A string is a sequence of characters that we store as an object. This string can be divided into substrings, where the possible number of substrings of length k (termed k-mers) is the number of possible characters to the k. The linguistic complexity of the string is defined as the number of k-mers that are observed for all possible k-mer lengths, divided by the total number that are theoretically possible.

Write a python script that, when run using the command line, outputs the linguistic complexity of each sequence in a file of sequences. For the sake of simplicity let's assume that we use only 4 possible characters (A, C, G, T), so the number of possible k-mers is 4^k. The file should be specified by the end user as a command line argument.

As an example, consider the string ATTTGGATT. From the following table you can see that the linguistic complexity is 35 / 40 = 0.875. Note that the possible number of kmers (usually 4^k) is limited by the length of the sequence. Thus Possible Kmers is calculated as the minimum of (1) the length of the string minus k plus 1, and (2) 4^k (i.e. the number of possible k-mers of length 9 in the sequence is 1, not 4^9).

k	Observed kmers	Possible kmers
1	3	4
2	5	8
3	6	7
4	6	6
5	5	5
6	4	4
7	3	3
8	2	2
9	1	1
Total	35	40

To achieve this goal:

1. Define a function to count kmers of size k, where k is specified as an argument.
2. Define a function to create a pandas data frame containing all possible k and the associated number of observed and expected kmers (see above table).
3. Define a function to calculate linguistic complexity.
4. Be sure that all your functions have appropriate docstrings.

5. Use the main function in your script to read in your file and output results to files.
6. Write a script to *thoroughly* test each of your functions.
7. Include thorough comments for all of your code.
8. Create a github repository including a README (in markdown) to submit your work.

Submit your work as a link to a github repo. The repo should have

1. README with instructions on what the code does and how to run it.
2. Python script
3. Python test script
4. Example data file with a couple of short strings

When I run pytest everything should pass.

When I run the python script on the data file I should get one output file for each string containing a data frame, and a statement about complexity printed to the command line or saved in a separate file.

This exam is open book and open note. You may take as much time as you like. You may discuss this exam with classmates and the instructor. However, I rely on your personal ethics to ensure that you do not copy from classmates or have others do your work for you.