WE1S “counting” module

Included in the WE1S Workspace (see S-2), the “counting” module contains three Jupyter notebooks for finding the frequency of documents, tokens, and phrases (ngrams) in a collection of texts referred to below as a user’s “project.”

`count_documents.ipynb`. This notebook counts the number of documents per unique source per year in a project. It offers two different methods of counting. It also includes a few options for saving and visualizing count totals.

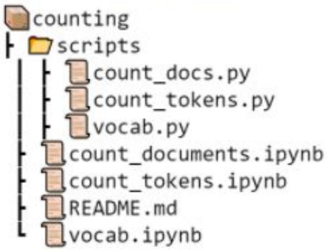
`count_tokens.ipynb`. This notebook provides a series of functions for doing the following with a project:

- counting the number of documents containing a specific word or phrase;
- obtaining uni-, bi-, and trigram frequency counts and relative frequencies;
- obtaining tf-idf scores for specific words;
- and for utilizing some basic collocation metrics for determining the relationships between words.

Some examples of questions that a user can explore:

How many documents in a project contain the phrase *first-generation*? What is the frequency of the word *college* in a specific document, and/or across all of a project's documents? What are some of the most distinctive words for particular documents in a project? What are some of the most important words for understanding the content of documents in the project overall?

Module structure

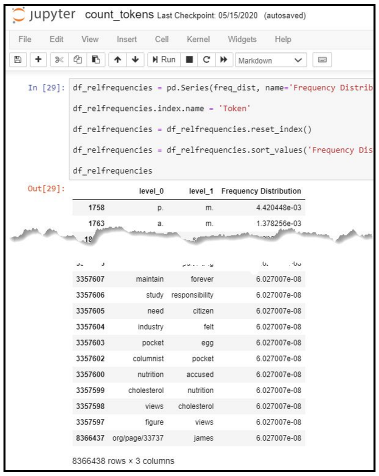


```
counting
├ scripts
│ ├ count_docs.py
│ ├ count_tokens.py
│ ├ vocab.py
├ count_documents.ipynb
├ count_tokens.ipynb
├ README.md
└ vocab.ipynb
```

How strongly associated is the word *private* with the word *college* in a project? What are some meaningful or important collocations (2- or 3-word phrases) in a project, and how can one determine and describe what it means for them to be "meaningful" or "important" in different ways?

`vocab.ipynb`. This notebook builds a single json file (the vocab file) containing term counts for all the documents in a json directory. It provides methods for generating the following data (among others) about a project: a list of filenames; a list of document names, the number of documents, the number of terms or tokens, a dataframe containing the terms and counts in the vocab or a list of documents specified by filenames or name field values.

Dataframe of bigram (2-word phrase) frequencies created in `counts_tokens.ipynb` (larger)



```
In [29]: df_relfrequencies = pd.Series(freq_dist, name='Frequency Distrib
         df_relfrequencies.index.name = 'Token'
         df_relfrequencies = df_relfrequencies.reset_index()
         df_relfrequencies = df_relfrequencies.sort_values('Frequency Dis
         df_relfrequencies
```

Out[29]:

	level_0	level_1	Frequency Distribution
1758	p.	m.	4.420448e-03
1763	a.	m.	1.378256e-03
3357607	maintain	forever	6.027007e-08
3357606	study	responsibility	6.027007e-08
3357605	need	citizen	6.027007e-08
3357604	industry	felt	6.027007e-08
3357603	pocket	egg	6.027007e-08
3357602	columnist	pocket	6.027007e-08
3357600	nutrition	accused	6.027007e-08
3357599	cholesterol	nutrition	6.027007e-08
3357598	views	cholesterol	6.027007e-08
3357597	figure	views	6.027007e-08
8366437	org/page/33737	james	6.027007e-08

8366438 rows × 3 columns

Further Information:

* M-15 (about Jupyter notebooks)

Main Jupyter notebooks in this module:

* count_documents.ipynb, count_tokens.ipynb, vocab.ipynb

Code source: [TBD] (MIT License)

5 Sept. 2020; rev. 13 Sept. 2020 (Lindsay Thomas, Scott Kleinman)