Project 1 Discussion
Pre-processing un-encrypted text files

- Use the 6 novels that I posted...
- ...to extract letter frequencies and frequencies of small words (1-letter, 2-letter or 3-letters for e.g.).

- Also you can use it to build a dictionary (as in HW8) or download an online dictionary.

- **Question:** Should you try and ignore proper nouns? How would you identify proper nouns?
Repeat the processing that you did for the unencrypted files on the cipher text...
For each $ch$, let $\text{freq}(ch)$ denote its frequency in the un-encrypted files.

For each $ch$, let $\pi(ch)$ be the set of chars whose frequencies in the cipher text are most "similar" to $\text{freq}(ch)$.

You should think about how best to define a good “similarity” measure.

Would you try and force the size of $\pi(ch)$ to be small for all $ch$?

Would $\pi(ch)$ be ordered - most likely match first?
Small word frequency matching

- Gather the most frequent small words in the cipher text.

- First match frequent 1-letter words in plain text to frequent 1-letter words in cipher text.

- In the plain text the word frequencies I found were: a 16709; b 15; c 22; d 192; e 33; f 9; g 6; h 6; i 10918; j 12; k 1; l 53; m 264, etc.

- Should you turn these into percentages for better comparison?

- This should cause $\pi(ch)$ to decrease in size for some letters $ch$. If size of $\pi(ch)$ becomes 1 for a letter $ch$, then we’ve found an exact match for $ch$. 
Repeat 2-letter and 3-letter words

- Repeat this process for 2-letter and 3-letter words

- Try to do exactly the same thing that you were doing for 1-letter words, so that it is easier to think about and you can use the same code.

- At the end of processing small words, lots of letters $ch$ (but not all) may have exact matches.

- What happens if $pi(ch)$ becomes empty for some $ch$ at this point?
Final matching

- Now consider longer words in the cipher text that have been partially deciphered.

- Find valid English words in the dictionary that match such encrypted words and use this to decrypt the missing letters.
General Advice

- Write your program in stages: at each stage you should have a *working program* that decrypts cipher texts.

- **Stage 1**: Letter frequency analysis
- **Stage 2**: Letter frequency + 1-word frequency analysis
- **Stage 3**: Letter frequency + small word frequency analysis.
- **Stage 4**: Letter frequency + small word + long word analysis