The prof mostly references the text book. The below are excerpts from it.

Let’s continue to use the same text as we used in the previous chapter (Susan B. Anthony’s speech). Was it a positive speech? Let’s find out. First, let’s load the positive and negative files and clean them up to get ready to use. To find a positive and a negative word list, we can use a nice website on sentiment analysis: https:// www.cs.uic.edu/ ~ liub/ FBS/ sentiment-analysis.html About halfway down the web page, you can see a section titled “Opinion Lexicon (or Sentiment Lexicon).” The first bullet in this section contains a link to a list of positive and negative opinion words or sentiment words for English. When you click on this link it downloads a compressed file that contains a folder that has both positive and negative words. Specifically, there will be one text file is for the positive words and one text file is for the negative words. Once you download those files, save them to a place where R can easily access them. Now let’s get to the R coding:

#define the name of the word files

pos <- "positive-word.txt"

neg <- negative-words.txt

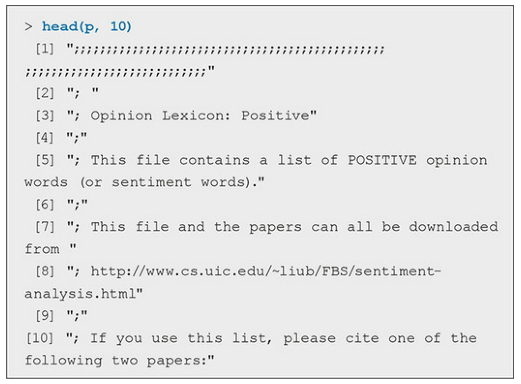
#read the files and separate each word

p <- scan(pos, character(0), sep = "\n")

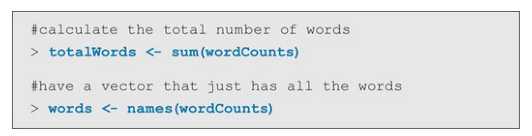
Read 2040 items

n <- scan(neg, character(0), sep = "\n")

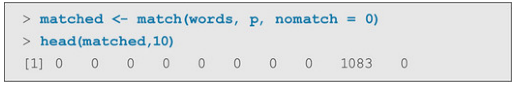
OK. Let’s review what we just did. First, we read the text files. Note this code assumes the text files are in R’s default directory. If you didn’t know, you can get and set R’s working directory with the getwd and setwd functions. The positive file had 2,040 words, and the negative file had 4,817 words, but some of this might have been headers. This seems to suggest that there are more ways to say something in a negative way than in a positive way. That’s kind of interesting in itself, but let’s press on and take a look at the start of the positive file:



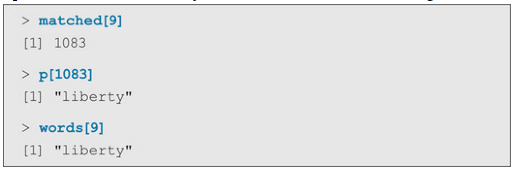
That looks like the start of a good list of positive and negative words! Since we already have the word file we want to examine (from the previous chapter, the wordCounts variable), the next step is to count the total number of words in our text (stored in wordCounts).



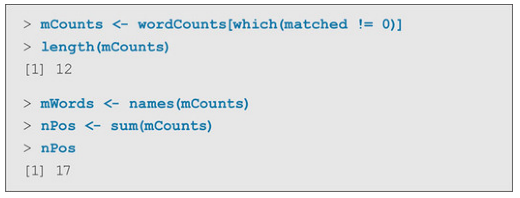
Next, we somehow need to figure out how to count the number of positive and negative words in wordCounts. To do this, we need to use the match function, which returns a vector of the positions of (first) matches of its first argument in its second. When there is no match, we can specify to return a specific number (such as zero [0]).



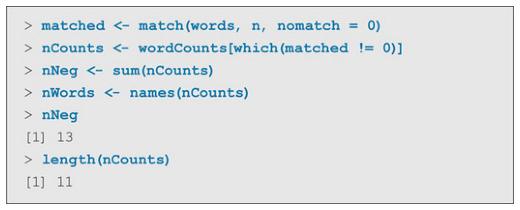
We can see that the first eight words did not match, but the ninth word did match. This is confirmed with the code below, where we can see that the ninth word in “words” should be the same as the 1,083rd word in “p”— and indeed, they are both the word liberty.



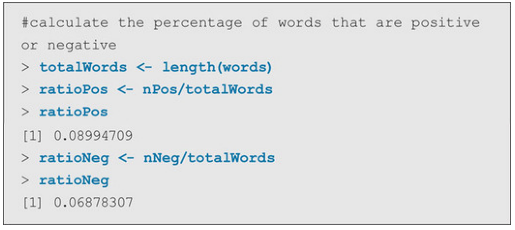
Now that we have the list of matched positive words, we just have to get the counts of all the words that did match.



So, we can see that there were 17 positive words (note that some of the words were repeated, and in fact, there were 12 unique positive words used in the speech— which is why there was only 12 items in mCounts). Now let’s do the same for the negative words.



We can see that there were 13 negative words (and 11 unique negative words that were used). Great! After all this work, we have all the information we need to calculate the percentage of positive and negative words for this speech, which you can see in the following lines of R code:



Given this, we can see that Susan B. Anthony’s speech was made up of about 9% positive and a little less than 7% negative words. What do you conclude about the nature of her speech?