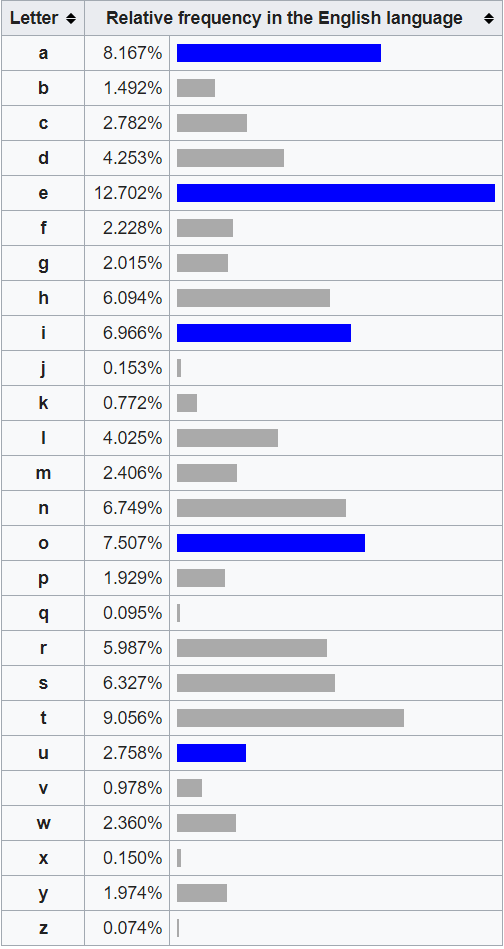
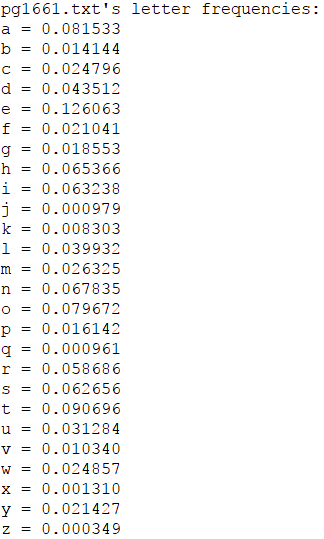
**Exercise 1 Report**

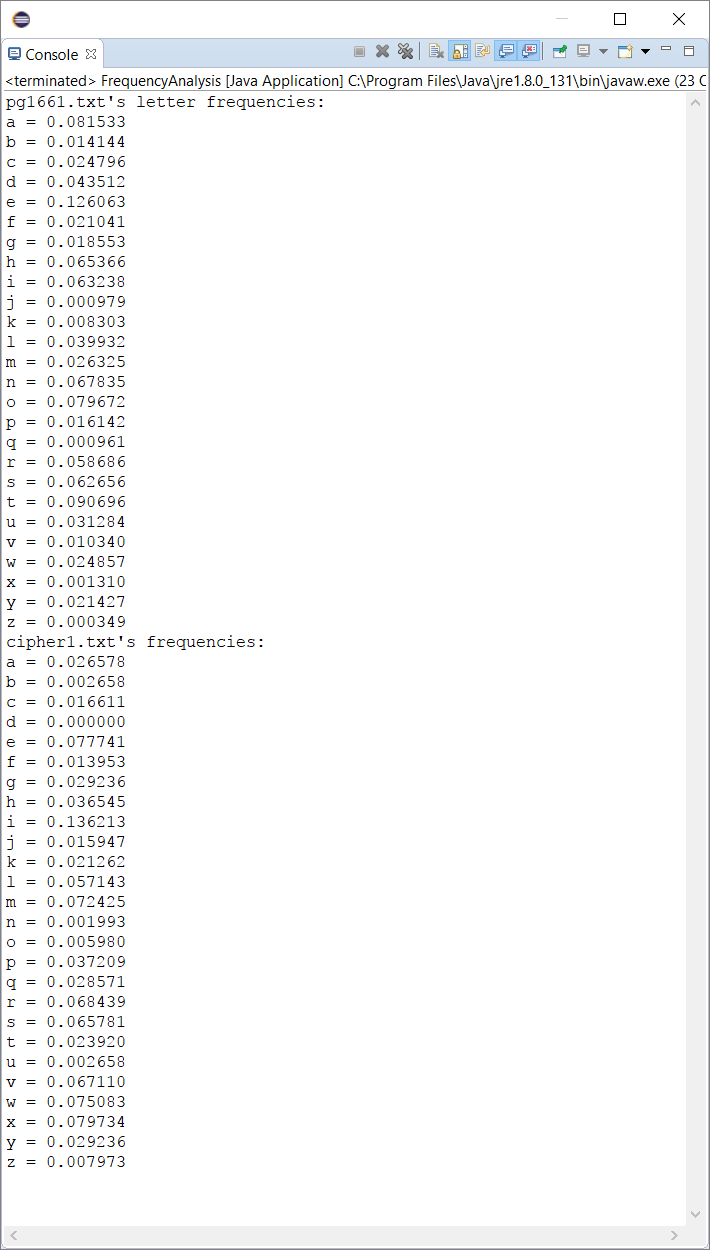
I initially found this exercise to be quite difficult. First, I set out to get the frequency of each letter in each of the .txt files, and then based on these, I tried to reprint the cipher text to a new file, altering each character using the letter frequency. The nth most common letter in the cipher text would be replaced with the nth most common letter from pg1661.txt, for example ‘i’ would be replaced with ‘e’, and ‘d’ would be replaced with ‘z’ etc. This didn’t work.

I then noticed that the most and least common letters of each file had just 3 letters between them. This showed me that the file may have been encrypted using a Caesar cipher, so I rewrote cipher1.txt character by character, replacing each with the character 4 places behind it. This did work, giving the plaintext in ‘deciphered message.txt’.

My frequency analysis was very similar to commonly known results, as shown in this side by side comparison:



A screenshot of my full console output is shown here. This helped me, as it shows ‘i’ was the most frequent letter in the cipher text, and ‘d’ was the least frequent, not appearing at all. This led me to the realisation that a Caesar cipher may have been used.



I implemented a getKeyFromValue method, which would return the character that was mapped to a certain frequency value. This would have been used to order my results, but it didn’t work as the cipher text had several letters with the same frequency, b and u, and g and y.