A simple example of Text Analysis with R

Claudio Sartori

University of Bologna
DISI – Department of Computer Science and Engineering

Task

Extract from a normal text, such as a book chapter, the *most frequent* and *significant* words, then plots them in a *wordcloud*, according to the frequencies.
This is the workflow:

1. Load text
2. Clean removing non-alphabetic characters
3. Reduce all characters to lowercase
4. Split lines into words
5. Remove empty words
6. Compute frequencies of words and store them into a dataframe
7. Sort by descending frequency
8. Download the list of stopwords
9. Exclude stopwords from the dataframe
10. Produce a word-cloud with the most frequent words

The statements preceded by the comment `# control` are added to check if the transformations proceed as desired

1 Load text

[1]:
```
rm(list = ls())
#file_ref <- "https://www.gutenberg.org/files/2701/old/moby10b.txt"
file_ref="moby_dick_ch01.txt"
text_lines <- readLines(file_ref)
```

[2]:
```
# control
cat(text_lines[1])
```

```
Call me Ishmael. Some years ago-never mind how long precisely-having
```

[3]:
```
# control
length(text_lines)
```

174

2 Clean removing non-alphabetic characters

[4]:
```
text_clean0 <- gsub("[^a-zA-Z]", " ", text_lines)
# control
text_clean0[1]
```

'Call me Ishmael Some years ago never mind how long precisely having'

3 Reduce all characters to lowercase

[5]:
```
text_clean <- tolower(text_clean0)
# control
text_clean[1]
```

'call me ishmael some years ago never mind how long precisely having'

4 Split lines into words

`strsplit` function split the first *argument* according to the *separator* in the second argument

[6]:
```
# control - test before doing the actual transformation
strsplit(text_clean[1], " ")
```

1. (a) 'call' (b) 'me' (c) 'ishmael' (d) '' (e) 'some' (f) 'years' (g) 'ago' (h) 'never' (i) 'mind' (j) 'how' (k) 'long' (l) 'precisely' (m) 'having'

[7]:
```
# strsplit produces a list - unlist flatten the list
# control - test before doing the actual transformation
unlist(strsplit(text_clean[1], " "))
```

1. 'call' 2. 'me' 3. 'ishmael' 4. '' 5. 'some' 6. 'years' 7. 'ago' 8. 'never' 9. 'mind' 10. 'how' 11. 'long' 12. 'precisely' 13. 'having'

[8]:
```
words0 <- unlist(strsplit(text_clean, " "))
# control
head(words0)
```

1. 'call' 2. 'me' 3. 'ishmael' 4. '' 5. 'some' 6. 'years'

5 Remove empty words

[9]:
```
words <- words0[words0 != ""]
# control
head(words)
```

1. 'call' 2. 'me' 3. 'ishmael' 4. 'some' 5. 'years' 6. 'ago'

6 Compute frequencies of words and store them into a dataframe

The `table` function uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels.

In this case there is only one factor, the word itself, and a one-dimensional contingency table is produced

[10]:
```
words_tb <- as.data.frame(table(words))
# control
head(words_tb)
```

A data.frame: 6 × 2

	words <fct>	Freq <int>
1	a	69
2	abandon	1
3	abominate	1
4	about	7
5	absent	1
6	account	2

7 Sort by descending frequency

The `order` function produces the index permutation which sorts a vector in ascending order.

To obtain the descending order it is sufficient to change the sign of frequencies

[11]:
```
words_tb <- words_tb[order(-words_tb$Freq),]
# control
head(words_tb)
```

A data.frame: 6 × 2

	words <fct>	Freq <int>
726	the	124
472	of	81
28	and	73
1	a	69
753	to	53
327	in	48

[12]:
```
# control
nrow(words_tb)
```

854

8 Download the list of stopwords

Stopwords are the words in a langage whose frequency to make them too common to allow any insight on the text

[13]:
```
# stopwords_file_ref <- "https://raw.githubusercontent.com/stopwords-iso/stopwords-en/master/stopwords-en.txt"
stopwords_file_ref <- "stopwords-en.txt"
stopwords <- readLines(stopwords_file_ref)
# control
head(stopwords)
```

1. '\'ll' 2. '\'tis' 3. '\'twas' 4. '\'ve' 5. '10' 6. '39'

9 Exclude stopwords from the dataframe

The `setdiff` function computes the set difference where sets are represented as vectors

[14]:
```
words_non_stop <- setdiff(words_tb$words, stopwords)
words_non_stop_tb <-
  words_tb[is.element(words_tb$words, words_non_stop), ]
# control
head(words_non_stop_tb)
```

A data.frame: 6 × 2

	words <fct>	Freq <int>
624	sea	13
807	water	8
371	land	6
752	time	6
801	voyage	6
611	sailor	5

[15]:
```
# control
nrow(words_non_stop_tb)
```

605

10 Produce a word-cloud with the most frequent words

[16]:
```
# uncomment line below if the wordcloud package was never installed
# then comment it again
# install.packages("wordcloud")
```

[17]:
```
require(wordcloud)
# library(RColorBrewer)
```

```
Loading required package: wordcloud

Loading required package: RColorBrewer
```

[18]:
```
# will display only words with frequency not less than threshold
freqThreshold = 3
# control
cat("There are", sum(words_non_stop_tb$Freq>=freqThreshold)
    , "words with frequency not less than", freqThreshold)
```

```
There are 27 words with frequency not less than 3
```

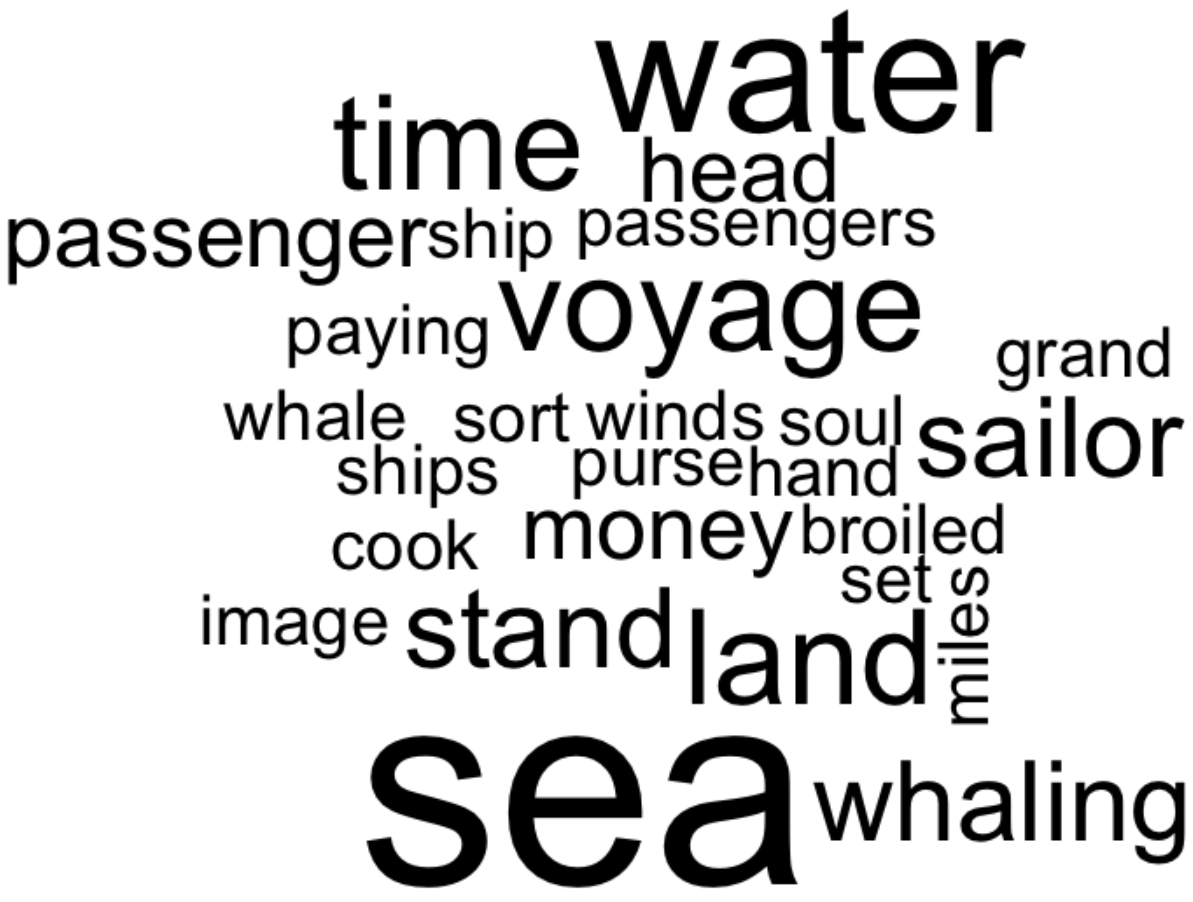
[19]:
```
# Control
words_non_stop_tb[words_non_stop_tb$Freq>=freqThreshold,]
```

A data.frame: 27 × 2

	words <fct>	Freq <int>
624	sea	13
807	water	8
371	land	6
752	time	6
801	voyage	6
611	sailor	5
685	stand	5
822	whaling	5
289	head	4
431	money	4
501	passenger	4
81	broiled	3
137	cook	3
267	grand	3
278	hand	3
326	image	3
427	miles	3
502	passengers	3
507	paying	3
558	purse	3
637	set	3
643	ship	3
644	ships	3
674	sort	3
675	soul	3
821	whale	3
839	winds	3

[20]:
```
wordcloud(
      words_non_stop_tb$words[words_non_stop_tb$Freq>=freqThreshold]
    , words_non_stop_tb$Freq[words_non_stop_tb$Freq>=freqThreshold]
         , scale = c(8,.2)
```

```
)
```


water
time
head
passengership
passengers
voyage
paying
grand
whale
sort
winds
soul
sailor
ships
purse
hand
money
broiled
cook
set
image
stand
land
miles
sea
whaling

[ ]: