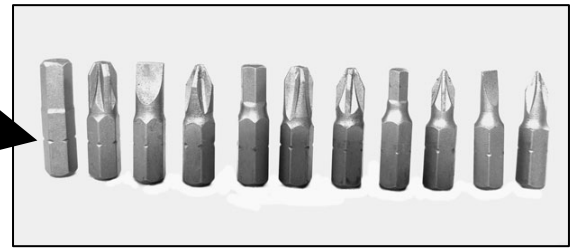
Command-line tools

- Tools you download and run from the command line
- MALLET (Java-based)
  - Topic modeling and classification
- Stanford NLP (Java-based)
  - Natural language processing

Code libraries

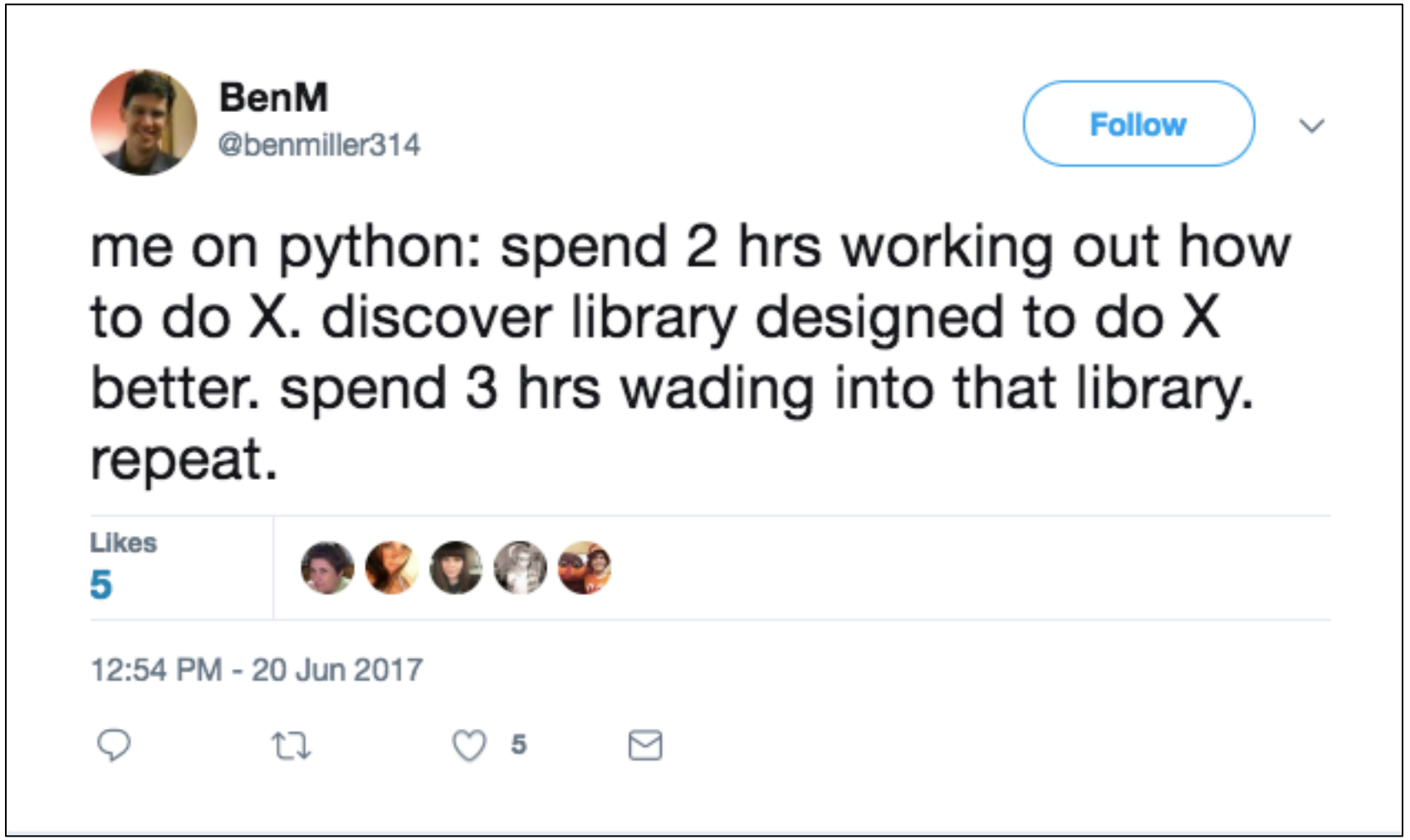
- Part of programming is re-using code, often shared as ***libraries***
- The existing code is installed via ***packages***
- Packages are collections of bits of code, made up of ***modules***
- Modules facilitate programming tasks

Package



Modules

Using libraries in practice



Installing packages for Python

- Install using a package manager
- Example: pip
  - Package manager for Python
  - Generally comes with Python install
- Basic syntax
  - pip install <name>
- Other installers: Homebrew, Conda

Using a library's module in a script

- Once library is installed…
- Must declare that it will be used at the top of the script
  - "Import" the module
- Use the desired module in the body of the script

/ home / elfdickson / get_verbs.py

Keyboard shortcuts: Normal

```
1  from htrc_features import FeatureReader
2  import pandas as pd
3  import sys
4
5
6  # if __name__ == '__main__':
7  #     if len(sys.argv) < 2:
8  #         print("Missing filename argument")
9  #         sys.exit(1)
10
11
12 filename = sys.argv[1]
13 idx = pd.IndexSlice
14
15 def get_proper_nouns(vol):
16     tl = vol.tokenlist(pages=False)
17     tl.index = tl.index.droplevel(0)
18 #    tl['date'] = vol.year
19 #    tl = tl.set_index('date', append=True).reorder_levels(['date', 'token', 'pos'])
20 #    tl = tl.set_index#.reorder_levels(['token', 'pos'])
21     try:
22         proper_nouns = tl.loc[idx[:,:,('VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ')],]
23         proper_nouns.index = proper_nouns.index.droplevel(2)
24         return proper_nouns[proper_nouns['count'] < 150]
25     except:
26         return pd.DataFrame()
```

Imports modules, i.e. the components of the package

Use the module

Python text analysis libraries

- Machine learning
  - SciKit Learn
- Data science
  - Pandas
- Natural language processing
  - Natural Language Toolkit (NLTK)

- Example: `nltk.words_tokenize()`
  - *Can you guess what this does?*

HTRC Feature Reader Python library

- Python library for working with HTRC Extracted Features
  - Bits of pre-written code to parse the JSON
- Install using a package manager, like pip
  - Source code lives on Github
- Also need to have Pandas installed
  - Remember: pandas = Python library for working with data
  - Luckily it's included in PythonAnywhere!

Sample Reference Question

I’m a student in history who would like to incorporate digital methods into my research. I study American politics, and in particular I’d like to examine how concepts such as liberty change over time.

Approach: view adjectives in Extracted Features file

Hands-on activity

☞ *See Handout pp. 1-2*

In this activity, you will install the Feature Reader library and run a Python script to create a list of the most-used adjectives and the number of times they occur in different sets of volumes of presidential papers (using the Extracted Features files of these volumes).

What You Need:

Script: top_adjectives.py

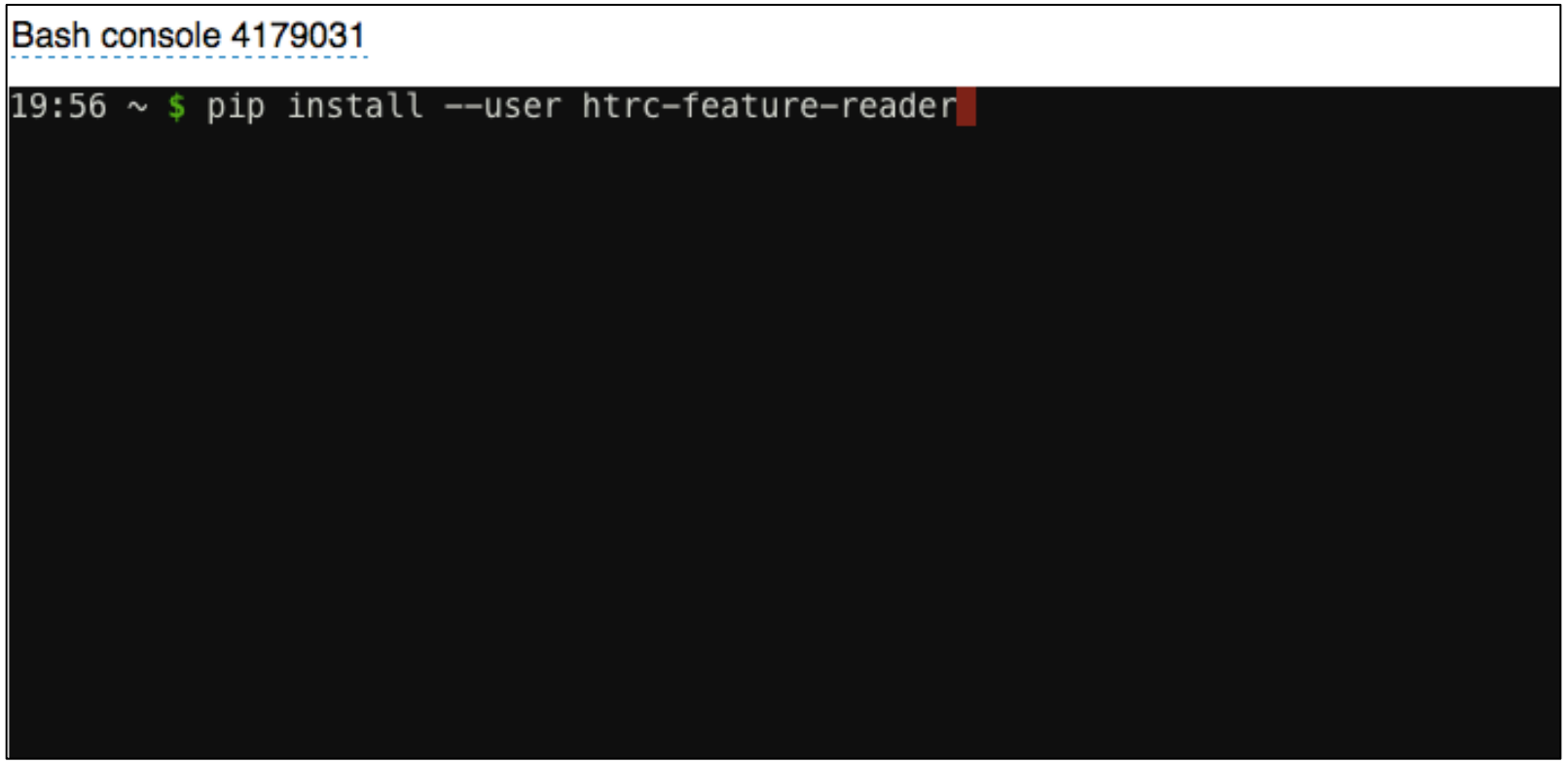
Files for analysis, in 2 directories:

- 1970
- 1930

Files and directories

- 1930 and 1970 directories and required files should already be in your files in PythonAnywhere
- Check using the “Browse files” button
- 1970/ = Extracted Features files of presidential speeches from the 1970s (corresponds to our political science workset!)
- 1930/ = Extracted Features files of presidential speeches from the 1930s

Install Feature Reader library




```
pip install --user htrc-feature-reader
```

Examine code

```
1  from htrc_features import FeatureReader
2  import glob
3  import pandas as pd
4  import sys
5
6  decade = sys.argv[1]
7
8  paths = glob.glob(decade+'/*.json.bz2')
9  idx = pd.IndexSlice
10 fr = FeatureReader(paths)
11 vol = next(fr.volumes())
12 tl = vol.tokenlist(pages=False)
13 tl.index = tl.index.droplevel(0)
14 adjectives = tl.loc[idx[:,('JJ')],]
15 adj_dfs = [adjectives for vol in fr.volumes()]
16 all_adj = pd.concat(adj_dfs).groupby(level='token').sum().sort('count', ascending=False)[:50]
17
18 print(all_adj)
```

Prepares us to use the libraries needed for this task.

'JJ' is the code for adjectives from the Penn TreeBank.

Open shell and run command


```
python top_adjectives.py 1970
```

View results

```
13:45 ~ $ python top_adjectives.py 1970
              count
token
_             21248
other         17824
own           11552
new           10400
good           9984
great          9440
American       7936
many           7840
major          6480
last           6288
public         6224
important      6096
first          5808
such           5760
economic       5504
human          5424
international  5216
national       4864
same           4688
next           3920
nuclear        3840
local          3744
foreign        3696
political      3648
comprehensive  3488
few            3472
sure           3440
possible       3408
```

On your own

☞ *See Handout p. 2*

- **Compare the adjectives used by presidents in the 1970s with those used in the 1930s.**
- **Work and discuss with your neighbor.**

- *How do you need to change your command?*
- *What differences do you see? Similarities?*

Challenge

Try modifying the script to search verbs or another part of speech.

(Hint: Don't forget the Penn Tree Bank!)

Exploratory data analysis

- Approach for getting familiar with data
- Easier to recognize patterns (and problems)
  - Hard to see trends in a spreadsheet or text file!
- There are whole books about it
- Strategies:
  - Plot raw data
  - Plot simple statistics
  - Compare plots to look for patterns

Visualization libraries

- **Python**
  - *Matplotlib* library, *pyplot()* function
  - *ggplot* library
- **Others:**
  - R: ggplot2
  - D3.js: JavaScript library for visualizations

Sample Reference Question

I’m a student in history who would like to incorporate digital methods into my research. I study American politics, and in particular I’d like to examine how concepts such as liberty change over time.

Approach:

View word count over a volume using an Extracted Features file

Hands-on activity

☞ *See Handout p. 3*

- In this activity, you will create a visualization using the the native plotting functionality in Pandas, called pyplot, to see the word count over a volume based on its Extracted Features file

Script: word_count.py

Files for analysis: mdp.49015002203033.json.bz2 (*Presidential Papers of the Presidents of the United States: Gerald Ford*)

Examine code

```
1  from htrc_features import FeatureReader
2  #import os
3  import glob
4  #import pandas as pd
5  import matplotlib
6  import matplotlib.pyplot as plt
7  matplotlib.use("Agg")
8
9
10 path = glob.glob('1970/mdp.49015002203033.json.bz2')
11 fr = FeatureReader(path)
12 vol = fr.first()
13
14 tokens = vol.tokens_per_page()
15
16 tokens.plot()
17 plt.savefig("words.png")
18
19
```

Imports our modules so we can call functions later, including matplotlib's pyplot

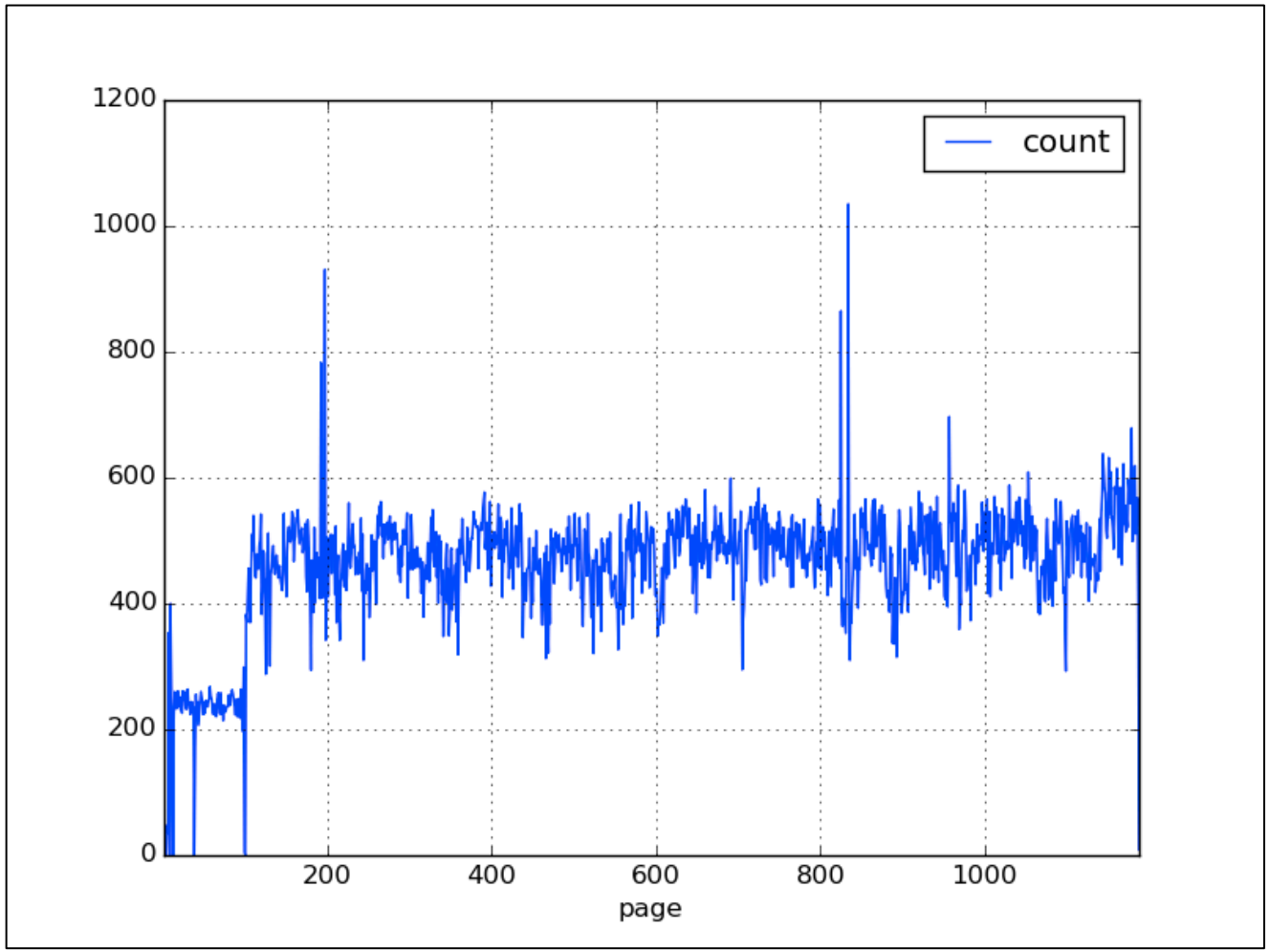
Set variable "tokens"

Plot tokens and save plot as png

Open shell and run command

python word_count.py

View graph



words.png

The graph visualizes how many words appear on each page in the volume.

Page numbers are on the horizontal axis, and the word count is on the vertical axis.

E.g. The 600th page has about 350 words.

On your own

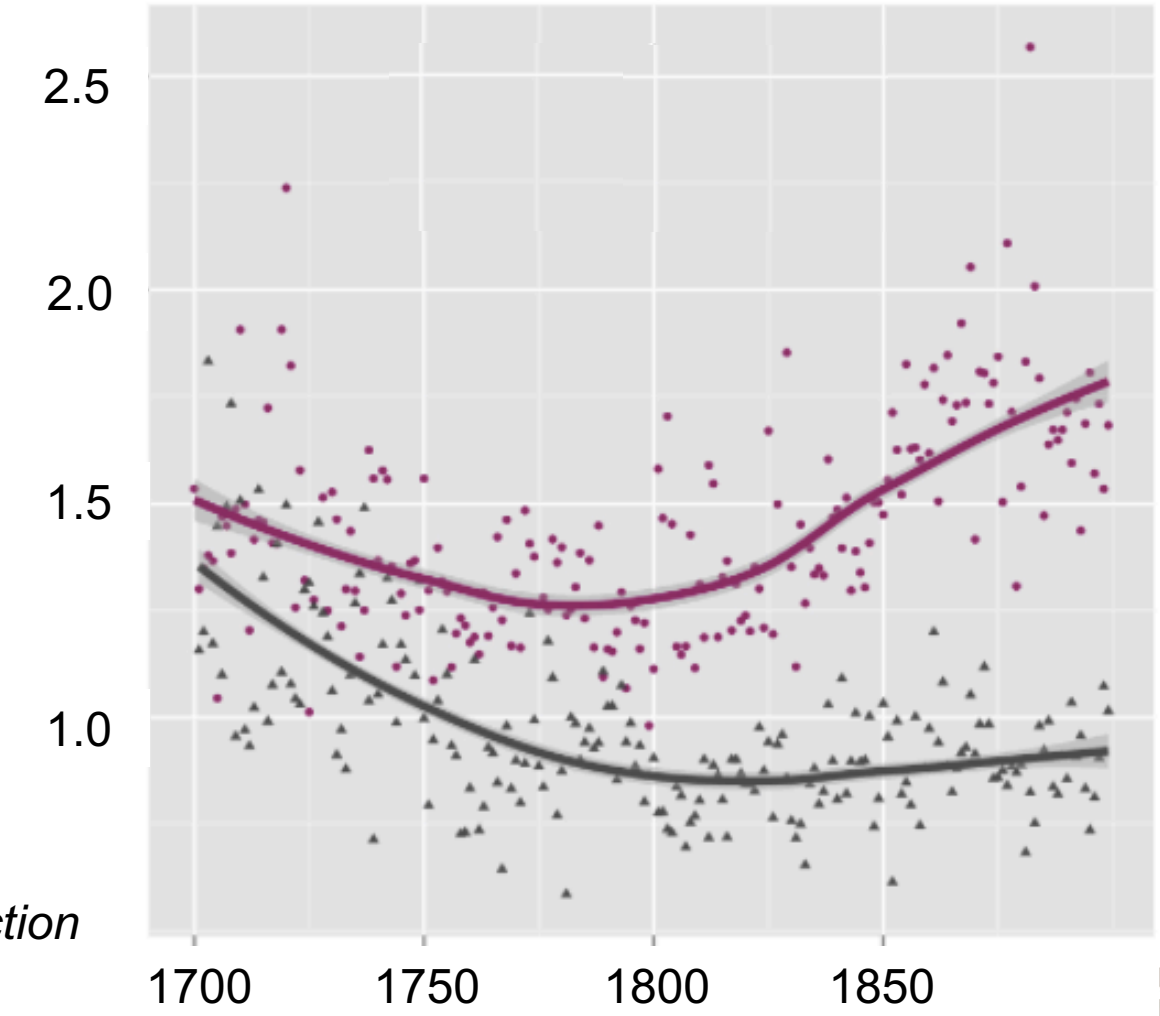
☞ *See Handout p. 3*

- Can you modify the script to look at another volume?
- Edit, save, and run the file
  - Hint: Change the path to another volume in the script
- View and discuss with your neighbor

Another look

From paper - graph of diction patterns between genres, using frequency counts (Underwood and Sellers, 2012)

- Even basic counts can lead to sophisticated research
- Ted used HTRC EF
- Classified genre at the page level



Case Study: *Inside the Creativity Boom*

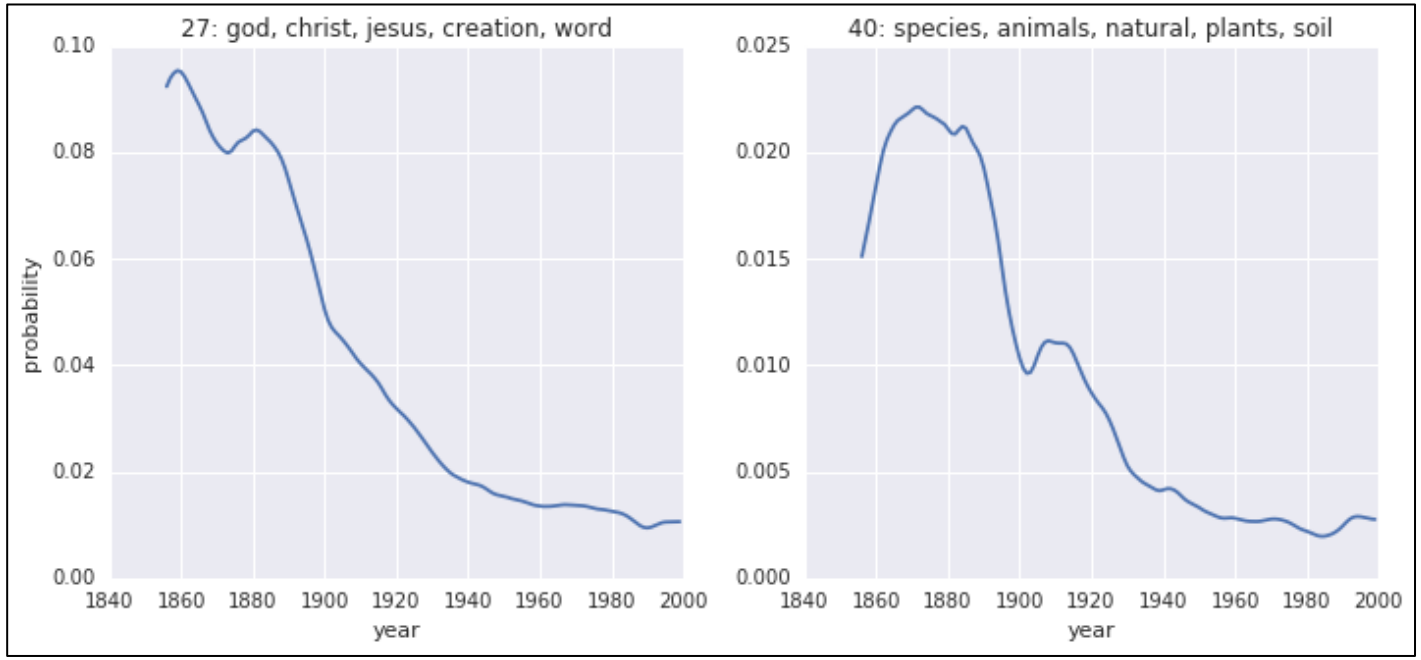
After reducing Creativity Corpus to pages containing forms of creativ*:

- Performed topic modeling on those pages
- Ended up with topics that reflect what themes are prevalent around concept of “creativity” in the 20th century
- Graphed the topics over time to see how their usage changed

Case Study: *Inside the Creativity Boom*

- Topics that decreased in usage over time
  - god, christ, jesus, creation, word
  - species, animals, natural, plants, soil
  - nature, mind, creative, world, human
  - invention, power, creative, own, ideas

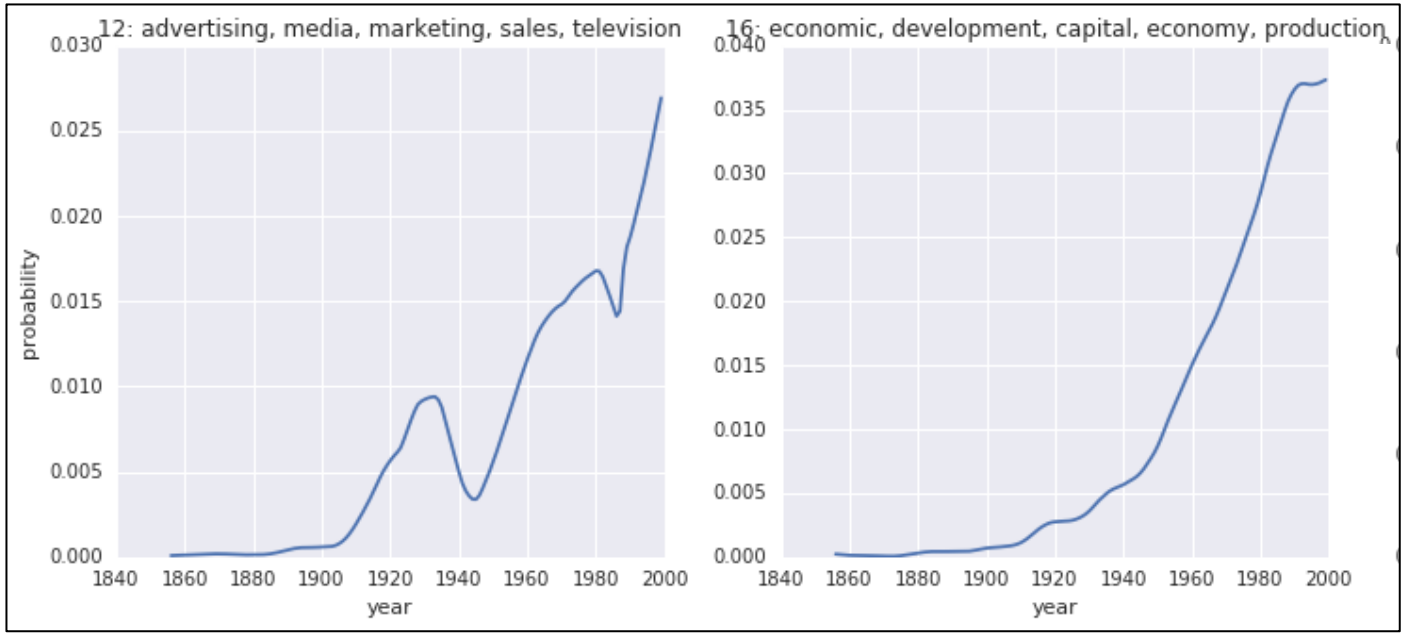
Creativity topics with falling usage



Case Study: *Inside the Creativity Boom*

- Topics that increased in usage over time
  - advertising, media, marketing, sales, television
  - economic, development, capital, economy, production
  - poetry, language, poet, poets, poems
  - social, creative, study, development, behavior

Creativity topics with increasing usage



Discussion

- *In what ways can librarians support advanced text analysis research?*
- *What additional skills would you need to learn in order to do so?*

Questions?

References

- Underwood, T., & Sellers, J. (2012). The emergence of literary diction. *Journal of Digital Humanities*, 1(2), 1-2. http://journalofdigitalhumanities.org/1-2/the-emergence-of-literary-diction-by-ted-underwood-and-jordan-sellers/ .