Assigment 2

Pavel Linder, Nikita Brancatisano

12/26/2019

0. Read input

```
train = read.table(file = 'train.tsv', sep = '\t', header = TRUE, stringsAsFactors = FALSE)
test = read.table(file = 'test.tsv', sep = '\t', header = TRUE)
length(which(!complete.cases(train)))
```

```
## [1] 0
```

```
train$text_a[1:3]
```

```
## [1] "Xanax was her death blow. \\\\xc2\\\\xa0That stuff is totally dangerous because you
## [2] "you are both morons and that is never happening"
## [3] "you are just an idiot blabbermouth that is gonna get stopped HARD one day! You W
```

1. Cleaning data

Remove punctuation and stopwords (?TODO: tolower)

```
train$text_a = as.character(train$text_a)
train$text_a = tm::removePunctuation(train$text_a)
train$text_a = tm::removeWords(x = train$text_a, stopwords(kind = "SMART"))
train$text_a = tm::stripWhitespace(train$text_a)
train$text_a[1:3]
```

```
## [1] "Xanax death blow xc2xa0That stuff totally dangerous build tolerance quickly stop abruptly xc2xa
## [2] " morons happening"
## [3] " idiot blabbermouth gonna stopped HARD day You WILL NOT saved"
```

Anonymize proper nouns

Remove unknown symbols (non UTF-8 characters)

```
train$text_a <- iconv(train$text_a, to='UTF-8', sub='byte')
train$text_a[1:3]
```

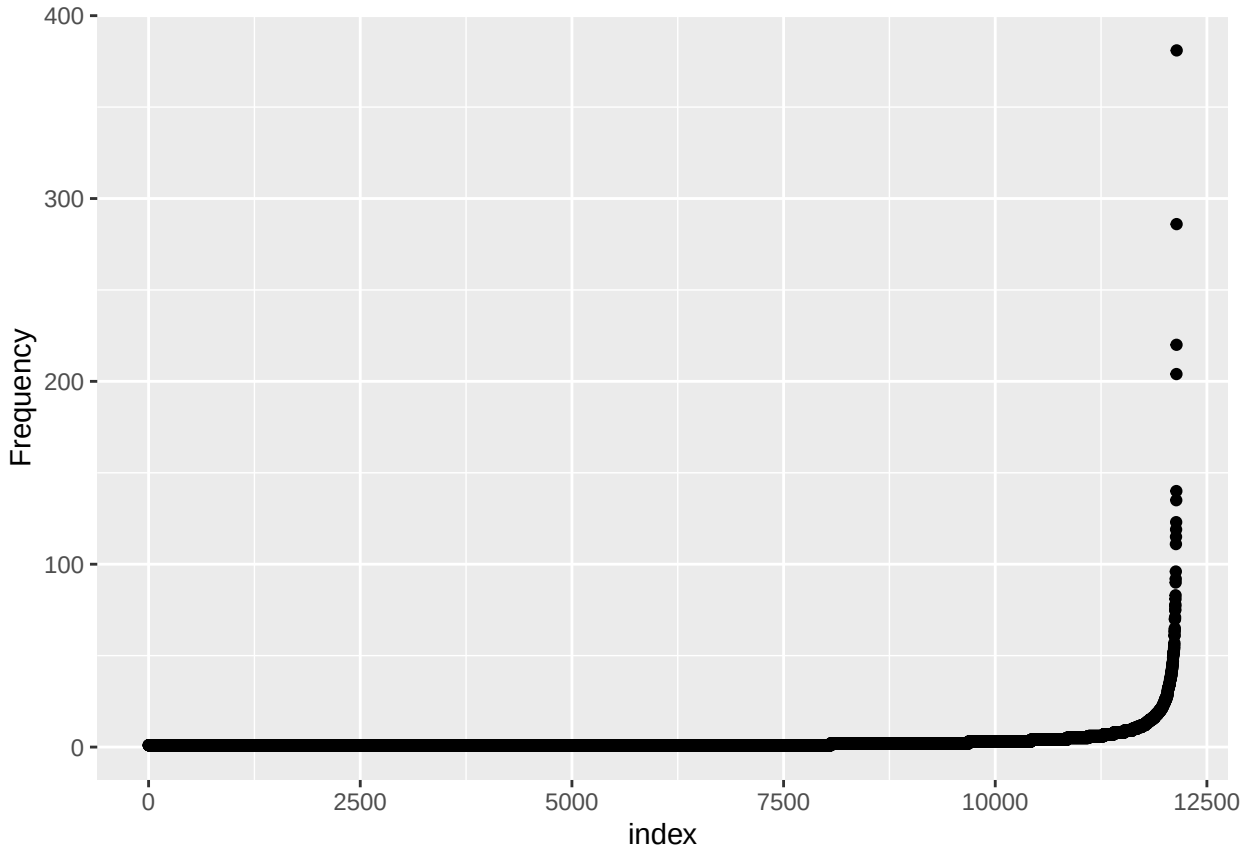
```
## [1] "Xanax death blow xc2xa0That stuff totally dangerous build tolerance quickly stop abruptly xc2xa
## [2] " morons happening"
## [3] " idiot blabbermouth gonna stopped HARD day You WILL NOT saved"
```

2. Exploration

I. Plot the frequency of words (without stemmization)

```
corpus <- Corpus(VectorSource(train$text_a)) # turn into corpus
tdm <- TermDocumentMatrix(corpus)
```

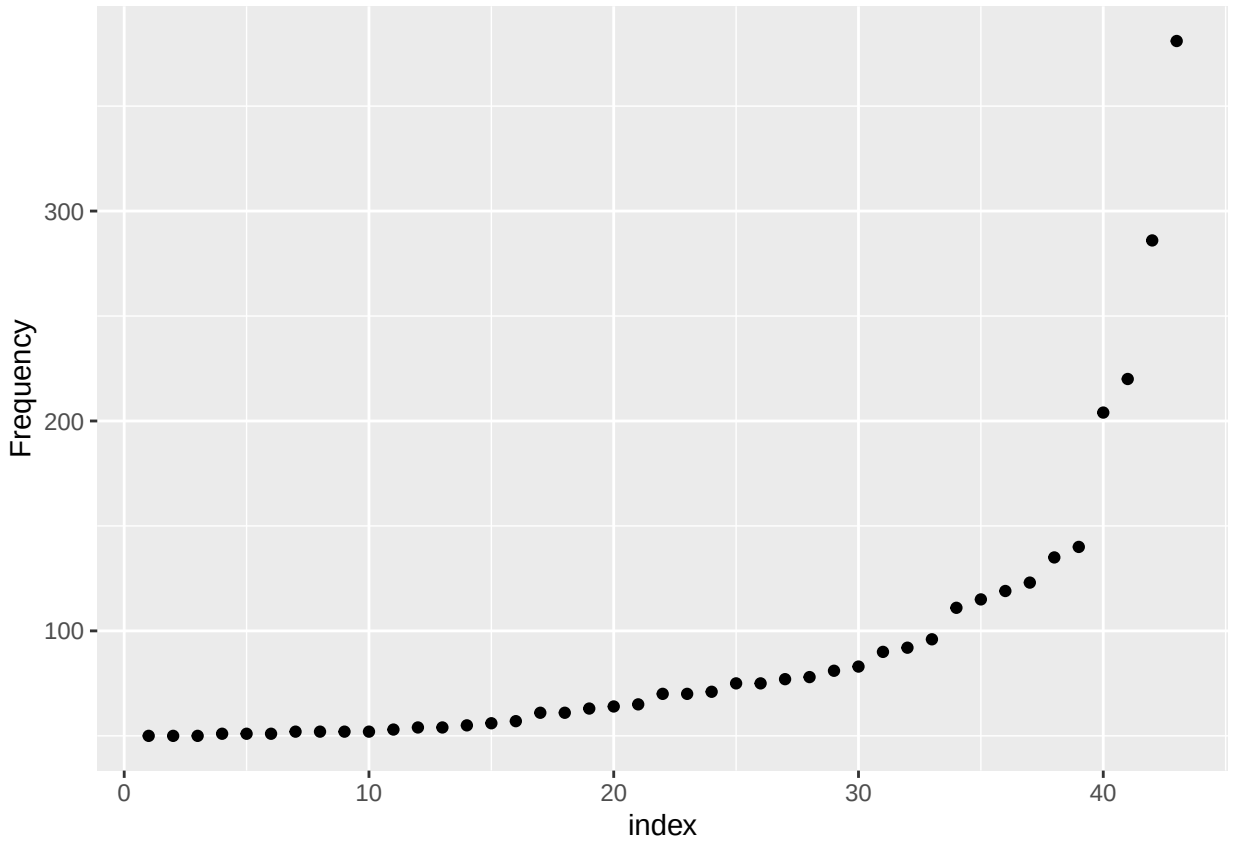
```
wordFreq <- sort(rowSums(as.matrix(tdm)), decreasing=TRUE)
qplot(seq(length(wordFreq)),sort(wordFreq), xlab = "index", ylab = "Frequency")
```




```
findFreqTerms(tdm, lowfreq=50)
```

```
##  [1] "big"      "didnt"    "dont"     "stop"     "day"      "idiot"
##  [7] "you"      "love"     "stupid"   "the"      "things"   "shit"
## [13] "fuck"     "thing"    "and"      "time"     "good"     "people"
## [19] "that"     "they"     "gay"      "white"    "man"      "doesnt"
## [25] "make"     "feel"     "all"      "fucking"  "what"     "ass"
## [31] "bitch"    "back"     "its"      "life"     "money"    "obama"
## [37] "post"     "this"     "world"    "years"    "your"     "youre"
## [43] "democrat"
```

```
mostFreq <- subset(wordFreq, wordFreq >= 50)
qplot(seq(length(mostFreq)),sort(mostFreq), xlab = "index", ylab = "Frequency")
```




```
length(wordFreq)
```

```
## [1] 12143
```

```
length(wordFreq[wordFreq<10])
```

```
## [1] 11618
```

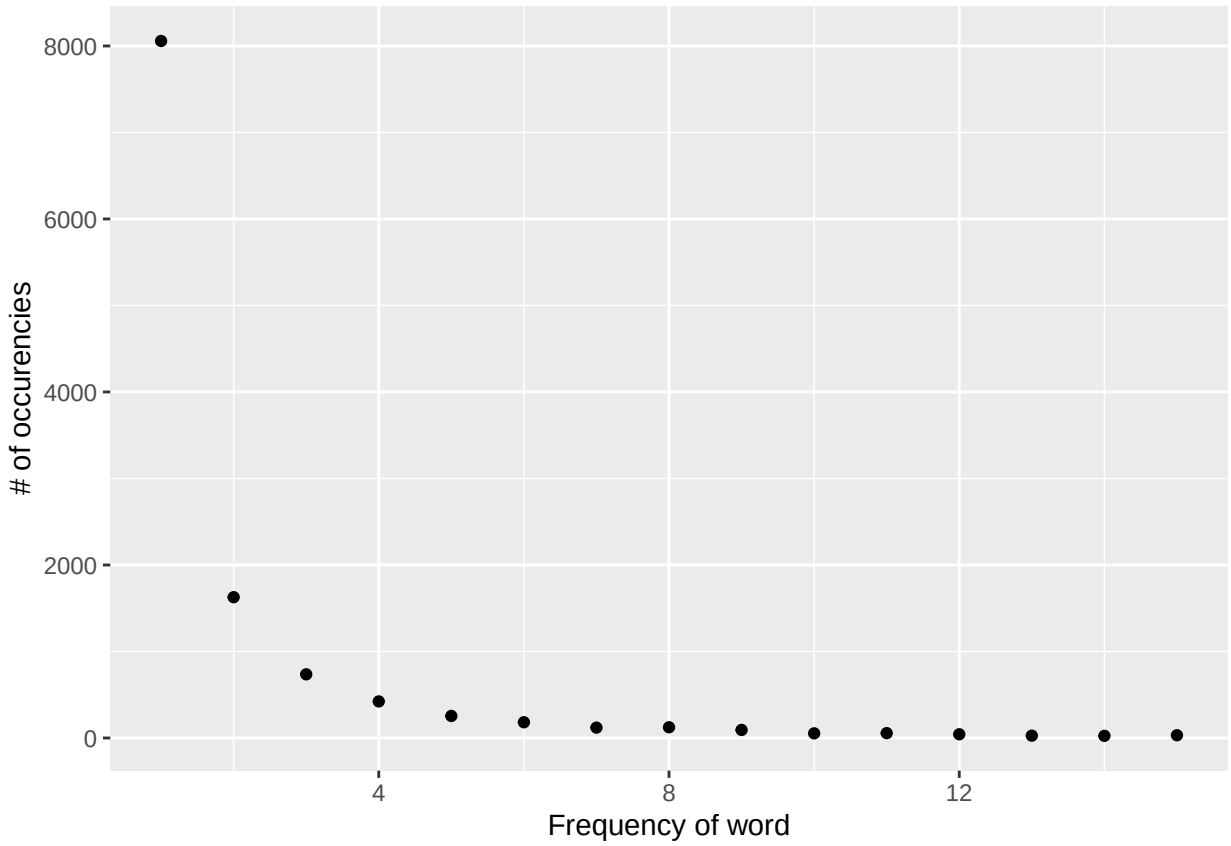
```
length(wordFreq[wordFreq<5])
```

```
## [1] 10844
```

```
length(wordFreq[wordFreq==1])
```

```
## [1] 8057
```

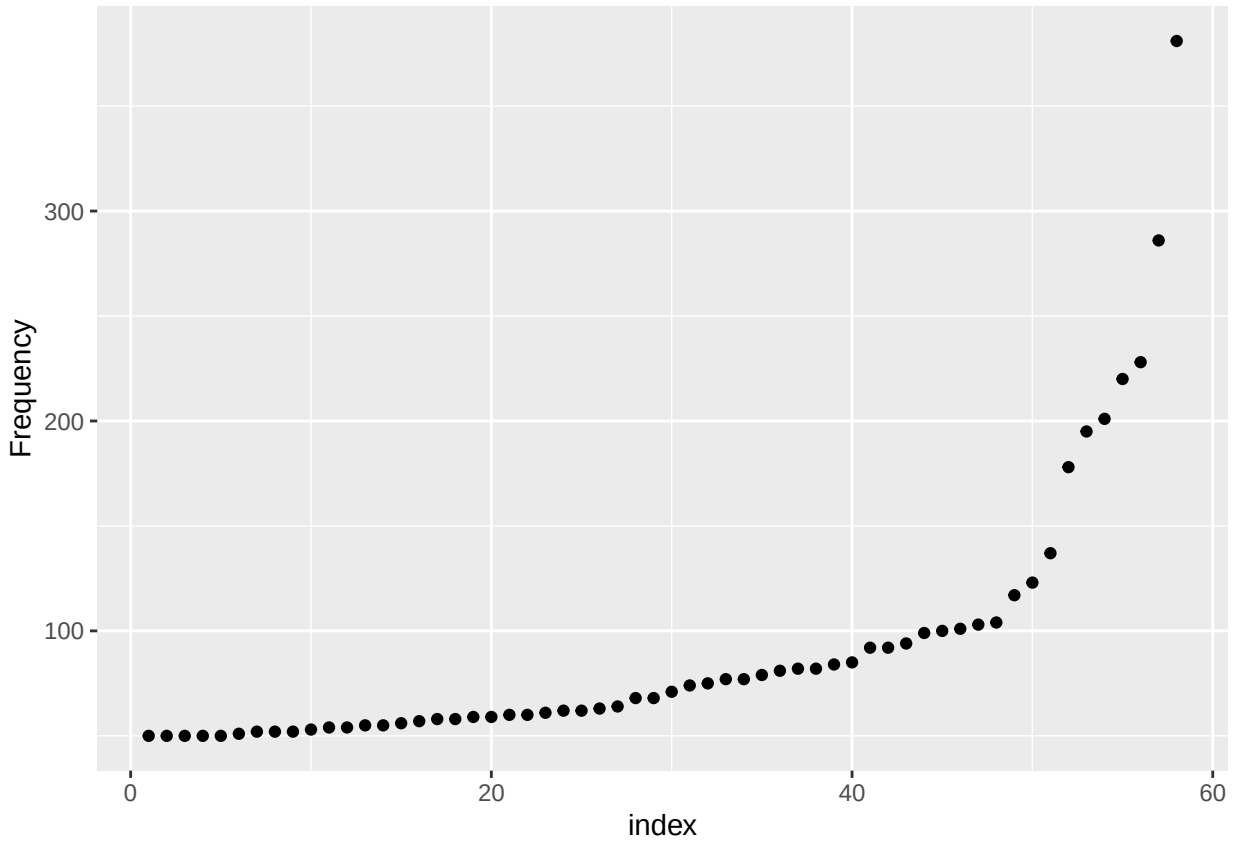
```
freq <- sort(unique(wordFreq), decreasing=FALSE)
occ <- vector()
for (i in 1:length(freq)) {
  occ[i] <- length(wordFreq[wordFreq == freq[i]])
}
qplot(freq[1:15], occ[1:15], xlab = "Frequency of word", ylab = "# of occurencies")
```



I. Plot the frequency of words (with stemmization)

```
stemmed <- stemDocument(train$text_a, language = "english")
corpus2 <- Corpus(VectorSource(stemmed)) # turn into corpus
```

```
qplot(seq(length(mostFreq)),sort(mostFreq), xlab = "index", ylab = "Frequency")
```




```
length(wordFreq)
```

```
## [1] 10124
```

```
length(wordFreq[wordFreq<10])
```

```
## [1] 9497
```

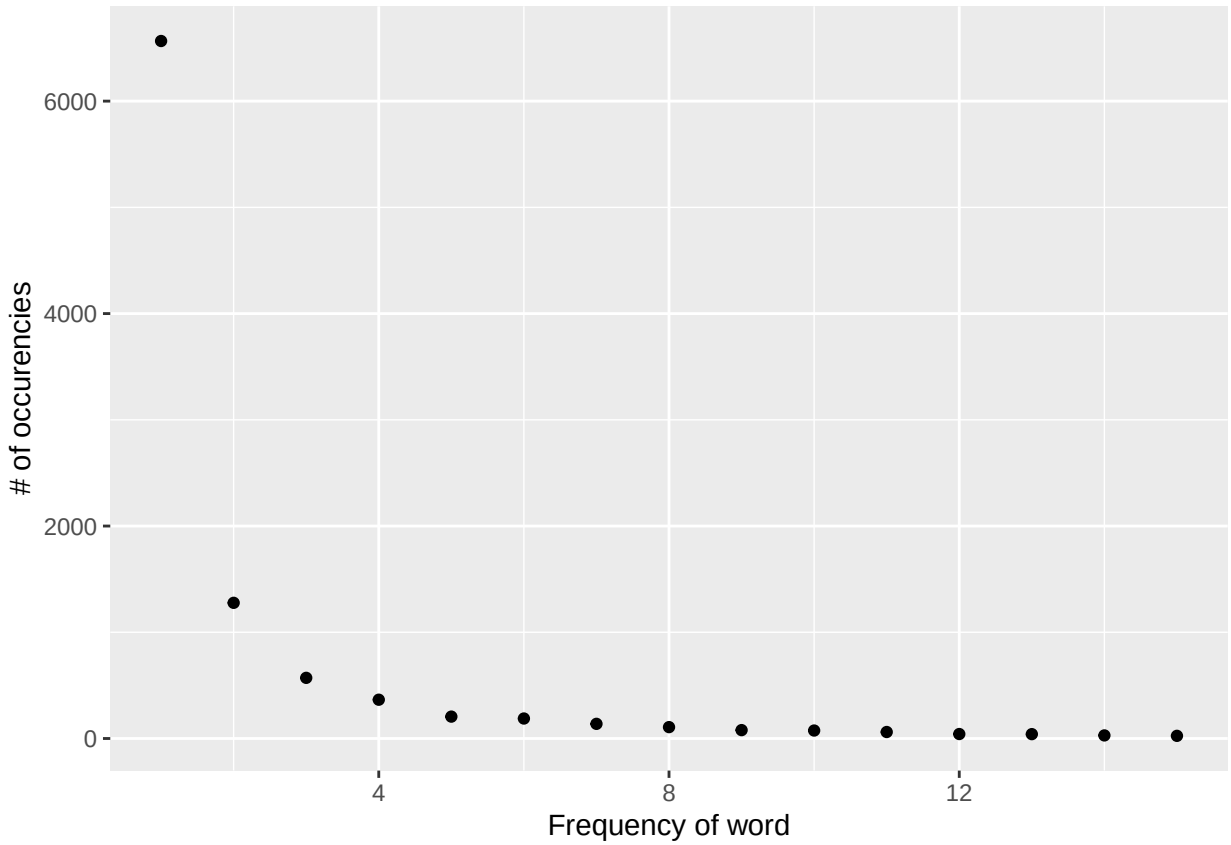
```
length(wordFreq[wordFreq<5])
```

```
## [1] 8779
```

```
length(wordFreq[wordFreq==1])
```

```
## [1] 6566
```

```
qplot(freq[1:15], occ[1:15], xlab = "Frequency of word", ylab = "# of occurencies")
```



II. Perform a clustering on the vectorized document space

We will use Weighted TF-IDF as a way to represent the document space:

```
tdm <- tm::DocumentTermMatrix(corpus)
tdm.tfidf <- tm::weightTfIdf(tdm)
```

```
## Warning in tm::weightTfIdf(tdm): empty document(s): 44
```

```
tdm.tfidf <- tm::removeSparseTerms(tdm.tfidf, 0.999)  # sparsity being not well handled overall in R
tfidf.matrix <- as.matrix(tdm.tfidf)
```

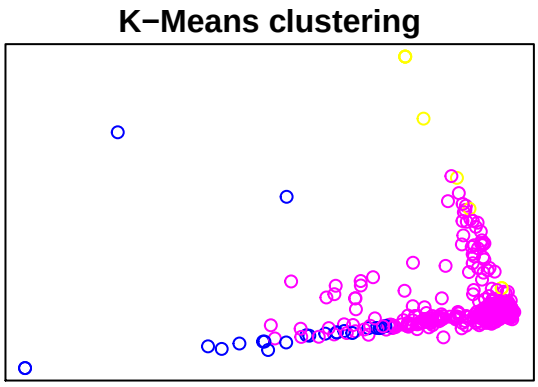
Afterwards, we perform kmeans algorithm to cluster in {2,4,8,16} classes.

```
k = 8
clustering.kmeans <- kmeans(tfidf.matrix, k)
master.cluster <- clustering.kmeans$cluster
```

We perform Classical multidimensional scaling (SMC) to map the data (distance matrix) into 2D dimension and then visualize it.

```
dist.matrix = proxy::dist(tfidf.matrix, method = "cosine")
points <- cmdscale(dist.matrix, k = 2)
palette <- colorspace::diverge_hcl(k) # Creating a color palette
previous.par <- par(mfrow=c(2,2), mar = rep(1.5, 4))

plot(points, main = 'K-Means clustering', col = as.factor(master.cluster),
     mai = c(0, 0, 0, 0), mar = c(0, 0, 0, 0),
     xaxt = 'n', yaxt = 'n', xlab = '', ylab = '')
```


K−Means clustering