ISTA 455/555: Project #1 (10 pts)
Analyzing Happiness

Due by 11:59 P.M., September 8

For this project you must submit:

- A single file containing the python code used to address all programming questions.
- A PDF document that must contain at least:
  - Description of the command line to run the python code, with an example.
  - Description of the code. You don't have to describe every function implemented. But you should describe the main part of the code and indicate where each question is addressed.
  - Results, i.e., the output of your code, for all the questions that required programming.
  - Answers for all questions that do not require programming.

Answers are always graded by inspecting both code and documentation. Missing code yields no credit. Missing documentation yields partial credit (if code exists and produces correct results).

Because the credit for graduate students adds to more than 10, graduate students' grades will be normalized at the end to be out of 10. For example, if a graduate student obtains 12 points on this project, the final grade will be $12 \times \frac{10}{13} = 9.23$. Undergraduate students do not have to solve the problems marked "grad students only." If they do, their grades will not be normalized but will be capped at 10. For example, if an undergraduate student obtains 11 points on this project (by getting some credit on the "grad students only" problem), her final grade will be 10.

Download the file `project1_tweets.txt`. This file contains almost 7,000 tweets, each tweet listed on a separate line. Using this file, solve the following problems:

1) (3 pts) Implement a tokenizer for tweets. If using `python`, you can start with NLTK's regular expression tokenizer (see the paragraph titled "NLTK's Regular Expression Tokenizer" in Section 3.7 of the NLTK book: `http://nltk.org/book/ch03.html`). Extend it to tokenize correctly: (a) emoticons (e.g., ":)" must be a single token), (b) Twitter user name (e.g., "@freakonomics" must be a single token), (c) hash tags (e.g., "#happy" must be a single token), and URLs (e.g., "https://twitter.com/freakonomics" must be a single token). Christopher Potts implemented regular expressions for most of these tokens here: `http://sentiment.christopherpotts.net/tokenizing.html`. You can use them in your code.

2) (2 pts) Which hashtags appear most commonly in the same tweet with the hashtag #happy? How about #sad? Write code to find the top 20 hashtags associated with #happy and #sad, respectively. Hint: you can use NLTK's `FreqDist` class to keep track of counts. Include the output of this code in the project document.

3) (2 pts) Which nouns appear most commonly in the same tweet with the hashtag #happy? How about #sad? Write code to find the top 20 nouns associated with #happy and #sad, respectively. Include the output of this code in the project document.

4) (1 pts) Which verbs appear most commonly in the same tweet with the hashtag #happy? How about #sad? Write code to find the top 20 verbs associated with #happy and #sad, respectively. Include the output of this code in the project document.

5) (2 pts) What problems do you observe with the output of the problems 4 and 5? How would you fix it? Describe potential solutions in the project document. No coding required.

2) (3 pts) **GRAD STUDENTS ONLY** Rank the hashtags that are commonly associated with the hashtags #happy and #sad, respectively, in descending order of their *conditional proportions*. You can compute the conditional proportions as the ratio of two counts. The numerator counts how many times you have seen the corresponding hashtag and #happy (or #sad, respectively) together in the same tweet. The denominator counts how many times you have the corresponding hashtag overall. Include the output of this code in the project document. What do you observe? How is this output different from problem 2? How would you fix any potential problems?