Homework 8

Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. You may work with students in both sections (375-01, -02). Only one person in the group should submit to Canvas.
Due: Names: Serop Kelkelian and Bryce Lin

1. Create a word cloud of some of the most commonly occurring words in Jane Austen's novel "Sense & Sensibility". The full text of all six novels are available in the `janeaustenr` package and its function `austen_books()`.

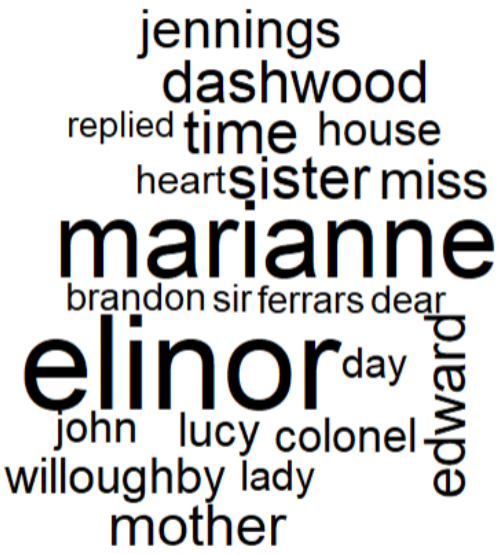
```
install.packages("janeaustenr")
library(janeaustenr)
austen_books() %>% …
```

Write R code to do the following tasks (you can refer to the text processing R code posted on Canvas):

- Get only the text of "Sense & Sensibility"
  ```
  austen_books() %>% filter(book == "Sense & Sensibility")
  ```
- Convert your data to the "tidy" format, i.e., one word per row (Hint: `unnest_tokens()`).
  ```
  text_df <- austen_books() %>% filter(book == "Sense & Sensibility") %>%
  select(text) %>% unnest_tokens(word, "text")
  ```
- Remove stop words (Hint: `anti_join(stop_words)`)
  ```
  text_df <- text_df %>% anti_join(stop_words)
  ```
- Retain only words that appear greater than 100 times in the novel (Hint: `count()`)
  ```
  text_df <- text_df %>% count(word) %>% filter(n > 100)
  ```

- Create a wordcloud

```
wordcloud(text_df$word, text_df$n)
```



Include both your code and the word cloud image.

2. Consider the following short documents [same as class work]:

Document 1: *good morning everybody*
Document 2: *good evening everybody*
Document 3: *good night*

This question is meant to be done by hand though you can use R to check your work (next question).

a) Show the Document-Term matrix weighted by Term-frequency (Tf).

	Document 1	Document 2	Document 3
good	1/3	1/3	1/2
morning	1/3	0	0
evening	0	1/3	0
everybody	1/3	1/3	0
night	0	0	1/2

b) What is the inverse document frequency (Idf) of each word?

i) Idf(*good*) = ln(3/3) = 0
ii) Idf(*morning*) = ln(3/1) = 1.099
iii) Idf(*evening*) = ln(3/1) = 1.099
iv) Idf(*everybody*) = ln(3/2) = 0.405
v) Idf(*night*) = ln(3/1) = 1.099

c) Show the Document-Term matrix weighted by Tf-Idf for this dataset.

	Document 1	Document 2	Document 3
good	⅓ * 0 = 0	⅓ * 0 = 0	½ * 0 = 0
morning	⅓ * 1.099 = 0.366	0	0
evening	0	⅓ * 1.099 = 0.366	0
everybody	⅓ * 0.405 = 0.135	⅓ * 0.405 = 0.135	0
night	0	0	½ * 1.099 = 0.55

3. Write R code to create the final Tf-Idf weighted Document-Term matrix (i.e., one column for every document) for the same three documents in Q2 using the tidytext package. [Give code and output]

```
    mydata <- tibble(document=1:3,text=c("good morning everybody", "good
evening everybody", "good night"))
    ...
```

```
mydata %>% unnest_tokens(input = text, output = "word") %>% count(document, word)
%>% bind_tf_idf(term=word, document = document, n=n) %>% select(document, word,
tf_idf) %>% pivot_wider(names_from = document, values_from = tf_idf, values_fill = 0)
```

4. Define a function that takes two vectors as input and computes their cosine similarity:

$$cossim(A, B) = \frac{sum(A_i B_i)}{\sqrt{sum(A_i^{\,2}) \times sum(B_i^{\,2})}}$$

○ Note: your function does not need any loops as you can use the vectorized operators in R. Test your function for correctness. For instance, the cosine similarity of any vector to itself is 1 and cosine similarity between vectors (1,2,3) and (0,2,5) should be 0.9429

- Give code, and output of `cossim( c(1,2,3), c(0,2,5) )`

```
cossim <- function(x, y){
   x.norm <- x / sqrt(sum(x^2))
   y.norm <- y / sqrt(sum(y^2))
   return (sum(x.norm*y.norm))
}
```

5. Use the function defined in Q3 to compute the cosine similarity between every pair of documents in Q1.

cossim(Document 1, Document 2) = 0.1198832
cossim(Document 1, Document 3) = 0
cossim(Document 2, Document 3) = 0

Which of the three documents is the odd one out? Why (1-2 sentences)?

Document 3, because IDF removes the only common words it has from other documents, and the unique word in Document 3 is not in any other documents.

5(a). The goal of this exercise is to use the word-based analysis from class to compare the six novels of Jane Austen. Use the `janeaustenr` package from Q1.

Write a single data pipeline that will transform the text data into Tf-Idf weighted vectors. The pipeline should:

- Convert your data to the "tidy" format, i.e., one word per row.
- Remove stop words
- Stem the words (Hint: `wordStem()` from `library(SnowballC)`)
- Retain only words that appear greater than 5 times in a novel
- Calculate Tf-Idf weights
- Convert to a table of vectors format (a column represents the vector representation of a novel).

Give the R code that performs these steps. What is the dimension of the vectors?

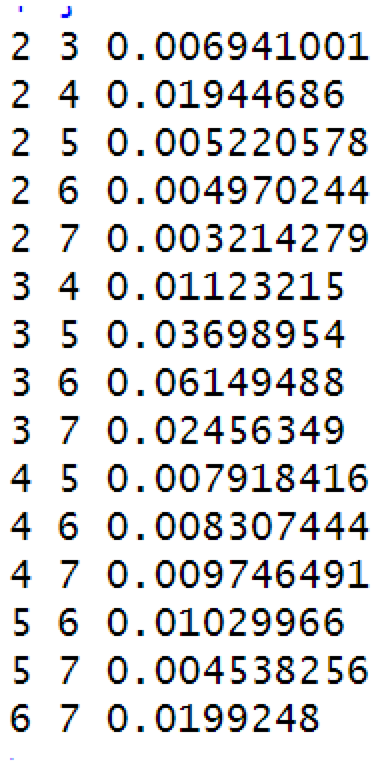
```
text_df <- austen_books() %>% group_by(book) %>% unnest_tokens(word, text) %>%
anti_join(stop_words) %>% count(book, word) %>% filter(n > 5) %>% bind_tf_idf(term =
word, document = book, n = n) %>% select(book, word, tf_idf) %>%
pivot_wider(names_from = book, values_from = tf_idf, values_fill = 0)
```

5(b). Write R code to compare ***every pair*** of Jane Austen's novels by computing the cosine similarity between their corresponding tf-idf vectors calculated in Q5(a).

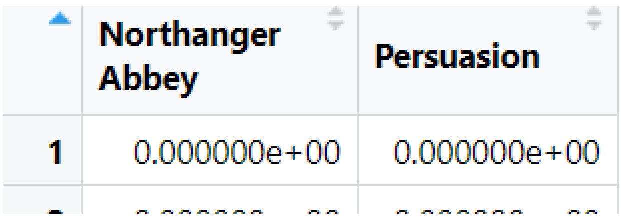
1. Give the R code that performs these steps. Hint: you may need to use loops and the `cossim()` function you wrote earlier.
2. Which two novels are the most similar?

3. What is the cosine similarity of these two novels?

```
for (i in 2:6){

   for(j in (i+1):7){

      cat(i, j, cossim(text_df[i], text_df[j]), "\n")

   }

}
```



```
2 3 0.006941001
2 4 0.01944686
2 5 0.005220578
2 6 0.004970244
2 7 0.003214279
3 4 0.01123215
3 5 0.03698954
3 6 0.06149488
3 7 0.02456349
4 5 0.007918416
4 6 0.008307444
4 7 0.009746491
5 6 0.01029966
5 7 0.004538256
6 7 0.0199248
```



	Northanger Abbey	Persuasion
1	0.000000e+00	0.000000e+00

These 2, 0.0199248