Assignment 4

due 27/2/2020

You always promised yourself that you would read *Ulysses* one day and this is your chance! You can find the entire text of Joyce's tome in `ulysses.docx`. The book is 700 pages long and tough going in places, but fear not!, we can get Python to do most of the heavy lifting and give us a summary (in the guise of an analysis of word frequencies).

Write a Python program named `a4.py` that includes a function `analyze(docfile)` that reads the text contained in the file named `docfile` (presumed to be a .docx file) and calculates the frequency of occurence of each word appearing therein. The frequency of a word is the ratio of the number of appearances of that word and the total number of words in the document e.g. the frequency of the word "the" in `Ulysses` is 0.056 i.e. around every twentieth word is "the".

The output should be written into a workbook containing a single two-column work sheet (words and their frequencies) in decreasing order of frequency. Include only those words with a frequency of 0.001 or greater. The name of the work book file should be derived from that of the file containing the text in the following manner:

```
a_book_name.docx    ==>    a_book_name_word_stats.xlsx
```

i.e by tacking "_word_stats" onto the end of the name and adding an ".xlsx" extension.

1. Use the text from the file `ulysses.docx` testing and use the `docx` package to read through it. The documentation for this package is available here:

   `python-docx.readthedocs.io/en/latest/`

   You may find it useful to create a truncated version of this file (say with just the first chapter) to use in the initial testing.

2. Write the output into a workbook file using the `pyexcel` package. The documentation for this package is available here:

   `docs.pyexcel.org/en/latest/`

   The work book should have a single sheet named "Word Frequency Stats". The work sheet should contain two columns, housing the words and their frequencies respectively. The sheet should not contain any column headings.

3. Ignore case: treat "the" and "The" as the same. Treat words sharing a common stem such as "cat" and "cats" as distinct. Similarly "cats", "cat's" and "cats'" should all be regarded as distinct words. Some spurious "words" (i.e. non-words) may feature in the text. Don't worry about these.

4. Recall that Python's `os.path` package has some useful functions for manipulating file names.

5. The raw text is taken from the Project Gutenberg website (`www.gutenberg.org`) that holds a large collection of out-of-copywright works in various formats. The .docx format used here was derived from the .txt version from that site. Some extraneous matter at the begining and the end has been removed leaving just the text itself.