**Frequency Analysis**

The following works will explain how I analysed a text file, containing only English letters and words, and used the results to determine how to decrypt some cipher text. I began by writing a program that would first analyse the text and then put the analysed letters in highest frequency order. I felt this would be beneficial later when I needed to decrypt the cipher text.

When I compared the plain text frequency with the known letter frequency found on Wikipedia, I found that though not all of the letters are in the exact same order of frequency. The difference in some letters is subtle, like a simple swap of place by two neighbouring letters, though in some places it was as many as 6 letters. Though letter frequency can vary, and is often determined on a writer and the subject. To get a more accurate letter frequency it would be better to analyse several texts and average the frequency. This is not the purpose of the task I am undertaking.

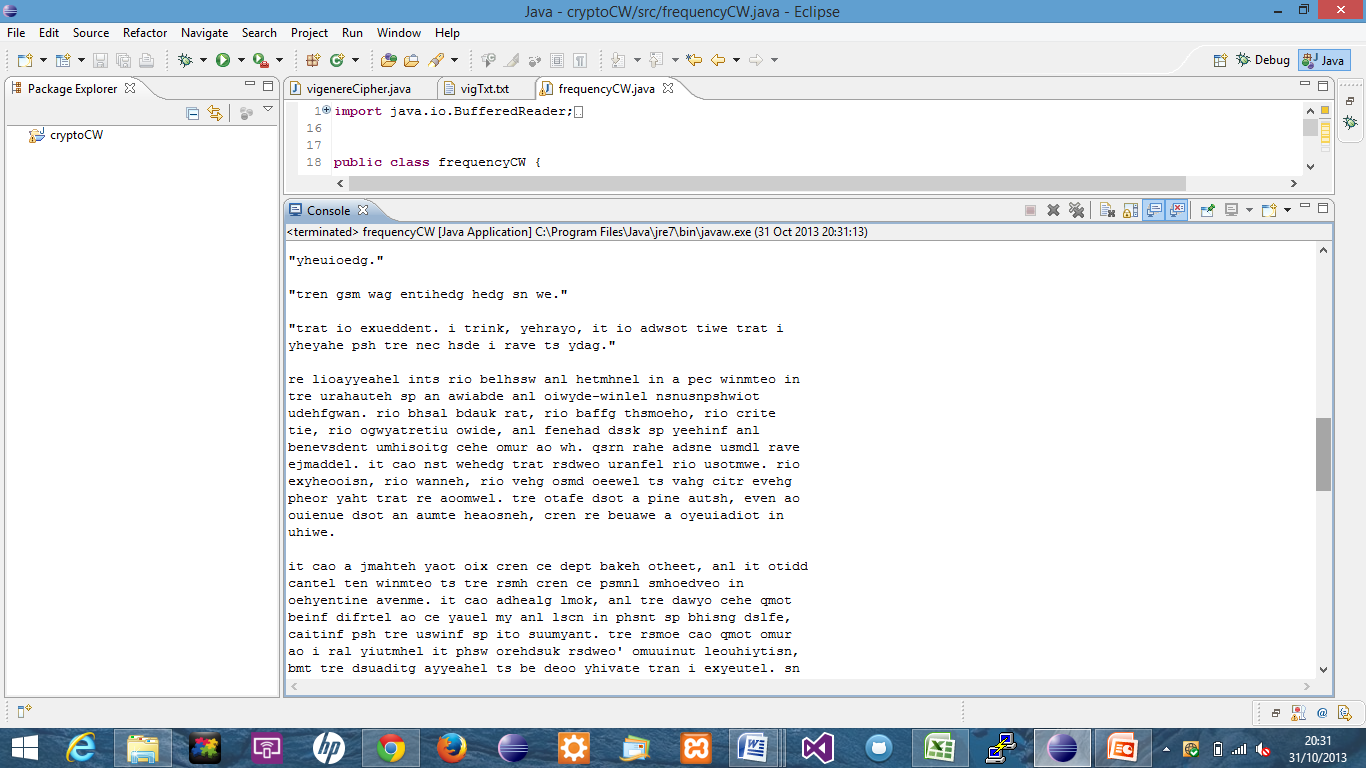
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Text | e | t | a | o | i | n | h | s | r | d | l | u | m | w | c | y | f | g | p | b | v | k | x | j | q | z |
| Wiki[1] | e | t | a | o | i | n | s | h | r | d | l | c | u | m | w | f | g | y | p | b | v | k | j | x | q | z |

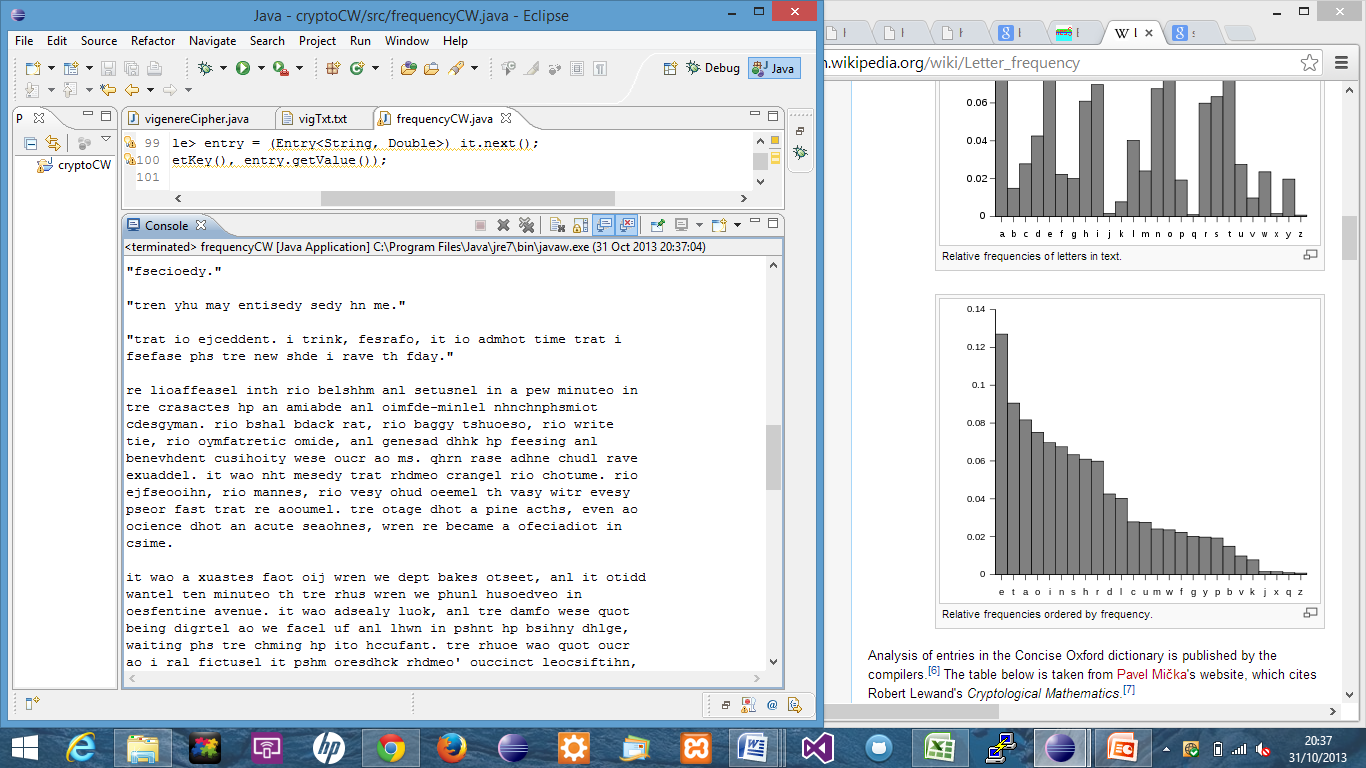
Though when compared on a graph, the plain text frequencies have similar percentages to those frequencies known.

I then compared the plain text analyses with the cipher text analysis, I could see that some letters occurred more frequently like i in the cipher text and e in the plain text. This is difficult to analyses on a graph.

When looked at in text form it is easier to see the letters that are closely matched. I also found that the cipher text only used 25 letters which through elimination I found the missing letter to be d.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cipher | i | x | e | w | m | r | v | s | l | p | h | g | y | q | a | t | k | c | j | f | z | o | b | u | n |  |
| plain | e | t | a | s | i | n | r | o | h | l | d | c | u | m | w | p | g | y | f | b | v | k | x | q | j | z |

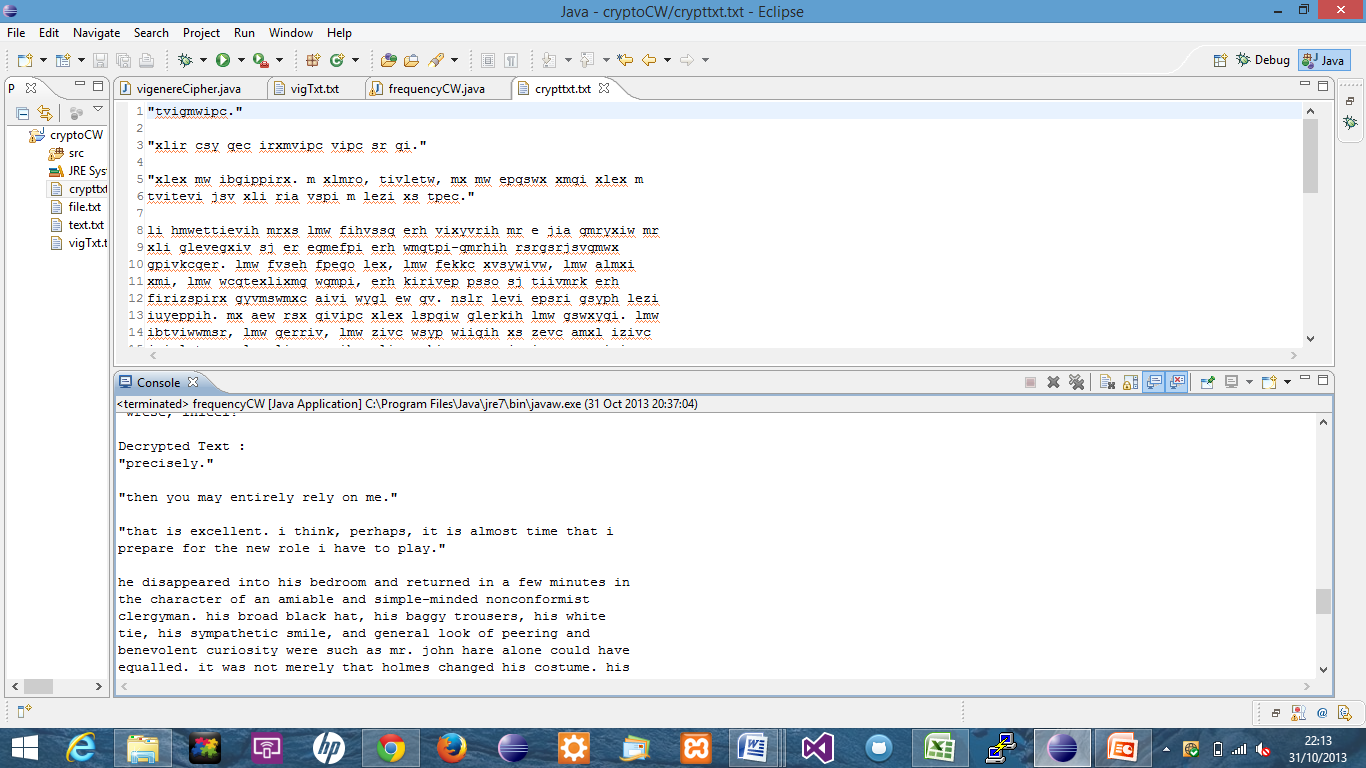
I decided to use the order of frequency to decrypt the cipher text, replacing the highest frequency letter of the cipher text with the highest letter frequency of the plain text.

I could see that this was not correct though some words had begun to take shape though no solid words; I opted to try using the known letter frequency as I had noticed some differences between the plain text frequencies.

I found that some words had been decrypted and even more had become apparent. I decided to compare what I had found and determine a pattern.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cipher | e | f | g | h | i |  |  | l | m |  |  |  | q | r |  |  |  | v | w | x | y | z |  |  | c |  |
| decrypt | 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 |

Looking at this, you can clearly see that the alphabet has been shifted five places to the left. I used this shift using ASCII values and decrypted the text.



I could have determined the shift using a method, subtracting the ASCII values of the most frequent letters, thus making my program fully automated. But I may not have fully understood how the cipher worked.