Great project proposal Yes Holly Warner.

text_complexity: Assessing the complexity of academic journal articles to demonstrate accessibility

The project will be based on the reading complexity framework Flesch Reading Ease readability scores; the higher the score, the easier the text is to read and understand. There are also bands within the scores that correspond to US education levels, ranging from 5th grade (11 years old) to 'Professional' (specialised and college graduates)

Readability Score	US Educational Level	Notes
90-100	5th grade	Very easy to read. Easily understood by an average 11-year-old
80-<90	6th grade	Easy to read. Conversational English for consumers
70-<80	7th grade	Fairly easy to read
60<70	8th and 9th grade	Plain English. Easily understood by 13- to 15-year-old students
50<60	10th to 12th grade	Fairly difficult to read
30-<50	College	Difficult to read
10-<30	College graduate	Very difficult to read. Best understood by university graduates
0-<10	Professional	Extremely difficult to read. Best understood by university graduates

Table adapted from https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests.

The scores are based on the formula:

$$206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right)$$

Formula from https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests.

The aim of the project is to analyse academic articles to understand which level their readability actually corresponds to; whether that be in/near the expected band of College, or more divergent in either direction (more simple or more complicated). To do this, I shall use articles from linguistics as my data. This is because all fields use specialist language, so that shouldn't be too significant. Linguistics may be a problematic choice in terms of the articles containing words used by participants which do not accurately reflect the academic's writing; however, I know I have access to linguistics articles from the Queen Mary Library and the Senate House library. That being said, once the model is working, I can remove any participant quotes from the text, creating a new copy of the text which I can run through the function and compare with the original.

In order to fulfil the formula, I will need to count the number of sentences, words, and syllables present in the text. Sentences can be calculated by counting the occurrences of certain punctuation (e.g. full-stop, question mark, exclamation mark) followed by a space or a new line, remembering that the end of the article will not have a whitespace character. Words will be challenging to define; however, once the text is all lowercase, with numbers and punctuation removed (making an exception for hyphens and full-stops not surrounded by whitespace characters), and then using whitespace characters to split the string into a word list. Syllables, linguistically, contain a nucleus, which in English is often a vowel sound, with the exception of noises such as 'shhh'. For the sake of this project, I will assume all syllables contain a vowel nucleus, so counting syllables would then involve counting the number of vowels present; however, this needs some modifications:

- The vowels need to be surrounded by consonants or whitespace characters on both sides or one consonant and one whitespace character on either side
- Vowels include the letter 'y', such as in 'loudly' because 'y' still makes a vowel sound as the nucleus of a syllable
- As in line with the first bullet point, the vowels must only be single vowels - so excluding consecutive vowels such as 'ee' or 'ou' or 'ey'. However, these double vowels would then not be counted as syllables due to an adjacent vowel. In that case, an appropriate syllable count could look like (`consonant/whitespace + vowel + vowel`) but NOT (`vowel + vowel + consonant/whitespace`), then the first vowel in a chain of consecutive vowels

would be counted, but further vowels in that cluster would not. This should still work for words like 'yacht'.

- Must exclude silent 'e's at the ends of words, such as in 'knife' – exclusion of `'e' + whitespace`

When creating usable text, I will clean the articles manually to ensure headers and footers are removed, along with graphs and tables, and information title pages and reference lists.

There will be multiple functions to handle the counting and calculations shown in the tentative list below:

- `sentence_count` to count the number of sentences
- `word_count` to count the number of words
- `syllable_count` to count the number of syllables
- `flesch_calculation` to calculate the Flesch Reading Ease score based on the counts from the previous functions using the formula shown above. This function will return the readability score as well as the US education bracket and the notes for perspective
- `article_comparison` to bring a visualisation, probably a bar chart, to easily compare the readability of different texts. This would take an unspecified number of parameters and run each through the `flesch_calculation` function to gain their Flesch Reading Ease, and then plot these scores in a bar chart for easy visual comparison and could theoretically be used to compare articles, academic fields, literary genres, or literature through time (or in cases of comparing the same linguistic articles with significant colloquial participant data removed).

Testing will involve running article texts that I have gathered through University libraries and checking the scores I get to ones created by Reading Ease calculators on the internet, such as one from https://readabilityformulas.com/free-readability-formula-tests.php. Given how differently the calculators may define words and syllables I will use multiple calculators and note the mean score. I will then compare the score from my calculator and see if it is near the mean, and within the range of scores from the internet calculators. The function testing will be done using Python IDLE and Jupyter Notebooks, with

the final project presented in a Jupyter Notebook using code and MarkDown. Microsoft Word may be used in the cleaning of the data before being transferred to a text file.

The functions will not be able to assess readability with more nuance by recognising short obscure words as difficult and long well-known words as easy because the Flesch Reading Ease scores also cannot do this; however, it is a good start for quickly understanding how readable a piece of writing could be, and how accessible it is to people within, and outside university education.

Excellent proposal. - I like it.
There is a lot to do in this proposal so do use an iterative development approach and build a simple version quickly so that you can get a better measure of the amount of work everything will take plus that experience will enable you to reflect on which parts of the application are the most useful / troublesome.

Break things down into small units of functionality so that the functions you build are each simple and specialised.