Word Cloud provides an excellent option to visualize the text data

in the form of tags, or words,

where the importance of a word is identified by its frequency.

install.packages("wordcloud2")

library(wordcloud2) library(readr) library(dplyr) library(e1071) library(mlbench)

install.packages("mlbench")

Text mining packages

install.packages("NLP")

install.packages("tm")

install.packages("SnowballC")

install.packages("wordcloud")

library(tm) library(SnowballC) library("wordcloud") library("RColorBrewer")

loading the data

t1 <- read_csv("Dataset/Womens Clothing E-Commerce Reviews.csv") glimpse(t1)

t1$Recommended_IND[1]

Create corpus

corpus = Corpus(VectorSource(t1$Review_Text))

Look at corpus

corpus[[1]][1]

Conversion to Lowercase

corpus = tm_map(corpus, PlainTextDocument) corpus = tm_map(corpus, tolower)

Removing Punctuation

corpus = tm_map(corpus, removePunctuation)

corpus[[1]][1]

Remove stopwords

corpus = tm_map(corpus, removeWords, c("cloth", stopwords("english")))

corpus[[1]][1]

Stemming

corpus = tm_map(corpus, stemDocument)

corpus[[1]][1]

Eliminate white spaces

corpus = tm_map(corpus, stripWhitespace)

corpus[[1]][1]

Create Document Term Matrix

DTM <- TermDocumentMatrix(corpus) mat <- as.matrix(DTM) f <- sort(rowSums(mat),decreasing=TRUE) dat <- data.frame(word = names(f),freq=f) head(dat, 5)

" Word Cloud Generation

Word Cloud in R is generated using the wordcloud function. The major arguments of this function are given below:

```
 words: The words to be plotted.

freq: The frequencies of the words.

min.freq: An argument that ensures that words with a frequency below min.freq will not be plotted in the word cloud.

max.words: The maximum number of words to be plotted.

random.order: An argument that specifies plotting of words in random order. If false, the words are plotted in decreasing frequency.

rot.per: The proportion of words with 90 degree rotation (vertical text).

colors: An argument that specifies coloring of words from least to most frequent.
"
```

WordCloud 1

set.seed(100) wordcloud(words = dat$word, freq = dat$freq, random.order=TRUE)

WordCloud 2

set.seed(100) wordcloud(words = dat$word, freq = dat$freq, random.order=FALSE)

WordCloud 3

set.seed(100) wordcloud(words = dat$word, freq = dat$freq, min.freq = 15, max.words=250, random.order=FALSE, rot.per=0.30, colors=brewer.pal(8, "Dark2"))

example 2

text <- readLines(file.choose())

Load the data as a corpus

VectorSource() function creates a corpus of character vectors

docs <- Corpus(VectorSource(text))

Inspect the content of the document

inspect(docs)

Text 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"

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)

Build a term-document matrix

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

Generate the Word cloud

```
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"))
wordcloud2(words = d$word, size = 0.7, shape = 'star')
```