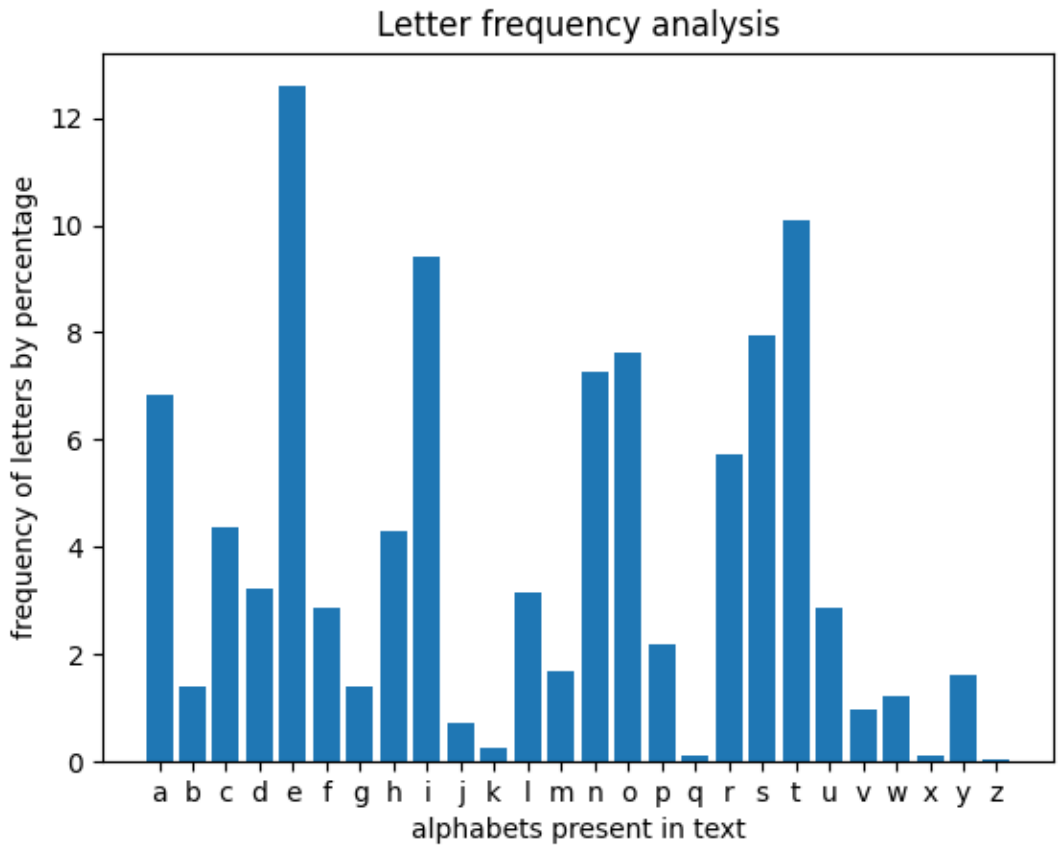
Name: Kokil Dhakal
Collaborator: None
Source: https://stackoverflow.com/questions/14032521/python-data-structure-sort-list-alphabetically
Extension: None

Working process of lab 4:
In this lab I am trying to analyze the data present in doc1.txt document using python language. I am writing some explanation of each of the method trying to do in this section.

1. word_count(): This method is basically count total number of words present in the document using read() and split() method.
2. character_count(): This method is basically count total number of characters present in the document that is including all letter, number, punctuation etc.
3. sentence_count(): this method is used to count total number of sentences present in the documents. This is pretty much easy method as there are three characters [!.?] one of which is used to end sentences. That is why I just count total number of those three characters present in the documents.
4. longest_word(): in this method I made a list of words present in the document and have the length of each word. and find length of word that has high value.
5. average_word_length(): this method is also straight forward. First, I tried to calculate the total length of all words and divide those length by total number of words.
6. average_sentence_length(): in this method I just call two methods word_count() and sentence_count() and divide word_count by sentence_count.
7. longest_sentence() and shortest_sentence(): both of this methods are similar. First, I made list of sentences, then find of length of each sentence. Find shortest and longest sentences.
8. most_frequent_words(): in this methods, first I made list of words and then make dictionary in which each unique word will have its frequency number as value. Now that dictionary is sorted based on value first. When there is tie, it is sorted by key's alphabetical order.
9. num_distinct_words(): in this method, I made list of words and then make set from list of words.
10. List_of_sentences(): in this method, I make list of words from the given documents after that find the indexes of the gives words in that list of words.
11. Word by prefix (): this method is also easy. First made list of words from documents and then made another list of word that start with given prefix using starts with () method. And sort () method at last to sort words alphabetically.
12. find_word():In this method first I made a list of all words excluding all characters and then find index of word we are looking for.one word can have multiple indices, that is why I used enumerate() function with value and its index and make list of indexes.
13. replace_word (): in this method. First, I make list of words and then count the number of words like new words already present in the list of words. After that I replaced the old word with new words and final count the total number of replaced word which is equal to total number of words (eg injustice) minus total number of words (i.e. Injustice) already present in list of word before replacing.
14. spell_check (): for this, I convert list of words in dictionary documents and doc1 documents. And I take each word from doc1 and check whether it is present in the list of words in dictionary documents. If it is not present in dictionary, I take its index and make an index of list of words that does not present in the dictionary.
15. auto_complete():in this method, first I make list of words and then make list of words that starts with the given input_string.After that make a dictionary which have words their counts as values. After that, I will sort in decreasing order. And take first 3 values which has higher number of words and their counts. If their words have same number of counts, they will be sorted alphabetically.
16. Character_fingerprint. In this method, I did make list of letters excluding all other characters. And then count the frequency of each letter present in the documents using dictionary and then it is sorted using sorted () method. At last, using matplotlib made chart of frequency of each letter.


Letter frequency analysis
frequency of letters by percentage
0
2
4
6
8
10
12
a b c d e f g h i j k l m n o p q r s t u v w x y z
alphabets present in text