```
library(NLP)
library(corpus)
library(tm)
library(wordcloud)
```

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

```
library(RColorBrewer)
library(ggplot2)
```

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

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

```
library(ggthemes)
library(RWeka)
```

```
library(qdapDictionaries)
library(qdapRegex)
```

```
##
## Attaching package: 'qdapRegex'
```

```
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
library(qdap)
```

```
## Loading required package: qdapTools
```

```
##
## Attaching package: 'qdap'
```

```
## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, as.TermDocumentMatrix
```

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

```
## The following object is masked from 'package:base':
##
##     Filter
```

```
options(mc.cores=1)
```

Loading data

```
data_breaches <- read.csv("Data_Breaches_r.csv")
data_breaches <- as.data.frame(data_breaches)
```

```
names(data_breaches)
```

```
##  [1] "ï..1St.Source"    "X2Nd.Source"      "X3Rd.Source"
##  [4] "Alternative.Name" "Entity"           "Method.of.Leak"
##  [7] "Records.Lost"     "Sector"           "Source.name"
## [10] "Story"            "Year"
```

```
# creating the corpus for the n reviews. corpus_review is a collection of the n revie
ws
corpus_breaches <- VCorpus(VectorSource(data_breaches$Story))
```

Text Pre-processing

In this part, the corpus created is pre-processed

```
# set stopwords you would like to remove
own_stopwords <- c()
```

```
# converting to lowercase
data_breaches <- tm_map(corpus_breaches, content_transformer(tolower))

# removing punctuation
data_breaches <- tm_map(data_breaches, removePunctuation)

#removing numbers from text
data_breaches <- tm_map(data_breaches, removeNumbers)

# removing stopwords
data_breaches <- tm_map(data_breaches, removeWords, stopwords("english"))

# remove our own stopwords
data_breaches <- tm_map(data_breaches, removeWords, own_stopwords)

# stemming the document
data_breaches <- tm_map(data_breaches, stemDocument)
```

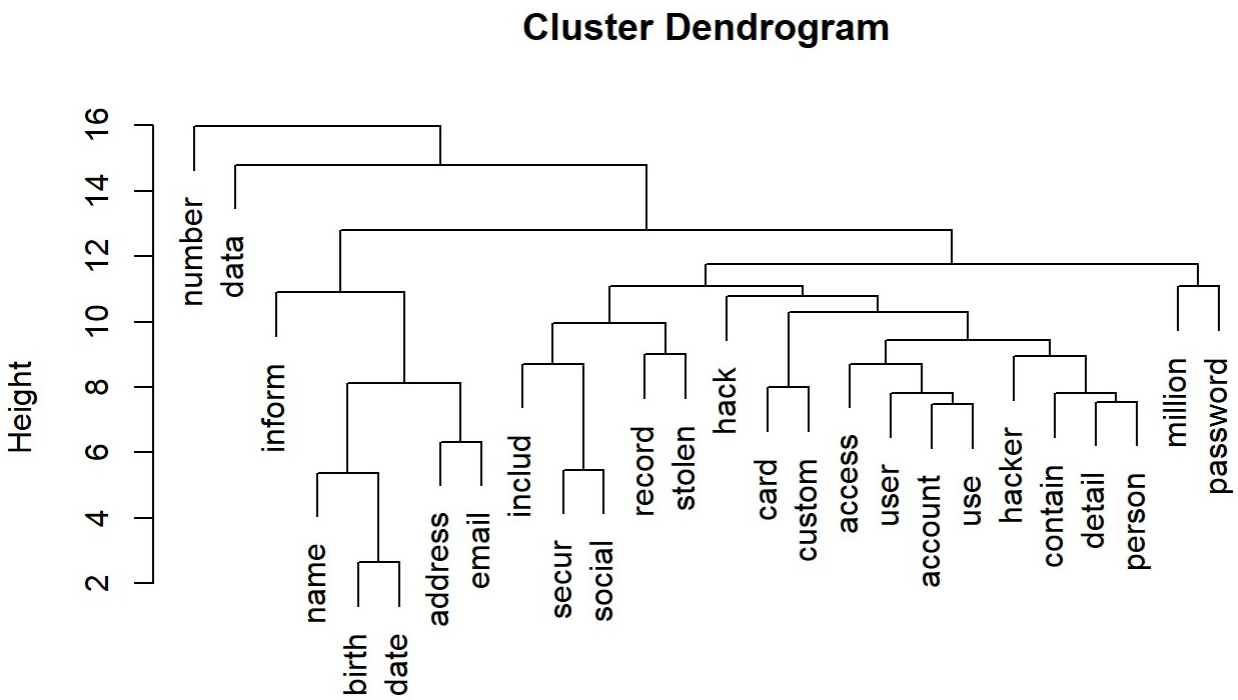
Document-Term-Matrix

The document-term-matrix counts the number of times a word appear in a document

```
#create the dtm and the tdm
breaches_dtm <- DocumentTermMatrix(data_breaches)
breaches_tdm <- TermDocumentMatrix(data_breaches)
```

Cluster Dendrogram

```
breaches_tdm2 <- removeSparseTerms(breaches_tdm, sparse = 0.9)
hc <- hclust(d = dist(breaches_tdm2, method = "euclidean"), method = 'complete')
plot(hc)
```

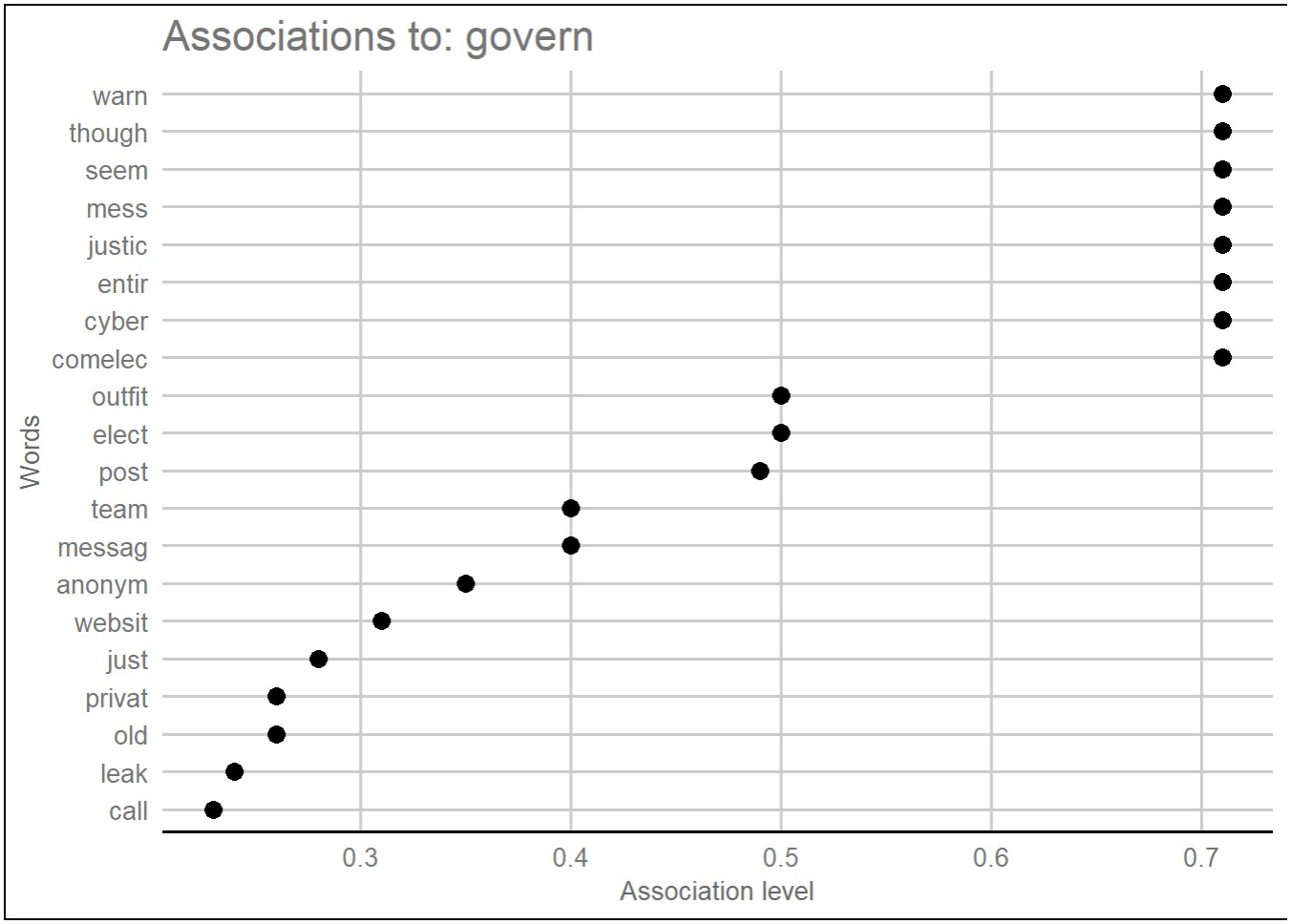


Word Associations

```
# WORD ASSOCIATIONS
word_assoc <- "govern"
associations <- findAssocs(breaches_tdm, as.String(word_assoc), 0.2)
```

```
# creating associations dataframe
associations_df <- list_vect2df(associations)[, 2:3]
```

```
ggplot(associations_df, aes(y = associations_df[, 1])) +
  geom_point(aes(x = associations_df[, 2]), data = associations_df, size = 3) +
  ggtitle("Associations to: "+as.String(word_assoc)) +
  theme_gdocs() + theme(text = element_text(size=10)) + labs(x="Association level", y
= "Words")
```



Frequency of the words

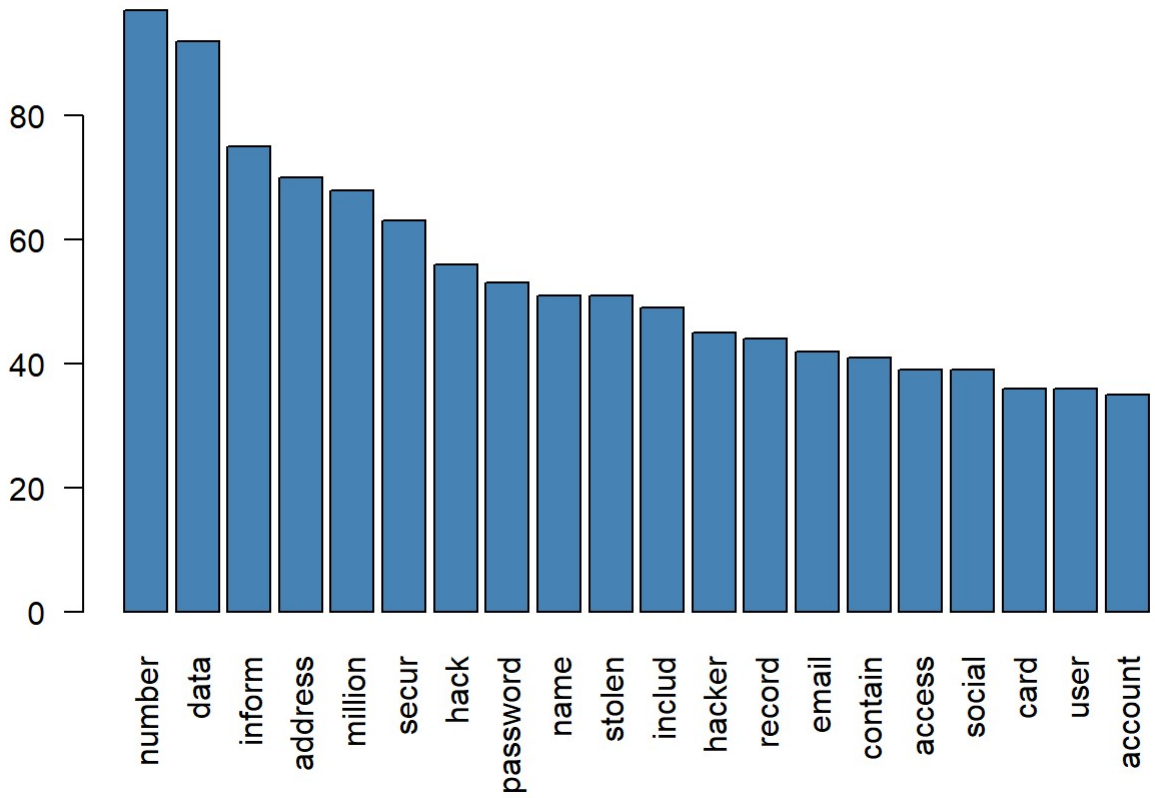
```
#convert TDM to matrix. review number on the columns and words on the wows. Values ar
e frequencies
breaches_matrix <- as.matrix(breaches_tdm)
```

```
# sums the frequency of each word in all documents
breaches_term_freq <- rowSums(breaches_matrix)
```

```
# sort by frequency
breaches_term_freq <- sort(breaches_term_freq, decreasing = T)
# view the top 10 most common words
breaches_term_freq[1:20]
```

```
##   number     data   inform  address  million    secur     hack password
##       97       92       75       70       68       63       56       53
##     name   stolen   includ   hacker   record    email  contain   access
##       51       51       49       45       44       42       41       39
##   social     card     user  account
##       39       36       36       35
```

```
barplot(breaches_term_freq[1:20], col = "steel blue", las = 2)
```




```
breaches_word_freq <- data.frame(term = names(breaches_term_freq), num = breaches_ter
m_freq)
# create wordcloud
wordcloud(breaches_word_freq$term, breaches_word_freq$num, max.words = 50, colors = c
("blue", "black", "tomato"))
```

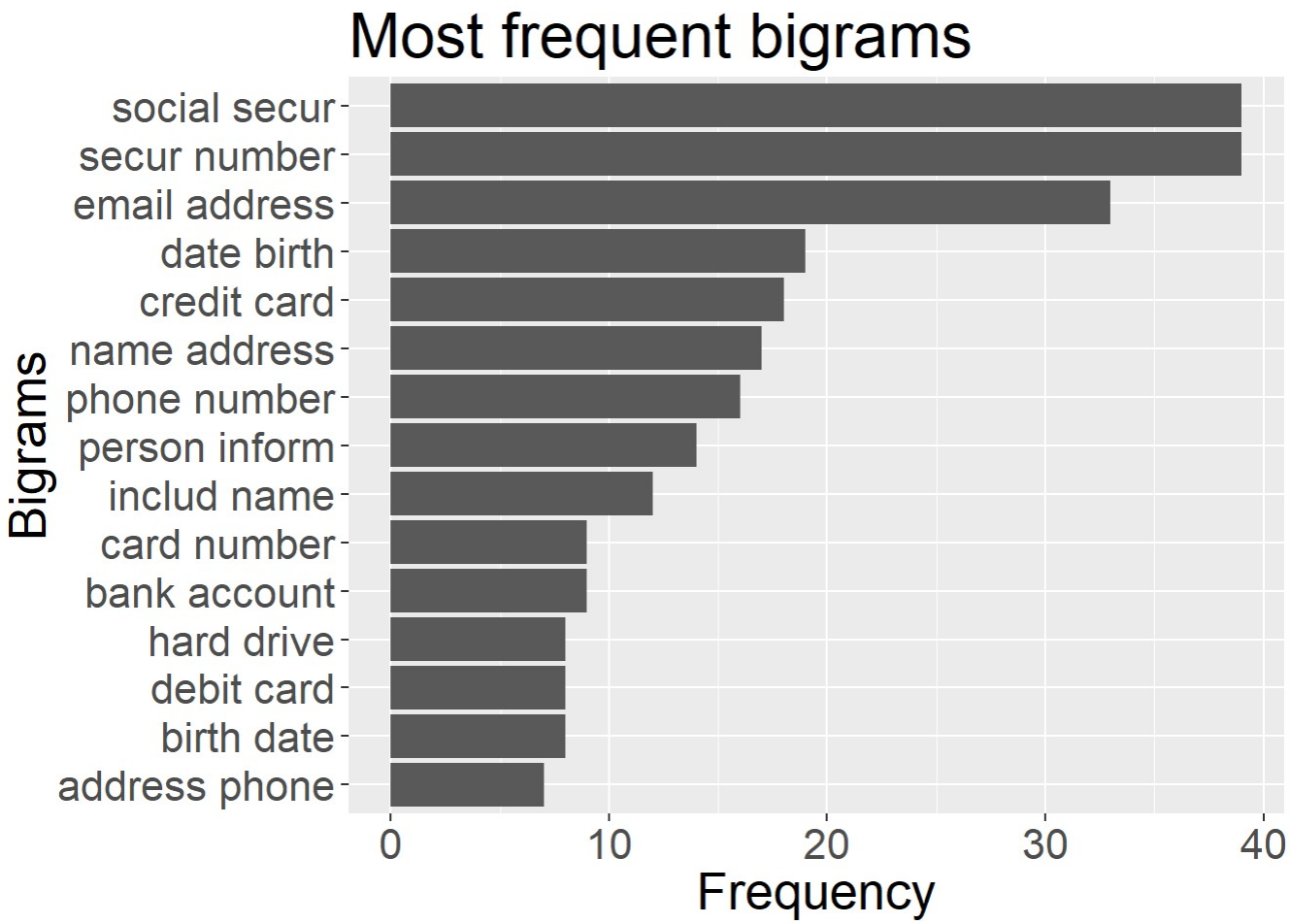

```
BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
tdm_breaches_bigram = TermDocumentMatrix(data_breaches, control = list(tokenize = Big
ramTokenizer))
```

Bi-gram frequency study

```
freq = sort(rowSums(as.matrix(tdm_breaches_bigram)),decreasing = TRUE)
freq.df = data.frame(word=names(freq), freq=freq)
head(freq.df, 20)
```

```
##                              word freq
## secur number          secur number   39
## social secur          social secur   39
## email address        email address   33
## date birth              date birth   19
## credit card            credit card   18
## name address          name address   17
## phone number          phone number   16
## person inform        person inform   14
## includ name            includ name   12
## bank account          bank account    9
## card number            card number    9
## birth date              birth date    8
## debit card              debit card    8
## hard drive              hard drive    8
## address phone        address phone    7
## credit debit          credit debit    7
## gain access            gain access    7
## account number      account number    6
## address password address password    6
## address social      address social    6
```

```
ggplot(head(freq.df,15), aes(reorder(word,freq), freq)) +
  geom_bar(stat = "identity") + coord_flip() +
  xlab("Bigrams") + ylab("Frequency") +
  ggtitle("Most frequent bigrams") + theme(text = element_text(size=20))
```


Most frequent bigrams
social secur
secur number
email address
date birth
credit card
name address
phone number
person inform
includ name
card number
bank account
hard drive
debit card
birth date
address phone
Bigrams
0
10
20
30
40
Frequency