Annabel Consilvio
Software Design Spring 2015
Mini- Project 3

Determining The Happiness of a Given Year Based on Gutenberg's Most Popular Books

Project Overview

Using Gutenberg's list of the Top 100 Most Popular books, I created a program that evaluates the happiness of a book's text and plots it according to when the book was published. The graph generated is supposed to give a general idea of the happiness of people in a given year, assuming that their happiness level is similar to the happiness level of the most popular book of that year.

Implementation

I used a variety of data structues in this program in order to sucessfully generate a happiness based graph. My program builds iteratively, and I was able to utilize strings, lists, dictionaries and tuples within it. The program begins simply by generating a list of strings which in turn get used as the keys in dictionaries and directly determine the keys' values. Eventually, using matplotlib, the program is able to precisely generate a graph of relative happiness over the years 1700-1900 using the initial list of strings and the subsequent dictionary.

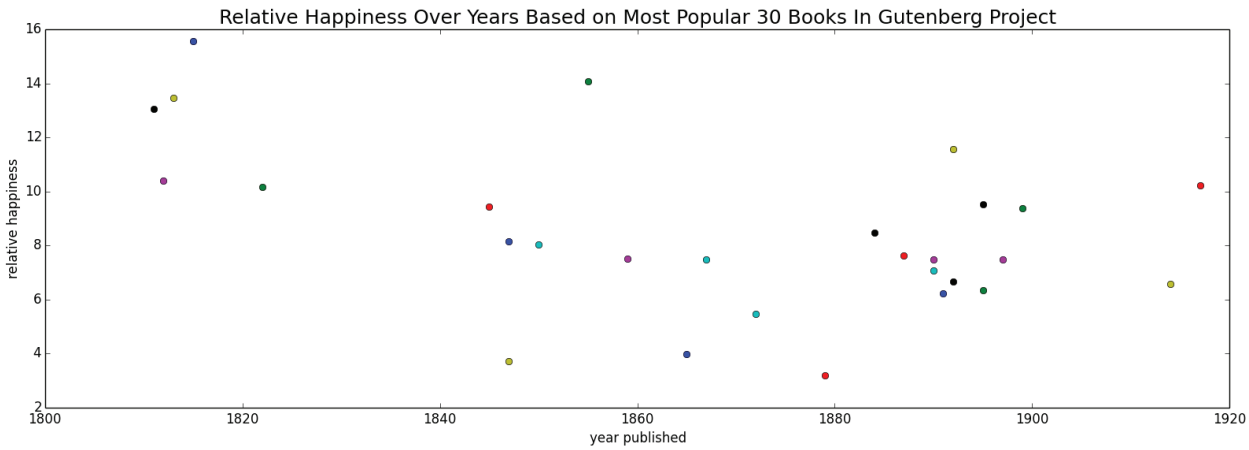
Because of the connection in dictionaries between the key and its value, I determined that it would be best to store my generated data in a dictionary. Additionally, the order in which the books (strings in the inital list)were graphed did not influence anything, so it was not important to keep the order consistent while plotting, so again the dictionary was a good choice. My dictionary of books is generated by calling a list of books and setting each key to have two values: a publishing year (integer) and a sentiment value (tuple). The program itself is looking for these two values in relation to each other to plot, so storing them in a dictionary allowed me to call them both from the same key, ensuring that they would always correspond with each other. Lists and other data structures may have made this process more complicated and the (year, happiness) pair could have been compromised.

Results

In this program, I was able to visualize the happiness level of ~30 of the most popular books from the Gutenberg Press. However, this program can be run with any set of books pickled from Gutenberg that include the publishing date early on in the text. I found some interesting things from my model. To begin with, I learned that nearly all books fall under the 20% happiness mark using the pattern sentiment tool. I believe that this is because the texts used in the program are so long, even if the books themselves are upbeat, the "happy" word precentage will be diluted by all the other words with no positive or negative connotation.

Additionally, I learned that book sentiment is less predictable than I imagined. I thought there might be some sort of trend within the books (ie going through a war would make the overall happiness level lower), but it appears that this trend is not in the most popular books of the time. However, it is still interesting to see how some of the greatest books in history appear on this graph. Moby Dick, Pride and Prejudice, and Frederick Douglass's Life Story are all shown in the graph at their respective happiness levels, and we can find that Douglass's writing (1845) is rather low on this particular happiness scale, Pride and Prejudice (1813) is rather high, and Moby Dick (1850) comes in about average.

The following graph shows the top ~30 Gutenberg press books according to publishing date and relative happiness level (out of 100). As you can see, there is no real pattern among the books used in this analysis. However, a trend may appear if you plotted books based on subject matter and not populatrity.



Reflection

There were multiple things that went well for me during this project. To begin with, I used a variety of combinations of data structures that I had never explored, so I learned quite a bit about how to efficently store data. Additionally, I learned how to plot things in python. I had never used matplotlib before this project, so consdiering that, I think the plotting portion of this program went well. On the other hand, in the future, I would like to improve how I search for publishing dates in the text (possibly using google instead of the physical text) to make sure they are accurate. Additionally, I think that my program could be more efficient in a variety of places, but due to the time constraint, I was not able to explore these. Overall, I think the project was well scoped considering my knowledge (or lack thereof) of python and the very short timeline. I just wish we had more time to work on this project.