Assignment 1

Text as Data

2022-09-08

Introduction

In this assignment, you are asked to produce analysis that follows a set of instructions. You can do this any way you like, as long as you show me your results and the code you used to get there. The easier this is for me to replicate, and the clearer the code is, the higher your mark will be. One option would be to make a copy of this file, add in code snippets, and submit the RMarkdown file along with the PDF of completed results. Another option would be to send me a link to an .ipynb notebook file on Github.

Getting and parsing texts

To start with, you are asked to retrieve *Songs of Innocence and of Experience* by William Blake from Project Gutenberg. It is located at https://www.gutenberg.org/cache/epub/1934/pg1934.txt. This is a collection of poems in two books: *Songs of Innocence* and *Songs of Experience*.

Parse this into a dataframe where each row is a line of a poem (there should be no empty lines). The following columns should describe where each line was found:

- line_number
- stanza_number
- poem_title
- book_title

Visualising text data

- Create a histogram showing the number of lines per poem
- Create a document feature matrix treating each line as a document
- Create a separate document feature matrix treating each poem as a document
- Using one of these document feature matrices, create a plot that compares the frequency of words in each book. Comment on the features that are more or less frequent in one book than another.

Parsing XML text data

Now we will work with German Parliamentary data, which is available in XML format here for the last two parliamentary periods. Remember XML format is very like HTML format, and we can parse it using a scraper and CSS selectors. Speeches are contained in `<rede>` elements, which each contain a paragraph element describing the speaker, and paragraph elements recording what they said. Not that class selectors won't work, because the class attribute is called "klasse". You can use normal attribute selectors.

Choose one of the sessions, and retrieve it using R or Python. Using a scraper, get a list of all the elements. For each element, get the name of the speaker, and a single string containing everything that they said. Put

this into a dataframe. Print the number of speeches, and the content of the first speech, by a politician of your choice.

Using regular expressions

Using a regular expression, get a list of words spoken in your parliamentary protocol that contain (in upper or lower case) the string "kohle" (coal). Show the number of occurrences of each of these words. If there are no mentions in the debate you have selected, try another protocol.