Sentiment Analysis of Hotel Reviews using R

Dr. Abhay Kumar Bhadani

21st August

Overview

Sentiment analysis is part of the *Natural Language Processing (NLP)* techniques that consists in extracting emotions related to some raw texts. This is usually used on social media posts and customer reviews in order to automatically understand if some users are positive or negative and why. The goal of this study is to show how sentiment analysis can be performed using R.

Sentiment analysis is a type of text mining which aims to determine the opinion and subjectivity of its content. When applied to Hotel Reviews, the results can be representative of not only the customer's preferences, but can also reveal tastes, cultural influences and so on. There are different methods used for sentiment analysis, including training a known dataset, creating your own classifiers with rules, and using predefined lexical dictionaries (lexicons). In this session, you will use the lexicon-based approach, but I would encourage you to investigate the other methods as well as their associated trade-offs.

Analysis Levels

Just as there are different methods used for sentiment analysis, there are also different levels of analysis based on the text. These levels are typically identified as document, sentence, and word.

Whether you like it or not, guest reviews are becoming a prominent factor affecting people's bookings/purchases. Think about your past experience. When you were looking for a place to stay for a vacation on Expedia/Booking/TripAdvisor, what did you do? I am willing to bet you'd be scrolling down the screen to check on the reviews before you knew it.

In hotel reviews, the document could be defined as sentiment per year, or chart-level. Word level analysis exposes detailed information and can be used as foundational knowledge for more advanced practices in topic modeling. Sentences can help in understanding the sentiments associated with the experience the customer went through.

Data Analysis

As a business owner or employee, if you still have doubts about how important guest reviews impacts the business, it may be worth checking out some stats: In other words, guest reviews clearly influence people's booking decision, which means, you'd better pay attention to what people are saying about your hotel!

Not only do you want good reviews, but also them in a way that can help you learn the most about your customers. Reviews can tell you if you are keeping up with your customers' expectations, which is crucial for developing marketing strategies based on the personas of your customers.

Exploratory Questions

Every data science project needs to have a set of questions to explore.

- How much data preparation is necessary?
- Can you link your results to real-life events?
- Does sentiment change over time?

Basic Terminology

- Corpus
- Stop words
- Tokenization
- Stemming
- Lemmatization
- Term Document Matrix
- Word Cloud

Corpus

In linguistics and NLP, **corpus** (literally Latin for body) refers to a collection of texts. Such collections may be formed of a single language of texts, or can span multiple languages.

Stop words

Stopwords are the words in any language which does not add much meaning to a sentence. "stop words" usually refers to the most common words in a language. There is no universal list of "stop words" that is used by all NLP tools in common.

Tokenization

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords. Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization.

Stemming

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

For grammatical reasons, documents are going to use different forms of a word, such as organize, organizes, and organizing. Additionally, there are families of derivationally related words with similar meanings, such as democracy, democratic, and democratization.

Lemmatization

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .

Lemmatization helps to get to the base form of a word, e.g. are playing $->$ play, eating $->$ eat, etc.

Stemming and Lemmatization are Text Normalization (or sometimes called Word Normalization) techniques in the field of Natural Language Processing that are used to prepare text, words, and documents for further processing.

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance:

$am, are, is -> be$

$car, cars, car's, cars' -> car$

The result of this mapping of text will be something like:

the boy's cars are different colors $->$ *the boy car be differ color*

If confronted with the token *saw*, **stemming** might return just *s*, whereas **lemmatization** would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun.

Term Document Matrix

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms.

For example, let us consider the following example:

- **doc1: "The quick brown fox jumps over the lazy dog."**
- **doc2: "The fox was red but dog was lazy."**
- **doc3: "The dog was brown, and the fox was red."**

Then the term-document matrix can be represented as shown below:

Table 1: Document-term Matrix

	The	quick	brown	fox	jumps	over	lazy	dog	was	red	and
doc1	2	1	1	1	1	1	1	1	0	0	0
doc2	2	0	0	1	0	0	1	1	2	1	1
doc3	2	0	1	1	0	0	0	1	2	1	1

Word Cloud

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

Libraries and Functions

To get started analyzing Hotel Reviews, we need to load the relevant libraries / packages. These may seem daunting at first, but most of them are simply for graphs and charts. Few of the libraries are: *tm*, *stringr*, *rvest*, *SnowballC*, *wordcloud*, *wordcloud2*

```
# Load library
library(tm)
```

```
## Loading required package: NLP
```

```
library(stringr)
library(rvest)
```

```
## Loading required package: xml2
```

```
library(SnowballC)

getwd()
```

```
## [1] "/home/abhay.bhadani/Desktop/TUTORIALS/IMI/Sentiment-Analysis-JW-Marriot"
```

```
setwd("~/Desktop/TUTORIALS/IMI/Sentiment-Analysis-JW-Marriot/")

trip = read.csv("data/tripadvisor_reviews.csv")
```

Including Plots

```
# sentiment, echo=FALSE}
# Sentiment Analysis of Hotel Reviews on Trip Advisor - JW Marriott
hotel_reviews <- as.character(trip$review)

# Load the positive and negative lexicon data and explore
positive_lexicon <- read.csv("data/positive-lexicon.txt")
negative_lexicon <- read.csv("data/negative-lexicon.txt")

# Load the stop words text file and explore
stop_words <- read.csv("data/stopwords_en.txt")
```

VectorSource

Takes a grouping of texts and makes each element of the resulting vector a document within your R Workspace. There are many types of sources, but VectorSource() is made for working with character objects in R.

```
vector_source_reviews = VectorSource(hotel_reviews)
```

Making a corpus out of the hotel reviews Corpora are collections of documents containing (natural language) text. *tm* packages provides the essential infrastructure to carry out sentiment analysis.

Create a corpus and view/inspect the contents after creating the corpus

```
reviews_corpus <- Corpus(vector_source_reviews)
#inspect(reviews_corpus)
```

To see specific review (in this case, we are seeing 2nd row/review), we can use the following command

```
reviews_corpus[[2]]$content
```

```
## [1] "I have never been to a place that so consistently gets it right. Total satisfaction every time
```

```
reviews_corpus[[2]]$meta
```

```
##   author       : character(0)
##   datetimestamp: 2020-08-21 01:03:55
##   description  : character(0)
##   heading      : character(0)
##   id           : 2
##   language     : en
##   origin       : character(0)
```

In this phase, we will remove stop words, punctuations, numbers, special characters from all the reviews before using it for analysis.

```
# Remove stop words
filtered_corpus_no_stopwords <- tm_map(reviews_corpus, removeWords, stopwords('english'))
```

```
## Warning in tm_map.SimpleCorpus(reviews_corpus, removeWords,
## stopwords("english")): transformation drops documents
```

```
#inspect(filtered_corpus_no_stopwords)

# Remove Punctuation
filtered_corpus_no_puncts <- tm_map(filtered_corpus_no_stopwords, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(filtered_corpus_no_stopwords, removePunctuation):
## transformation drops documents
```

```
#inspect(filtered_corpus_no_puncts)

# Remove Numbers
filtered_corpus_no_numbers <- tm_map(filtered_corpus_no_puncts, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(filtered_corpus_no_puncts, removeNumbers):
## transformation drops documents
```

```
#inspect(filtered_corpus_no_numbers)

# Remove unwanted white spaces
filtered_corpus_no_whitespace <- tm_map(filtered_corpus_no_numbers, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(filtered_corpus_no_numbers, stripWhitespace):
## transformation drops documents
```

```
#inspect(filtered_corpus_no_whitespace)

# Make all words to lowercase
filtered_corpus_to_lower <- tm_map(filtered_corpus_no_whitespace, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(filtered_corpus_no_whitespace,
## content_transformer(tolower)): transformation drops documents
```

```
#inspect(filtered_corpus_to_lower)


# Remove stop words of the external file from the corpus and whitespaces again and inspect
stopwords_vec <- as.data.frame(stop_words)
final_corpus_no_stopwords <- tm_map(filtered_corpus_to_lower, removeWords, stopwords_vec[,1])
```

```
## Warning in tm_map.SimpleCorpus(filtered_corpus_to_lower, removeWords,
## stopwords_vec[, : transformation drops documents
```

```
#inspect(final_corpus_no_stopwords)


final_corpus <- tm_map(final_corpus_no_stopwords, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(final_corpus_no_stopwords, stripWhitespace):
## transformation drops documents
```

```
#inspect(final_corpus)
```

Now, we are ready to see the difference in the review after performing the cleaning operation.

```
# Character representation of the corpus of first review
final_corpus[[1]]$content
```

```
## [1] " significant travel challenge day visit hotel hotel handled staff members chris jennifer expecta
```

```
hotel_reviews[1]
```

```
## [1] "I had a significant travel challenge during my 5 day visit to this hotel and the hotel could no
```

Now, we will perform the stemming operation

```
# Stem the words to their root of all reviews present in the corpus
stemmed_corpus <- tm_map(final_corpus, stemDocument)
```

```
## Warning in tm_map.SimpleCorpus(final_corpus, stemDocument): transformation drops
## documents
```

```
stemmed_corpus[[1]]$content
```

```
## [1] "signific travel challeng day visit hotel hotel handl staff member chris jennif expect chris wel
```

Now, we are ready to build a term document matrix of the stemmed corpus

```
TDM_corpus <- TermDocumentMatrix(stemmed_corpus)
# terms occurring with a minimum frequency of 5
findFreqTerms(TDM_corpus, 5)
```

```
##   [1] "common"        "day"           "difficult"     "expect"
##   [5] "handl"         "hotel"         "love"          "member"
```

```
##   [9] "profession"    "rememb"        "situat"        "staff"
##  [13] "time"          "travel"        "visit"         "welcom"
##  [17] "bed"           "clean"         "comfort"       "consist"
##  [21] "extrem"        "good"          "leav"          "marriott"
##  [25] "place"         "rest"          "stay"          "train"
##  [29] "access"        "centr"         "circl"         "cold"
##  [33] "connect"       "convent"       "deal"          "downtown"
##  [37] "easi"          "locat"         "lot"           "mall"
##  [41] "park"          "perfect"       "shop"          "underground"
##  [45] "walkway"       "adjac"         "area"          "arriv"
##  [49] "bar"           "decor"         "friend"        "help"
##  [53] "invit"         "lobbi"         "meet"          "modern"
##  [57] "nice"          "open"          "osteria"       "pronto"
##  [61] "restaur"       "room"          "starbuck"      "stock"
##  [65] "warm"          "citi"          "cocktail"      "dinner"
##  [69] "enter"         "high"          "italian"       "left"
##  [73] "offer"         "recommend"     "sport"         "spot"
##  [77] "view"          "watch"         "amen"          "big"
##  [81] "center"        "championship"  "entir"         "excel"
##  [85] "game"          "indoor"        "luca"          "oil"
##  [89] "short"         "stadium"       "super"         "ten"
##  [93] "walk"          "awesom"        "basebal"       "beauti"
##  [97] "comfi"         "field"         "floor"         "jet"
## [101] "night"         "river"         "shower"        "water"
## [105] "white"         "absolut"       "book"          "check"
## [109] "cost"          "experienc"     "knowledg"      "onsit"
## [113] "previous"      "spacious"      "stun"          "year"
## [117] "bad"           "great"         "happi"         "servic"
## [121] "thing"         "uber"          "central"       "confer"
## [125] "indianapoli"   "larg"          "nearbi"        "pillow"
## [129] "employe"       "food"          "heart"         "indi"
## [133] "ago"           "enjoy"         "featur"        "site"
## [137] "tunnel"        "clerk"         "conveni"       "desk"
## [141] "fair"          "felt"          "late"          "readi"
## [145] "amaz"          "complex"       "full"          "glass"
## [149] "govern"        "issu"          "museum"        "numer"
## [153] "rate"          "valet"         "venu"          "wall"
## [157] "annual"        "courteous"     "distanc"       "eat"
## [161] "experi"        "found"         "interest"      "point"
## [165] "recent"        "anniversari"   "card"          "care"
## [169] "checkin"       "deliv"         "feel"          "getaway"
## [173] "hand"          "weekend"       "breakfast"     "concierg"
## [177] "elev"          "long"          "pricey"        "wait"
## [181] "attend"        "back"          "fine"          "trip"
## [185] "appreci"       "build"         "face"          "loung"
## [189] "side"          "west"          "architectur"   "busi"
## [193] "chain"         "impecc"        "run"           "simpli"
## [197] "upscal"        "free"          "ice"           "machin"
## [201] "need"          "peopl"         "except"        "minut"
## [205] "receiv"        "told"          "choos"         "extra"
## [209] "held"          "husband"       "marriot"       "paid"
## [213] "reward"        "star"          "come"          "favorit"
## [217] "past"          "quiet"         "cheaper"       "dessert"
## [221] "elit"          "event"         "fee"           "garag"
```

```
## [225] "list"          "snack"         "street"        "concern"
## [229] "front"         "guest"         "spent"         "wonder"
## [233] "bit"           "charg"         "selfpark"      "atmospher"
## [237] "direct"        "facil"         "live"          "make"
## [241] "multipl"       "safe"          "smaller"       "attent"
## [245] "key"           "general"       "manag"         "nikki"
## [249] "top"           "veloc"         "detail"        "ive"
## [253] "marathon"      "towel"         "ceil"          "fit"
## [257] "furnish"       "huge"          "lack"          "problem"
## [261] "updat"         "upgrad"        "window"        "easili"
## [265] "organ"         "requir"        "skywalk"       "bell"
## [269] "club"          "custom"        "execut"        "housekeep"
## [273] "includ"        "jws"           "nicer"         "stand"
## [277] "turn"          "basic"         "decent"        "devic"
## [281] "note"          "return"        "show"          "wow"
## [285] "end"           "indiana"       "level"         "maintain"
## [289] "newer"         "pool"          "properti"      "work"
## [293] "call"          "overbook"      "standard"      "week"
## [297] "impress"       "intern"        "quick"         "space"
## [301] "drink"         "group"         "loud"          "serv"
## [305] "cater"         "job"           "negat"         "thought"
## [309] "coffe"         "phone"         "pleasant"      "pretti"
## [313] "futur"         "nicest"        "offic"         "reserv"
## [317] "road"          "surpris"       "tire"          "dine"
## [321] "poor"          "terrif"        "gym"           "plenti"
## [325] "superb"        "appoint"       "expens"        "luxuri"
## [329] "car"           "close"         "decemb"        "smooth"
## [333] "zoo"           "clear"         "system"        "block"
## [337] "layout"        "station"       "wifi"          "correct"
## [341] "gave"          "made"          "price"         "appear"
## [345] "bathroom"      "immacul"       "option"        "buffet"
## [349] "meal"          "part"          "class"         "kind"
## [353] "complaint"     "equip"         "exercis"       "gift"
## [357] "ncaa"          "corner"        "fantast"       "head"
## [361] "order"         "small"         "soft"          "type"
## [365] "confus"        "footbal"       "major"         "airport"
## [369] "forward"       "outstand"      "pleasur"       "home"
## [373] "sit"           "gorgeous"      "knew"          "singl"
## [377] "door"          "octob"         "person"        "question"
## [381] "ride"          "shuttl"        "midwest"       "notch"
## [385] "assist"        "advanc"        "find"          "wed"
## [389] "attach"        "compar"        "headquart"     "number"
## [393] "capitol"       "mattress"      "state"         "folk"
## [397] "disappoint"    "met"           "pull"          "afternoon"
## [401] "ate"           "bartend"       "eric"          "lunch"
## [405] "put"           "sunday"        "want"          "interior"
## [409] "fast"          "internet"      "process"       "fail"
## [413] "host"          "regular"       "ask"           "greet"
## [417] "select"        "upper"         "relax"         "ballroom"
## [421] "delici"        "incred"        "colt"          "cheap"
## [425] "reason"        "town"          "victori"       "request"
## [429] "horribl"       "ill"           "moment"        "smile"
## [433] "talk"          "hospit"        "mile"          "spectacular"
## [437] "attitud"       "encount"       "start"         "qualiti"
```

```
## [441] "even"          "look"          "minor"         "stop"
## [445] "strong"        "women"         "young"         "fabul"
## [449] "team"          "workout"       "compani"       "famili"
## [453] "plan"          "built"         "wife"          "worth"
## [457] "bellman"       "escort"        "luggag"        "overwhelm"
## [461] "size"          "date"          "septemb"       "local"
## [465] "varieti"       "refriger"      "pleas"         "saturday"
## [469] "typic"         "line"          "microwav"      "checkout"
## [473] "drive"         "hour"          "accommod"      "touch"
## [477] "world"         "compet"        "morn"          "review"
## [481] "light"         "mention"       "hous"          "menu"
## [485] "overlook"      "doubl"         "fact"          "fresh"
## [489] "celebr"        "decid"         "packag"        "purchas"
## [493] "prompt"        "race"          "coupl"         "holiday"
## [497] "juli"          "platinum"      "cleanli"       "pay"
## [501] "set"           "higher"        "lower"         "special"
## [505] "choic"         "canal"         "indian"        "chicken"
## [509] "downstair"     "averag"        "heard"         "larger"
## [513] "middl"         "nois"          "phenomen"      "comment"
## [517] "take"          "ideal"         "import"        "ball"
## [521] "lost"          "hot"           "respons"       "suit"
## [525] "swim"          "tub"           "chang"         "cheer"
## [529] "overnight"     "concert"       "spend"         "staf"
## [533] "august"        "brought"       "children"      "earli"
## [537] "friday"        "ladi"          "king"          "gencon"
## [541] "bring"         "hear"          "attract"       "queen"
## [545] "adequ"         "provid"        "personnel"     "spa"
## [549] "effici"        "suppos"        "weather"       "insid"
## [553] "beat"          "add"           "cover"         "secur"
## [557] "brand"         "chose"         "polit"         "complimentari"
## [561] "occas"         "wine"          "boy"           "kid"
## [565] "nation"        "temperatur"    "hall"          "server"
## [569] "tabl"          "credit"        "move"          "bridg"
## [573] "sky"           "respect"       "share"         "slept"
## [577] "skylin"        "basketbal"     "sceneri"       "play"
## [581] "slow"          "inform"        "due"           "frequent"
## [585] "fun"           "parti"         "main"          "treat"
## [589] "june"          "appet"         "glad"          "tast"
## [593] "volleybal"     "cool"          "immedi"        "resolv"
## [597] "speed"         "iron"          "miss"          "confirm"
## [601] "email"         "leagu"         "valentin"      "maker"
## [605] "sleep"         "write"         "exceed"        "daughter"
## [609] "prior"         "pack"          "hor"           "march"
## [613] "counter"       "aaa"           "girl"          "daili"
## [617] "addit"         "bill"          "final"         "condit"
## [621] "understand"    "excit"         "son"           "cozi"
## [625] "begin"         "mini"          "seat"          "contact"
## [629] "surround"      "crowd"         "adult"         "roomi"
## [633] "matter"        "manner"        "registr"       "arrang"
## [637] "birthday"      "chicago"       "activ"         "vega"
## [641] "courtyard"     "competit"      "complet"       "happen"
## [645] "month"         "didnt"         "give"          "design"
## [649] "edg"           "posit"         "hope"          "recept"
## [653] "air"           "fridg"         "hard"          "normal"
```

```
## [657] "rare"          "item"          "associ"        "vacat"
## [661] "seamless"      "opinion"       "differ"        "trail"
## [665] "trade"         "blue"          "colleagu"      "suggest"
## [669] "origin"        "guarante"      "store"         "christma"
```

From this step onwards, we are going to compute some simple metrics, such as total positive word count, total of negative word count, percentage, and so on.

```
total_pos_count <- 0
total_neg_count <- 0
pos_count_vector <- c()
neg_count_vector <- c()

size <- length(stemmed_corpus)

for(i in 1:size){
  corpus_words<- list(strsplit(stemmed_corpus[[i]]$content, split = " "))
  #print(intersect(unlist(corpus_words), unlist(positive_lexicon))) ## positive words in current review
  pos_count <- length(intersect(unlist(corpus_words), unlist(positive_lexicon)))
  #print(intersect(unlist(corpus_words), unlist(negative_lexicon))) ## negative words in current review
  neg_count <- length(intersect(unlist(corpus_words), unlist(negative_lexicon)))
  if(pos_count>neg_count){
    #print("It's a positive review")
  } else{
    #print("It's a negative review")
  }
  total_count_for_current_review <- pos_count + neg_count ## current positive and negative count
  pos_percentage <- (pos_count*100)/total_count_for_current_review
  neg_percentage <- (neg_count*100)/total_count_for_current_review
  #print(pos_percentage)                          ## current positive percentage
  #print(neg_percentage)                          ## current negtive percentage
  total_pos_count <- total_pos_count + pos_count ## overall positive count
  total_neg_count <- total_neg_count + neg_count ## overall negative count
  pos_count_vector <- append(pos_count_vector, pos_count)
  neg_count_vector <- append(neg_count_vector, neg_count)
}

print(pos_percentage)                            ## current positive percentage
```

```
## [1] 71.42857
```

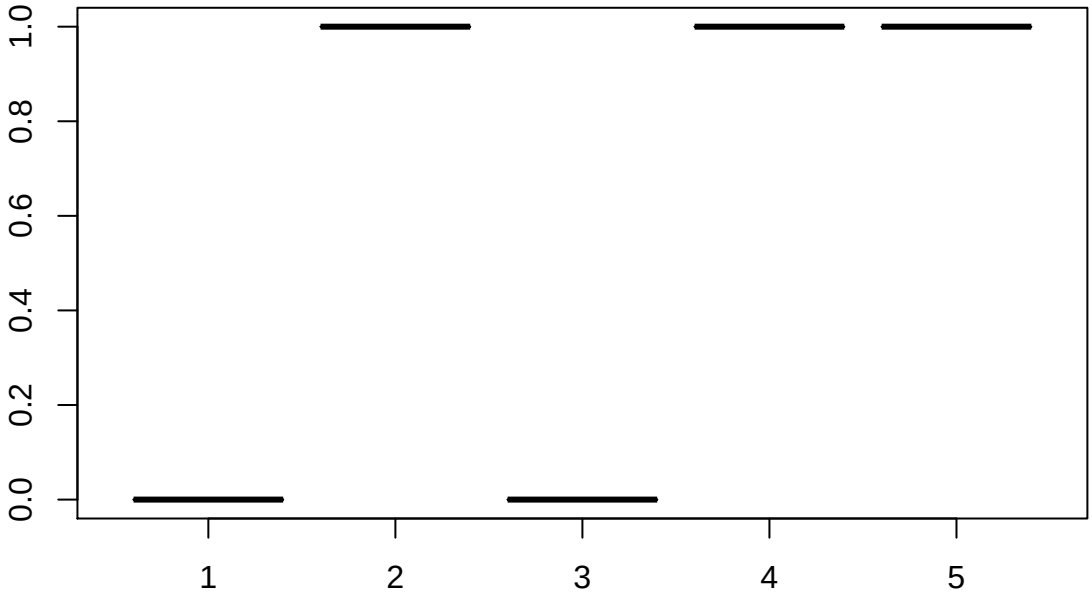
```
print(neg_percentage)                            ## current negtive percentage
```

```
## [1] 28.57143
```

Create a dataframe where number of positive words and negative words are stored against each review.

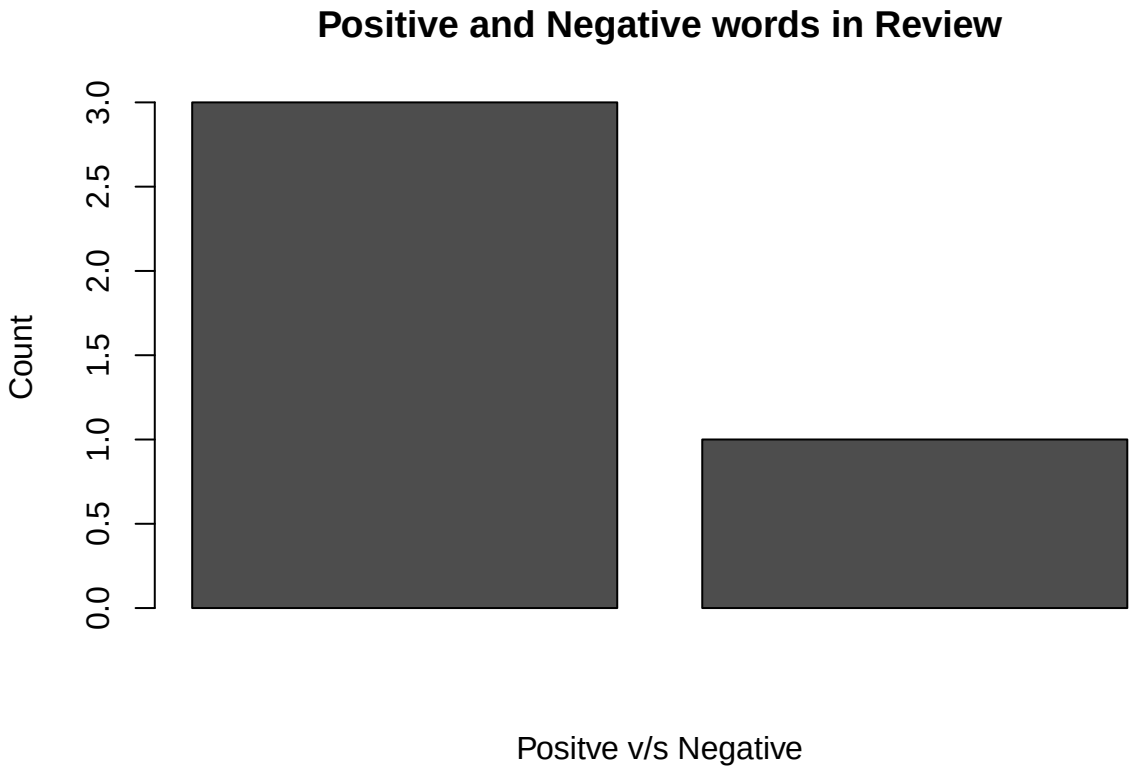
```
# Sentiment score of each review and visualizing using boxplot
counts <- data.frame(pos_count_vector, neg_count_vector)

sentiment <- data.frame(c(1:size),(pos_count_vector - neg_count_vector) / (pos_count_vector + neg_count_
names(sentiment)=c('review_id','SentimentScore')
boxplot(sentiment$SentimentScore[0:5]~sentiment$review_id[0:5])
```




```
# Visualiztion of positive and negative count of single review
singe_review <- c(counts$pos_count_vector[8], counts$neg_count_vector[8])

barplot(t(as.data.frame(singe_review)), ylab = "Count", xlab = "Positve v/s Negative",  main = "Positive
```




```
# Calculating overall percentage of positive and negative words of all the reviews
total_pos_count                                    ## overall positive count
```

```
## [1] 2449
```

```
total_neg_count                                    ## overall negative count
```

```
## [1] 365
```

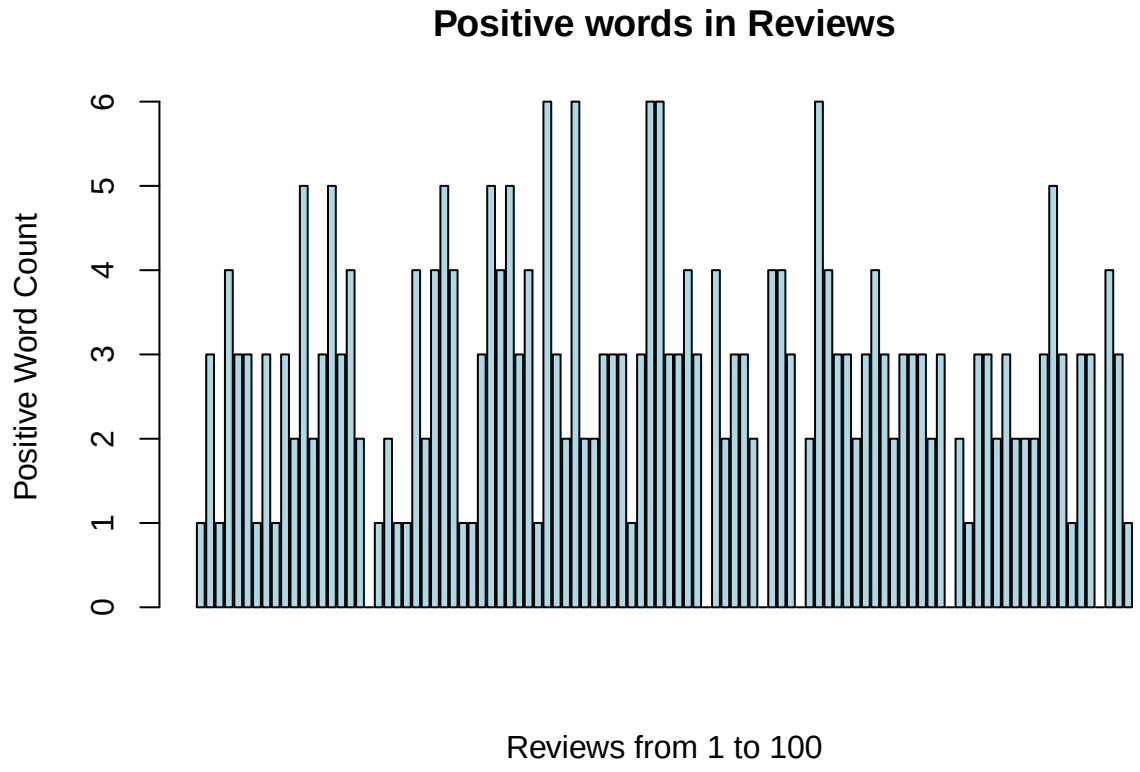
```
total_count <- total_pos_count + total_neg_count
overall_positive_percentage <- (total_pos_count*100)/total_count
overall_negative_percentage <- (total_neg_count*100)/total_count
overall_positive_percentage                      ## overall positive percentage
```

```
## [1] 87.02914
```

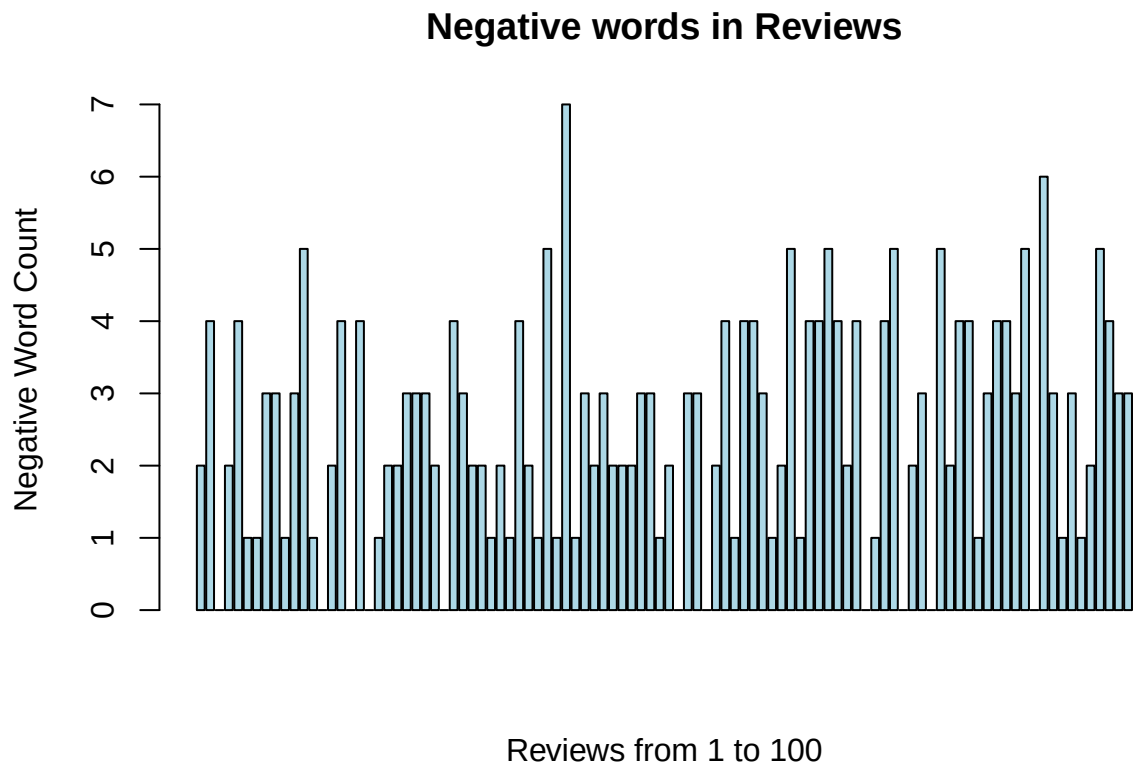
```
overall_negative_percentage                      ## overall negative percentage
```

```
## [1] 12.97086
```

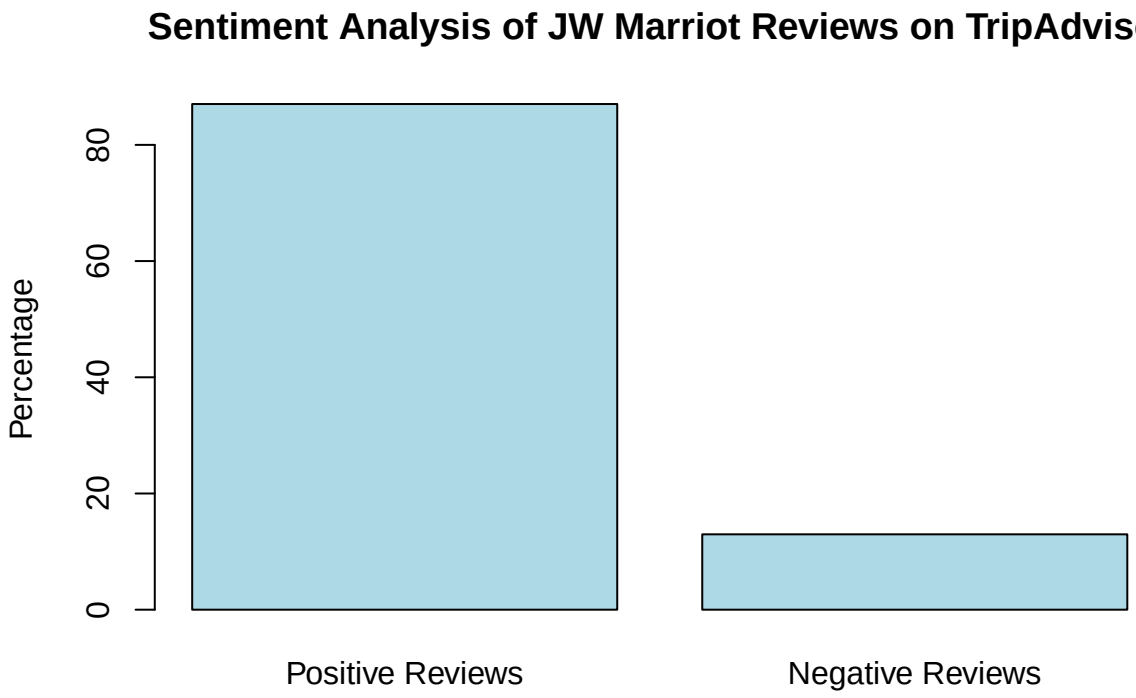
```
# Visualization of positive and negative word count for all the reviews
review_count_frame <- data.frame(matrix(c(pos_count_vector, neg_count_vector), nrow = 100, ncol = 2))
colnames(review_count_frame) <- c("Positive Word Count", "Negative Word Count")
barplot(review_count_frame$`Positive Word Count`, ylab = "Positive Word Count", xlab = "Reviews from 1 t
```




```
barplot(review_count_frame$`Negative Word Count`, ylab = "Negative Word Count", xlab = "Reviews from 1 t
```




```r
# Visualization of Overall positive and negative reviews
percent_vec <- c(overall_positive_percentage, overall_negative_percentage)
percent_frame <- as.data.frame(percent_vec)
rownames(percent_frame) <- c("Positive Reviews","Negative Reviews")
colnames(percent_frame) <- c("Percentage")
percentage <- t(percent_frame)
barplot(percentage, ylab = "Percentage", main = "Sentiment Analysis of JW Marriot Reviews on TripAdviso
```



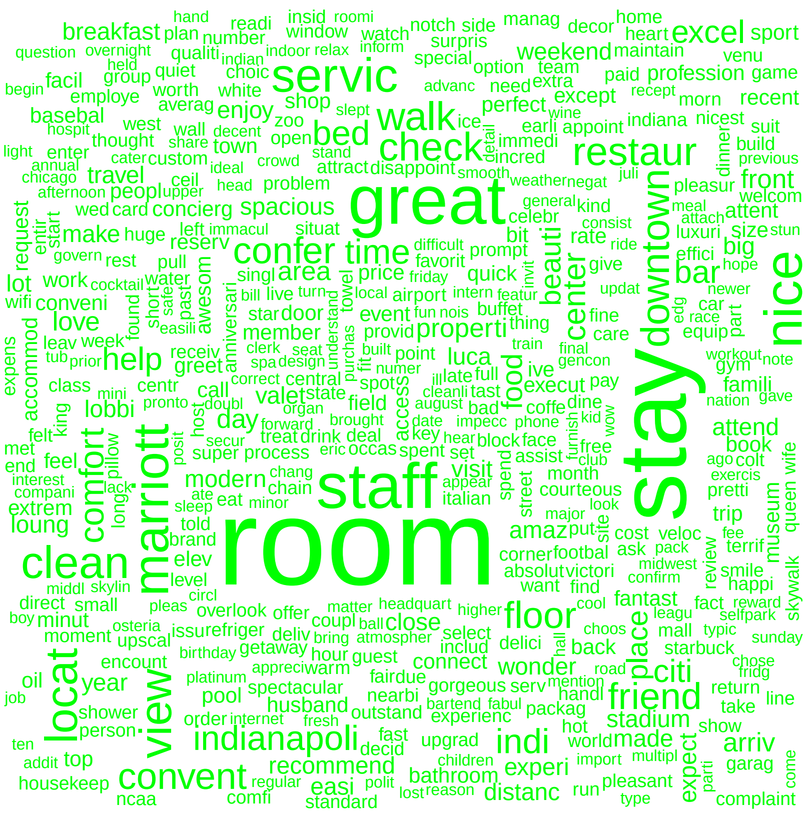
Visualizing the words used in reviews

```r
library(wordcloud)
```

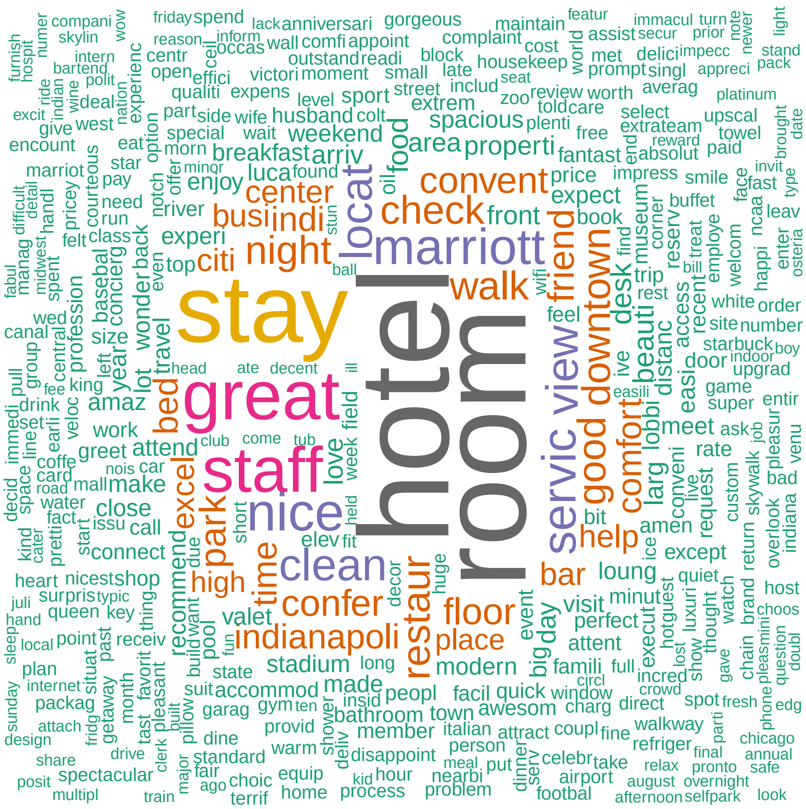
```
## Loading required package: RColorBrewer
```

```r
set.seed(1234) # for reproducibility
textM = as.matrix(TDM_corpus)
#textM[0:10]
textFreqs <- rowSums(textM)
freqTab <- data.frame(term = names(textFreqs), num = textFreqs)
write.csv(freqTab, "freq.csv", row.names = FALSE)
```

```r
wordcloud(freqTab$term, freqTab$num, max.words = 500, color = "green")
```




```r
wordcloud(words = freqTab$term, freq = freqTab$num, min.freq = 4,
          max.words=500, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```




```r
freqTab = read.csv('freq.csv', stringsAsFactors = FALSE)
dim(freqTab)
```

```
## [1] 2558    2
```

```r
head(freqTab)
```

```
##        term num
## 1  challeng   4
## 2     chris   3
## 3    common   5
## 4       day  98
## 5 difficult  11
## 6    expect  71
```

```r
library(wordcloud2)
#wordcloud2(data=freqTab)
wordcloud2(data=freqTab[0:100,], size=1.6, color='random-dark')
```

```r
wordcloud2(data=freqTab[100:200,], size = 0.7, shape = 'pentagon')
```

Reasons Why You Should Use Word Clouds

Word clouds are killer visualisation tools. They present text data in a simple and clear format, that of a cloud in which the size of the words depends on their respective frequencies. As such, they are visually nice to look at as well as easy and quick to understand.

Word clouds are great communication tools. They are incredibly handy for anyone wishing to communicate a basic insight based on text data — whether it's to analyse a speech, capture the conversation on social media

or report on customer reviews. Word clouds are insightful. Visually engaging, word clouds allow us to draw several insights quickly, allowing for some flexibility in their interpretation. Their visual format stimulates us to think and draw the best insight depending on what we wish to analyse.

When Should I Use Word Clouds?

Yes, word clouds are insightful, great communication and visualisation tools. Nonetheless, they also have their limits and understanding this is key to know when to use them and therefore also when not to. Word clouds are essentially a descriptive tool. They should therefore only be used to capture basic qualitative insights. Visually attractive, they are a great tool to start a conversation, a presentation or an analysis. However, their analysis is limited to insights that simply don't have the same calibre as more extensive statistical analysis.

Conclusion

Sentiment analysis is a very useful analysis. It carries huge value for the business houses as it can tell what their customers are talking about in public platform. We can deep dive into the reviews and start analyzing the exact reason for such reviews.