CREATING A WORD CLOUD IN R

Step 1: Create a text file

I have created a file wordcloud.txt in documents folder. The documents folder is working directory for my RStudio.

Step 2 : Install and load the required packages

Type the R code below, to install and load the required packages:

```
# Install
install.packages("tm")  # for text mining
install.packages("SnowballC") # for text stemming
install.packages("wordcloud") # word-cloud generator
install.packages("RColorBrewer") # color palettes
# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
```

Step 3 : Text mining

load the text

The text is loaded using **Corpus()** function from **text mining** (tm) package. Corpus is a list of a document (in our case, we only have one document).

1. **We start by importing the text file created in Step 1**

To import the file saved locally in your computer, type the following R code. You will be asked to choose the text file interactively.

```
text <- readLines(file.choose())
```

In the example below, I'll load a .txt file hosted on STHDA website:

```
# Read the text file from internet
filePath <- "http://www.sthda.com/sthda/RDoc/example-files/martin-luther-king-i-have-a-dream-speech.txt"
text <- readLines(filePath)
```

2. **Load the data as a corpus**

Corpus: A collection of written texts, especially the entire works of a particular author or a body of writing on a particular subject.

```
# Load the data as a corpus
docs <- Corpus(VectorSource(text))
```

VectorSource() function creates a corpus of character vectors

3. **Inspect the content of the document**

```
inspect(docs)
```

Text transformation

Transformation is performed using **tm_map()** function to replace, for example, special characters from the text.

Replacing "/", "@" and "|" with space:

```
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")
```

Cleaning the text

the **tm_map()** function is used to remove unnecessary white space, to convert the text to lower case, to remove common stopwords like 'the', "we".

The information value of 'stopwords' is near zero due to the fact that they are so common in a language. Removing this kind of words is useful before further analyses. For 'stopwords', supported languages are danish, dutch, english, finnish, french, german, hungarian, italian, norwegian, portuguese, russian, spanish and swedish. Language names are case sensitive.

I'll also show you how to make your own list of stopwords to remove from the text.

You could also remove numbers and punctuation with **removeNumbers** and **removePunctuation** arguments.

Another important preprocessing step is to make a **text stemming** which reduces words to their root form. In other words, this process removes suffixes from words to make it simple and to get the common origin. For example, a stemming process reduces the words "moving", "moved" and "movement" to the root word, "move".

Note that, text stemming require the package 'SnowballC'.

The R code below can be used to clean your text :

```
# Convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))
# Remove numbers
docs <- tm_map(docs, removeNumbers)
# Remove english common stopwords
docs <- tm_map(docs, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs <- tm_map(docs, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs <- tm_map(docs, removePunctuation)
# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
# Text stemming
# docs <- tm_map(docs, stemDocument)
```

Step 4 : Build a term-document matrix

Document matrix is a table containing the frequency of the words. Column names are words and row names are documents. The function *TermDocumentMatrix()* from **text mining** package can be used as follow :

```
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
```

```
        word freq
will       will   17
freedom   freedom   13
ring       ring   12
day         day   11
dream      dream   11
let         let   11
every      every    9
able       able    8
one         one    8
together together    7
```

Step 5 : Generate the Word cloud

The importance of words can be illustrated as a **word cloud** as follow :

```
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
      max.words=200, random.order=FALSE, rot.per=0.35,
      colors=brewer.pal(8, "Dark2"))
```