San Francisco Bay Guardian Paper Text Mining

Nelson Corrocher

August 21, 2016

The following section prepare the corpus of the text mining. In this exercise, the folder texts/ is being used so the text object must be put in that folder.

```
## [1] "C:/Users/corrocherfilhonw/workspace/R/Week 7/texts"
```

```
## [1] "San_Francisco_Bay_Guardian_48_17.txt"
```

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

The following section pre-process the words (removes punctuation and number, convert to lwoercase, remove stopwords, extract the word stem and remove whitepsaces).

```
docs <- tm_map(docs, removePunctuation)   # *Removing punctuation:*
docs <- tm_map(docs, removeNumbers)      # *Removing numbers:*
docs <- tm_map(docs, tolower)   # *Converting to lowercase:*
docs <- tm_map(docs, removeWords, stopwords("english"))   # *Removing "stopwords"
library(SnowballC)
docs <- tm_map(docs, stemDocument)   # *Removing common word endings* (e.g., "ing", "es")
docs <- tm_map(docs, stripWhitespace)   # *Stripping whitespace
docs <- tm_map(docs, PlainTextDocument)
```

Staging the Data (as Document-Term Matrix)

```
dtm <- DocumentTermMatrix(docs)
tdm <- TermDocumentMatrix(docs)
```

The following section is used for some data familiarization and to catch evident outliers:

```
freq <- colSums(as.matrix(dtm))
length(freq)
```

```
## [1] 7744
```

```
ord <- order(freq)
m <- as.matrix(dtm)
dim(m)
```

```
## [1]    1 7744
```

```
write.csv(m, file="DocumentTermMatrix.csv") # The excel file in the working directory now holds word co
```

Some cleaning up of the data below. The output shows two rows of numbers. The top number is the frequency with which words appear and the bottom number reflects how many words appear that frequently:

```
#  Start by removing sparse terms:
dtms <- removeSparseTerms(dtm, 0.1) # This makes a matrix that is 10% empty space, maximum.
head(table(freq), 20) ### Word Frequency
```

```
## freq
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 5107 1218  469  257  156  113   82   65   44   28   33   31   15    9    5
##   16   17   18   19   20
##   14   10   10    4    5
```

```
tail(table(freq), 20)
```

```
## freq
##  35  36  37  38  40  41  42  43  47  51  52  54  56  61  62  64  70  73
##   2   2   2   1   1   2   1   1   1   4   1   1   1   1   3   1   1   1
## 112 134
##   2   1
```

Next, let's consider only biggest frequencies (in this example, 42):

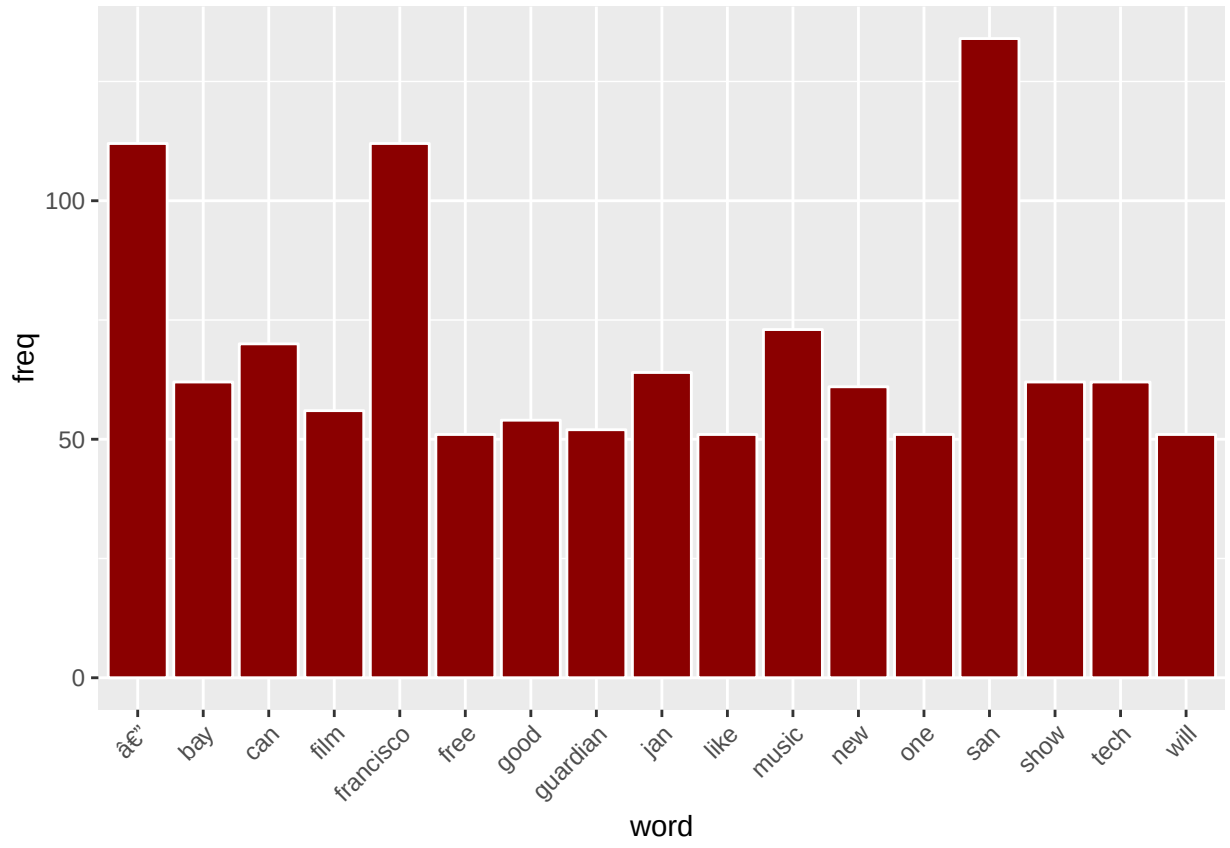
```
freq <- colSums(as.matrix(dtms))  # Matrix is too big to be shown on this document
f_list <- findFreqTerms(dtm, lowfreq=30)   # <<<< Change it to the most appropriate for the data
f_list
```

```
##  [1] "â<U+0080><U+0094>"       "also"      "arts"      "bay"       "call"
##  [6] "can"       "city"      "classifi"  "cultur"    "drink"
## [11] "feb"       "film"      "first"     "food"      "francisco"
## [16] "fre"       "free"      "get"       "good"      "guardian"
## [21] "housing"   "ing"       "jan"       "just"      "last"
## [26] "like"      "music"     "new"       "news"      "now"
## [31] "one"       "open"      "people"    "san"       "sat"
## [36] "selector"  "sfbg"      "show"      "sun"       "tech"
## [41] "time"      "will"
```

Ploting Word Frequencies (Only words that appear at least 50 times):

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```



Next, we can try to find the correlation between terms that occurs in the corpus. In this context, correlation is a quantitative measure of the co-occurrence of words in documents. In this example we use the words "san" and "francisco"

```
findAssocs(dtm, c("san","francisco") ,corlimit = 0.6) # <<<< Adjust the corlimit to the desired correla
```

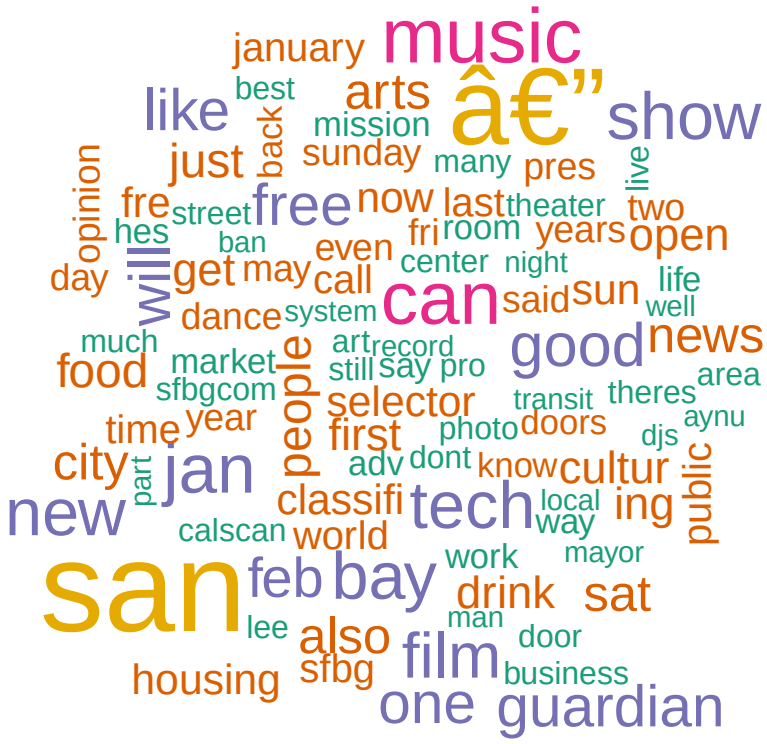
```
## $san
## numeric(0)
##
## $francisco
## numeric(0)
```

```
#findAssocs(dtm, f_list ,corlimit = 0.6) <<<<< Alternatively, we can use the list of words that had fre
```

Here we create a word cloud based on the frequency of the words.

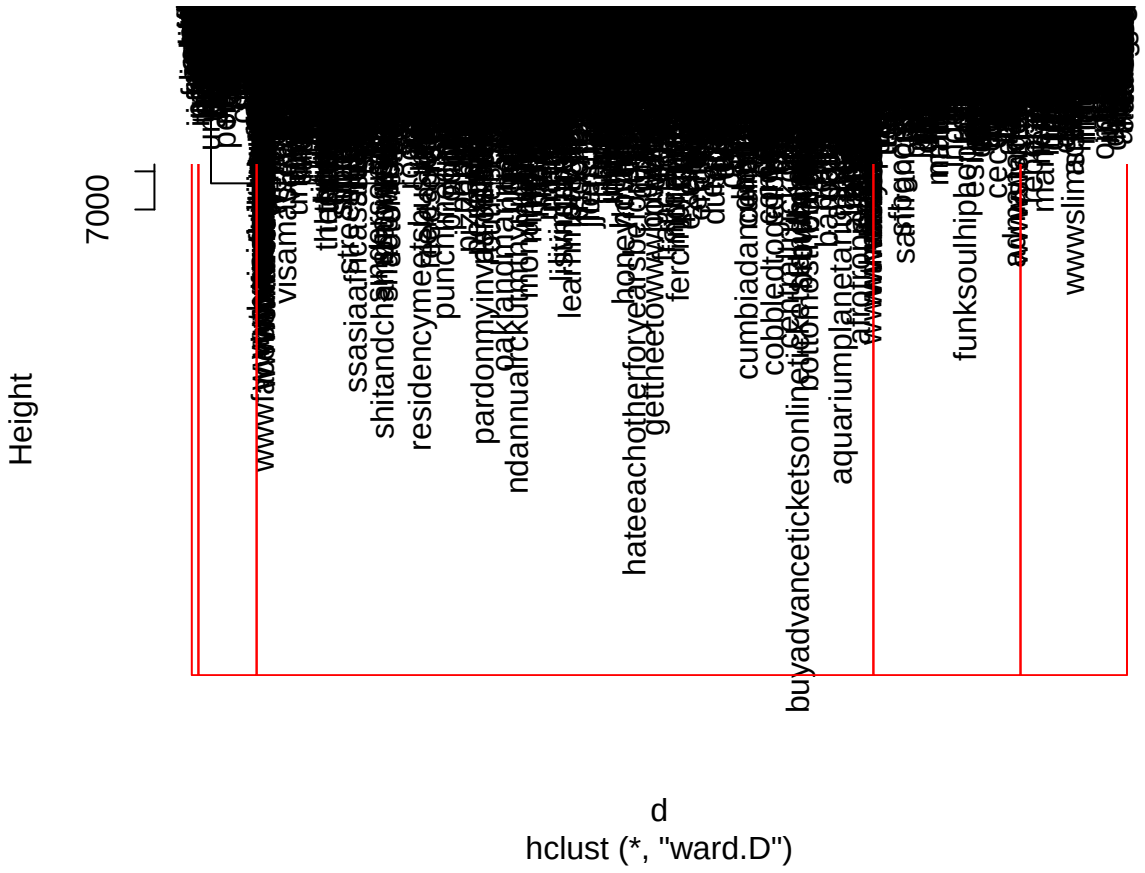
```
## Loading required package: RColorBrewer
```

```
## Warning in wordcloud(names(freq), freq, max.words = 100, rot.per = 0.1, :
## francisco could not be fit on page. It will not be plotted.
```



The code below draws a tree diagram of the cluster.

```
dtms <- removeSparseTerms(dtm, 0.15) # This makes a matrix that is only 15% empty space.
library(cluster)
d <- dist(t(dtms), method="euclidian")   # First calculate distance between words
fit <- hclust(d=d, method="ward.D")
plot.new()
plot(fit, hang=-1)
groups <- cutree(fit, k=5)   # "k=" defines the number of clusters you are using
rect.hclust(fit, k=5, border="red") # draw dendogram with red borders around the 5 clusters
```



The code below should clusterize the corpus using k-means algorithms. For some reason, the code either takes too long or get into an infinite loop. For the sake of this exercise, it was commented out.

```
#library(fpc)
#library(cluster)
#dtms <- removeSparseTerms(dtm, 0.15) # Prepare the data (max 15% empty space)
#d <- dist(t(dtms), method="euclidian")
#kfit <- kmeans(d, 2)
#clusplot(as.matrix(d), kfit$cluster, color=T, shade=T, labels=2, lines=0)
```