Assignment number 1
Entropy of natural language

Deadline: March, 28

Creation of "clean" text for experimentation. Pick your favorite books in plain text format (for example, you may use Project Gutenberg) and remove from the file the parts not belonging to the text. Learn how to import the content of the file as a string in Python. Write a function that does the following:

- `clean_text(txt)` takes a string `txt` of text in natural language and outputs a string that is a simplified "clean version" of it. The output string must contain only lowercase letters `a` to `z` and the space character, with words separated by a single space. The function has to convert uppercase to lowercase, turn end-of-lines to spaces, eliminate punctuation, replace "decorated" letters like á, à, â, ü, ç, ñ, l·l, etc. with their plain counterparts, ...

Create a supply of cleaned texts for experimentation, including different texts of the same author, texts of different authors in the same language, texts in different languages, etc. Share them with your classmates. Choose *very long* texts (more than half million letters, say), so that the statistics are more relevant.

Compare the average length of words in different texts.

Computation of entropies. Let `txt` be a string containing a text cleaned using the `clean_text` function. Let X be the random variable whose values are letters from `txt` taken randomly, and let X and Y be the pair of random variables whose values are two consecutive letters of `txt` taken randomly.

Write the following functions for computing information-theoretic properties of the string `txt`:

- `entropy(txt)` computes the entropy $H(X)$ of the variable X.
- `joint_entropy(txt)` computes the joint entropy $H(X, Y)$ of the pair X, Y.
- `conditional_entropy1(txt,ltr)` computes the entropy $H(Y|x_i)$ of the random variable $Y|x_i$ whose value is a random letter of `txt` after the letter $x_i =$ `ltr`. Here `ltr` denotes one of the letters of `txt`.

- `conditional_entropy(txt)` computes the conditional entropy $H(Y|X)$.

Use these functions to experiment with the texts created previously. In particular, compute and compare the several entropies for different texts (same author, different author same language, different languages, etc.), and also the entropy of the first letter (resp. the last letter) of a word. What are the x_i for which $H(Y|x_i)$ is maximum and minimum?

Check your functions using the identity

$$\sum_{i=1}^{m} p(x_i)h(Y|x_i) = H(Y|X) = H(X,Y) - H(X).$$

Creation of new text. The objective now is to create "artificial" with the same distribution of probabilities of letters than a given text `txt`, belonging to a natural language. You will need to import functions from the random package. Write the following functions:

- `new_text(txt)` outputs a random text with the same frequencies of letters of the input text `txt`.
- `new_text_joint(txt)` outputs a random text with the same joint (and conditional) entropy than the input text `txt`, with respect to pairs of consecutive letters.

With several texts do the following: save the original text belonging to natural language and also two the artificial random texts with the same probabilities distributions in three different files. Compress the three files using a standard compressor (for example Zip or 7-Zip). Compare the results. Experiment with different parameters in the compressors.

Delivering. Deliver a .py file including your implementations of the functions and also (in commented lines) a short report on your experiments and the things you observed that have drawn your attention more.