> cname <- file.path(".", "Text Mining", "Documents")

> cname

[1] "./Text Mining/Documents"

> length(dir(cname))

[1] 1

> dir(cname)

[1] "Treasure Island.txt"

> docs <- Corpus(DirSource(cname))

> docs

<<SimpleCorpus>>

Metadata: corpus specific: 1, document level (indexed): 0

Content: documents: 1

> inspect(docs[1])

<<SimpleCorpus>>

Metadata: corpus specific: 1, document level (indexed): 0

Content: documents: 1

... <truncated>

The Project Gutenberg EBook of Treasure Island, by Robert Louis Stevenson\n\nThis eBook is for the use of anyone anywhere at no cost and with\nalmost no restrictions whatsoever. You may copy it, give it away or\nre-use it under the terms of the Project Gutenberg License included\nwith this eBook or online at www.gutenberg.org\n\n\nTitle: Treasure Island\n\nAuthor: Robert Louis Stevenson\n\nRelease Date: February 25, 2006 [EBook #120]\n[Last Updated: November 10, 2010]\n\nLanguage: English\n\n\n\*\*\* START OF THIS PROJECT GUTENBERG EBOOK TREASURE ISLAND \*\*\*\n\n\n\n\nProduced by Judy Boss, John Hamm and David Widger\n\n\n\n\n\nTREASURE ISLAND\n\nby Robert Louis Stevenson\n\n\n\n\nTREASURE ISLAND\n\nTo S.L.O., an American gentleman in accordance with whose classic taste\nthe following narrative has been designed, it is now, in return for\nnumerous delightful hours, and with the kindest wishes, dedicated by his\naffectionate friend, the author.\n\n\n\n TO THE HESITATING PURCHA... <truncated>

> #Text preprocessing

> #Simple transformation

> toSpace <- content\_transformer(function(x, pattern) gsub(pattern, " ", x))

> docs <- tm\_map(docs, toSpace, "/")

> docs <- tm\_map(docs, toSpace, "@")

> docs <- tm\_map(docs, toSpace, "nn|")

> #Converting to lower cases

> docs <- tm\_map(docs, content\_transformer(tolower))

> #Removing number

> docs <- tm\_map(docs, removeNumbers)

> #Removing punctuations

> docs <- tm\_map(docs, removePunctuation)

> #Removing English stop words

> docs <- tm\_map(docs, removeWords, stopwords("english"))

> #English stop words

> length(stopwords("english"))

[1] 174

> stopwords("english")

[1] "i" "me" "my" "myself" "we" "our" "ours" "ourselves" "you"

[10] "your" "yours" "yourself" "yourselves" "he" "him" "his" "himself" "she"

[19] "her" "hers" "herself" "it" "its" "itself" "they" "them" "their"

[28] "theirs" "themselves" "what" "which" "who" "whom" "this" "that" "these"

[37] "those" "am" "is" "are" "was" "were" "be" "been" "being"

[46] "have" "has" "had" "having" "do" "does" "did" "doing" "would"

[55] "should" "could" "ought" "i'm" "you're" "he's" "she's" "it's" "we're"

[64] "they're" "i've" "you've" "we've" "they've" "i'd" "you'd" "he'd" "she'd"

[73] "we'd" "they'd" "i'll" "you'll" "he'll" "she'll" "we'll" "they'll" "isn't"

[82] "aren't" "wasn't" "weren't" "hasn't" "haven't" "hadn't" "doesn't" "don't" "didn't"

[91] "won't" "wouldn't" "shan't" "shouldn't" "can't" "cannot" "couldn't" "mustn't" "let's"

[100] "that's" "who's" "what's" "here's" "there's" "when's" "where's" "why's" "how's"

[109] "a" "an" "the" "and" "but" "if" "or" "because" "as"

[118] "until" "while" "of" "at" "by" "for" "with" "about" "against"

[127] "between" "into" "through" "during" "before" "after" "above" "below" "to"

[136] "from" "up" "down" "in" "out" "on" "off" "over" "under"

[145] "again" "further" "then" "once" "here" "there" "when" "where" "why"

[154] "how" "all" "any" "both" "each" "few" "more" "most" "other"

[163] "some" "such" "no" "nor" "not" "only" "own" "same" "so"

[172] "than" "too" "very"

> #Creating a Document Term Matrix

> dtm <- DocumentTermMatrix(docs)

> dtm

<<DocumentTermMatrix (documents: 1, terms: 6696)>>

Non-/sparse entries: 6696/0

Sparsity : 0%

Maximal term length: 31

Weighting : term frequency (tf)

> inspect(dtm)

<<DocumentTermMatrix (documents: 1, terms: 6696)>>

Non-/sparse entries: 6696/0

Sparsity : 0%

Maximal term length: 31

Weighting : term frequency (tf)

Sample :

Terms

Docs captain doctor like man now one said silver upon well

Treasure Island.txt 209 162 204 234 268 266 322 202 153 185

> class(dtm)

[1] "DocumentTermMatrix" "simple\_triplet\_matrix"

> dim(dtm)

[1] 1 6696

> tdm <- TermDocumentMatrix(docs)

> tdm

<<TermDocumentMatrix (terms: 6696, documents: 1)>>

Non-/sparse entries: 6696/0

Sparsity : 0%

Maximal term length: 31

Weighting : term frequency (tf)

> #Exploring frequency term document

> freq <- colSums(as.matrix(dtm))

> length(freq)

[1] 6696

> #Least frequent terms

> ord <- order(freq)

> freq[head(ord)]

abandon abe abeam abide ablaze ablowing

1 1 1 1 1 1

> # Most frequent terms.

> freq[tail(ord)]

like captain man one now said

204 209 234 266 268 322

> #Identifying Frequent Items and

> findFreqTerms(dtm, lowfreq=100)

[1] "back" "captain" "come" "cried" "doctor" "good" "hand" "hands" "ill" "last" "like" "little"

[13] "long" "man" "now" "old" "one" "said" "see" "silver" "sir" "still" "time" "two"

[25] "upon" "well"

> #Word frequency

> freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)

> head(freq, 10)

said now one man captain like silver well doctor upon

322 268 266 234 209 204 202 185 162 153

> wf <- data.frame(word=names(freq), freq=freq)

> head(wf)

word freq

said said 322

now now 268

one one 266

man man 234

captain captain 209

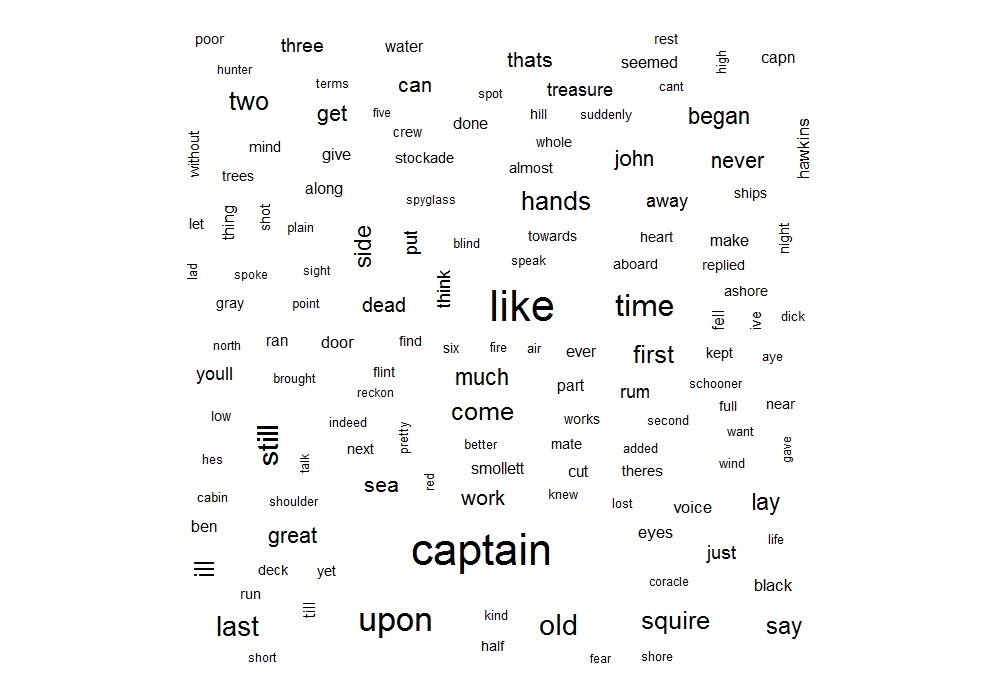
like like 204

> #Word cloud

> library(wordcloud)

> set.seed(123)

> wordcloud(names(freq), freq, min.freq=25)



> #Word cloud with colors and fewer terms

> wordcloud(names(freq), freq, min.freq=50, colors=brewer.pal(6, "Dark2"))

