**1. Function**

DocumentTermMatrix()

TermDocumentMatrix()

Inspect()

Wordcloud()

termFreq()

as.matrix()

hclust()

tm\_map(acq, content\_transformer(tolower))

tm\_map(revs, removeWords, stopwords("english"))

tm\_map(revs, removePunctuation)

tm\_map(revs, removeNumbers)

tm\_map(revs, stripWhitespace)

wordcloud(words = d$word, freq = d$freq, min.freq = 1,max.words=200, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))

fit<-hclust(distMatrix,method="ward.D2")

a = tokenize\_words(e)

FreqMat <- spc(Vm=FreqMat$Freq, m=FreqMat$ST)

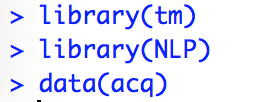
plot(FreqMat, log="x")

A:

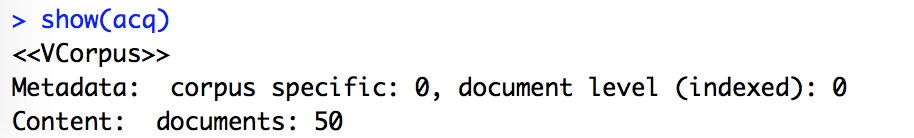
To work on the dataset acq , we need first to installing the package tm . the following screenshot shows the command to install the package and the output confirms installing the package.



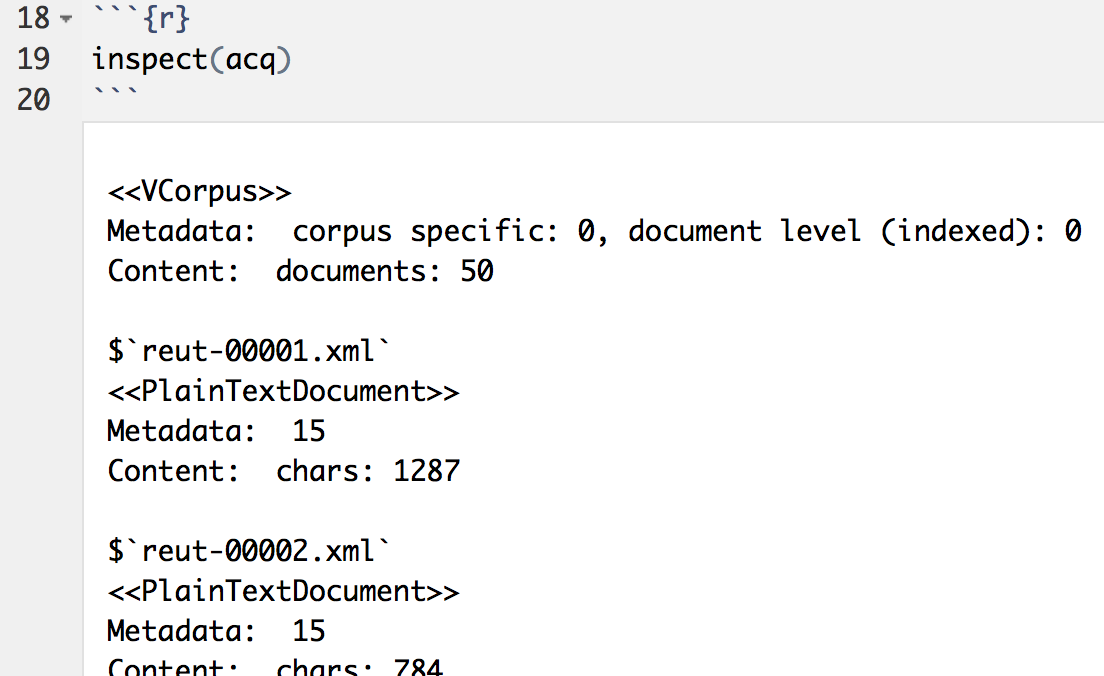
After installing the package, we call the libraries TM and NLP, then we call the data acq. The following three lines add the dataset to the project



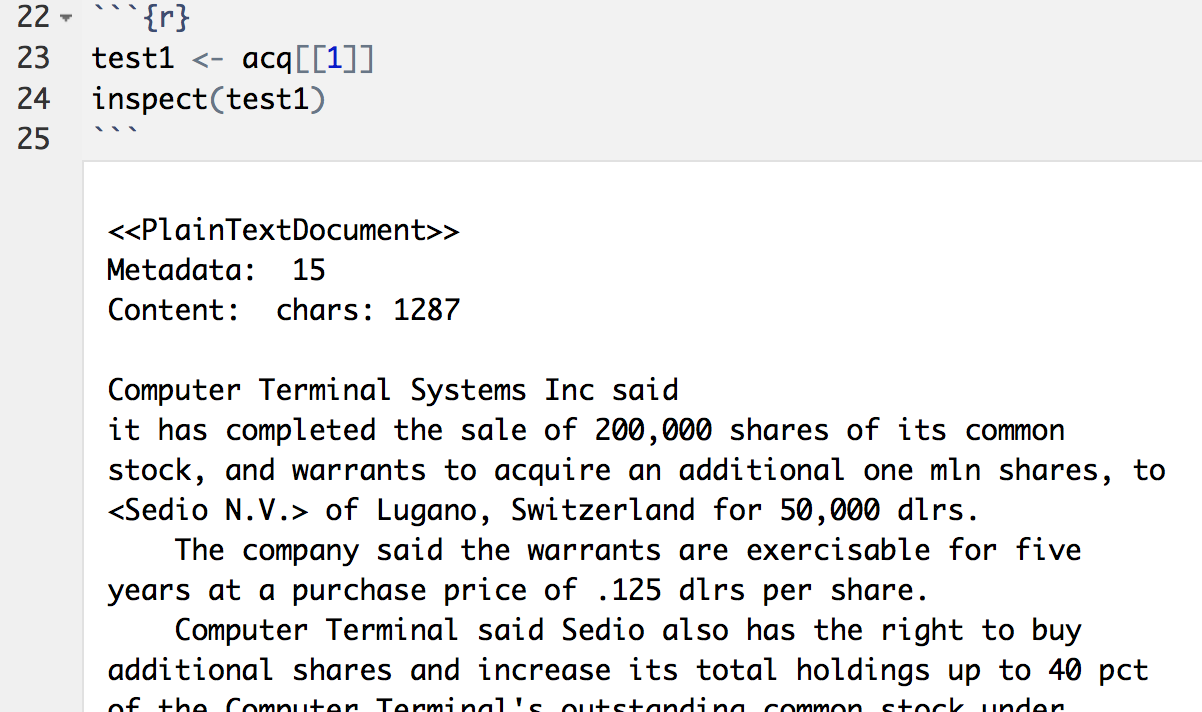
In order to make sure that the dataset is imported we can use the following command. It shows summary of the object. It says that acq contains 50 documents.



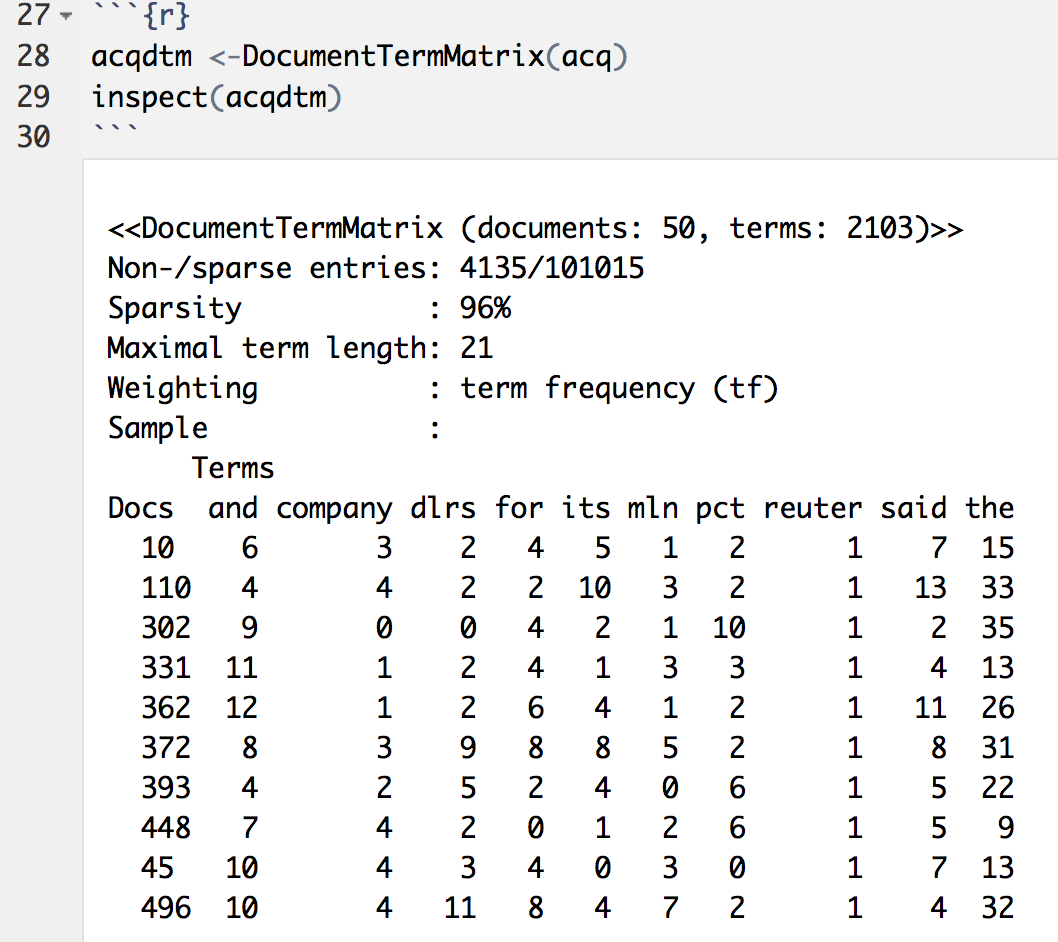
We may use the function inspect(x), so we get more details about the data. It will show the number documents and the number of characters in every file as follows. The screenshot displays part of the output. The rest of the output is omitted.



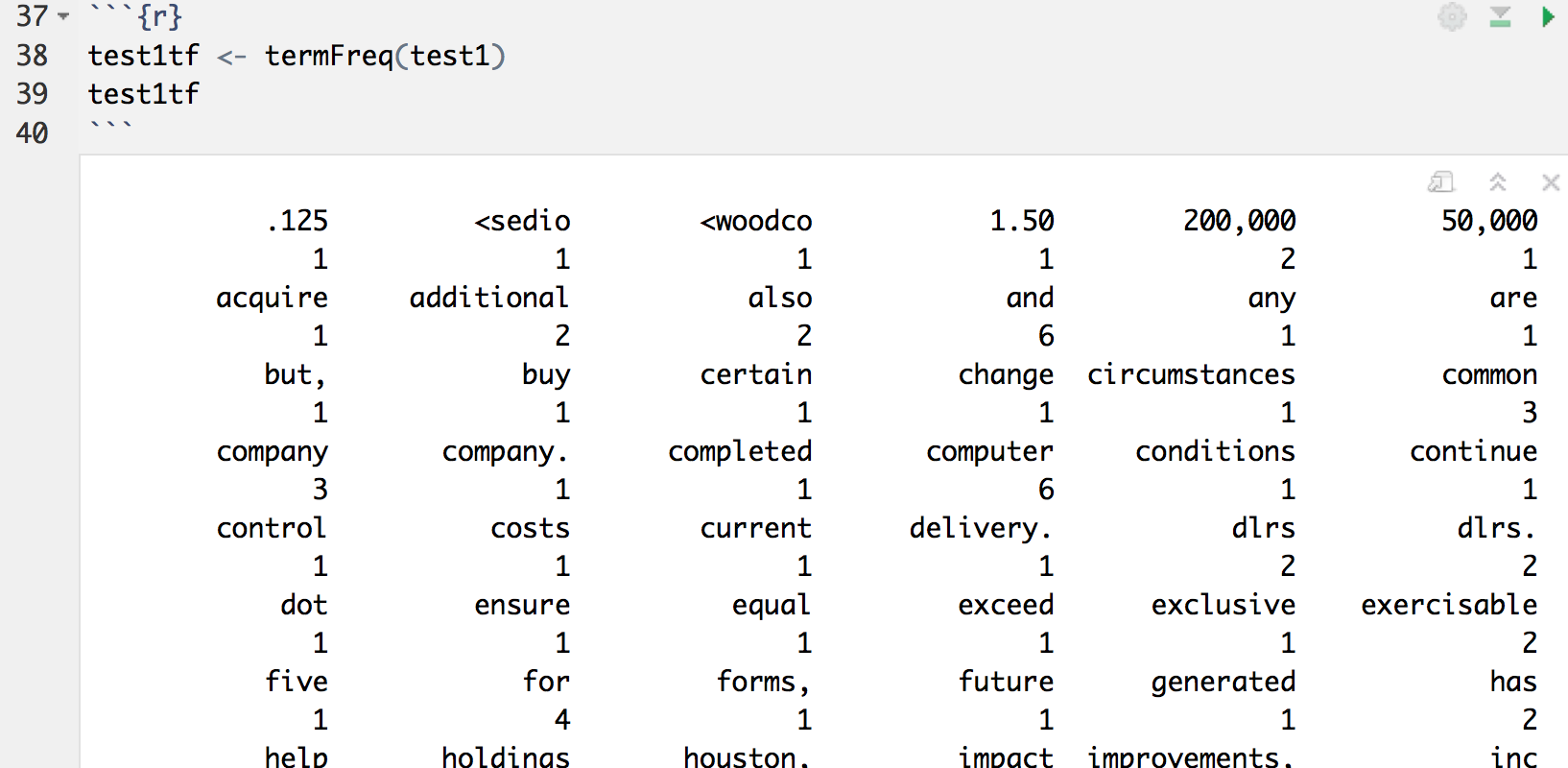
We may also use the function inspect(x) to see the content of every file. The following commands shows the summary of the first document in acq and shows the content of the file. The screenshot displays part of the output. The rest of the output is omitted.



We may use the function DocumentTermMatrix(x) to count how many times a word is existing in each document. Then, When we use the function inspect(x), we get a summary of the document-term matrix and a sample of it. Inspecting the matrix says that sparsity is 96% and the maximal term length is 21.



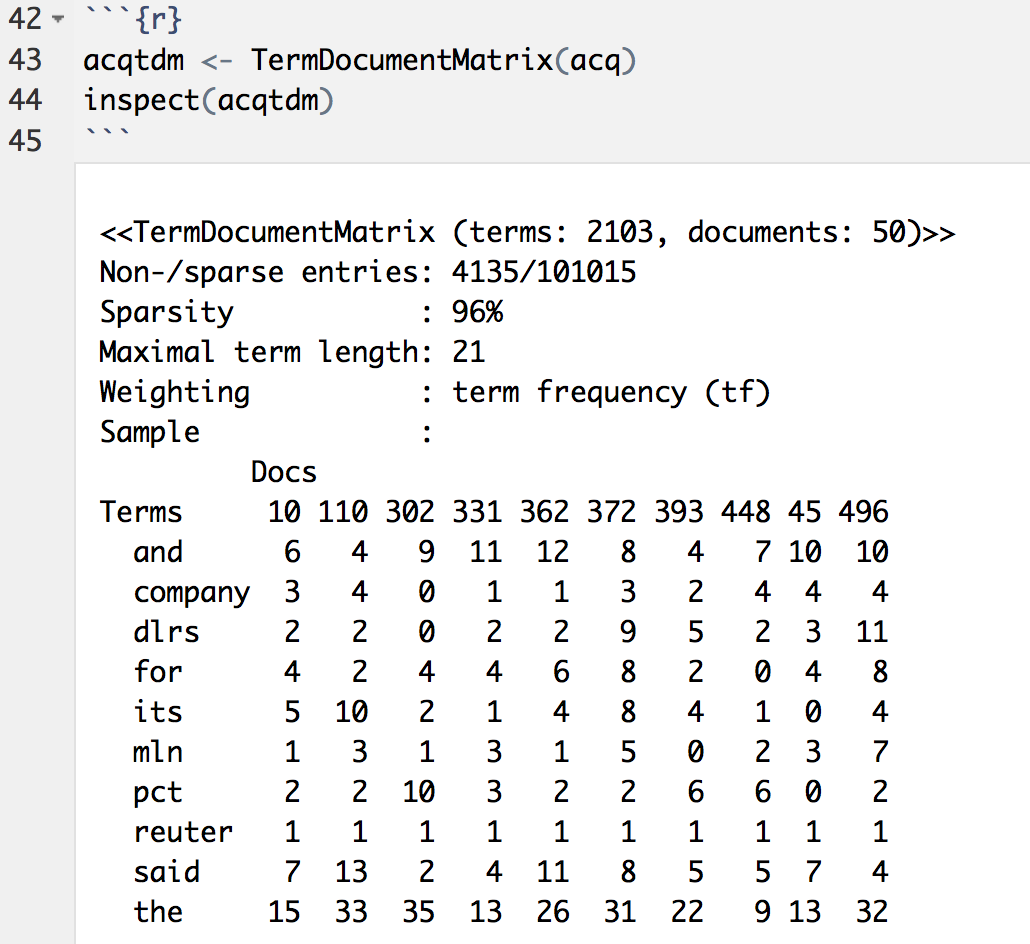
We may also count the number of times a word exists in a file by using the function termFreq(x). We can see that the word computer has high frequency of 6 appearances in the first file of acq.



Then, we convert the term frequency into dataframe by using the function as.data.frame(x). In the following example, we converted the term frequency of the first document. The result is 111 rows as follows.

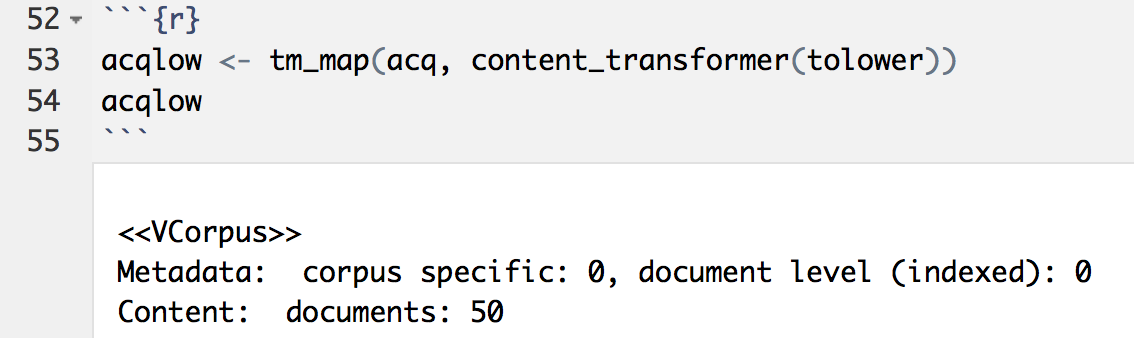


We may use the function TermDocumentMatrix(x) to produces a term-document matrix. After inspection, we see that the first document has sparsity of 96% and the maximal term length is 21. This is obvious because we already tested the document-term matrix earlier.

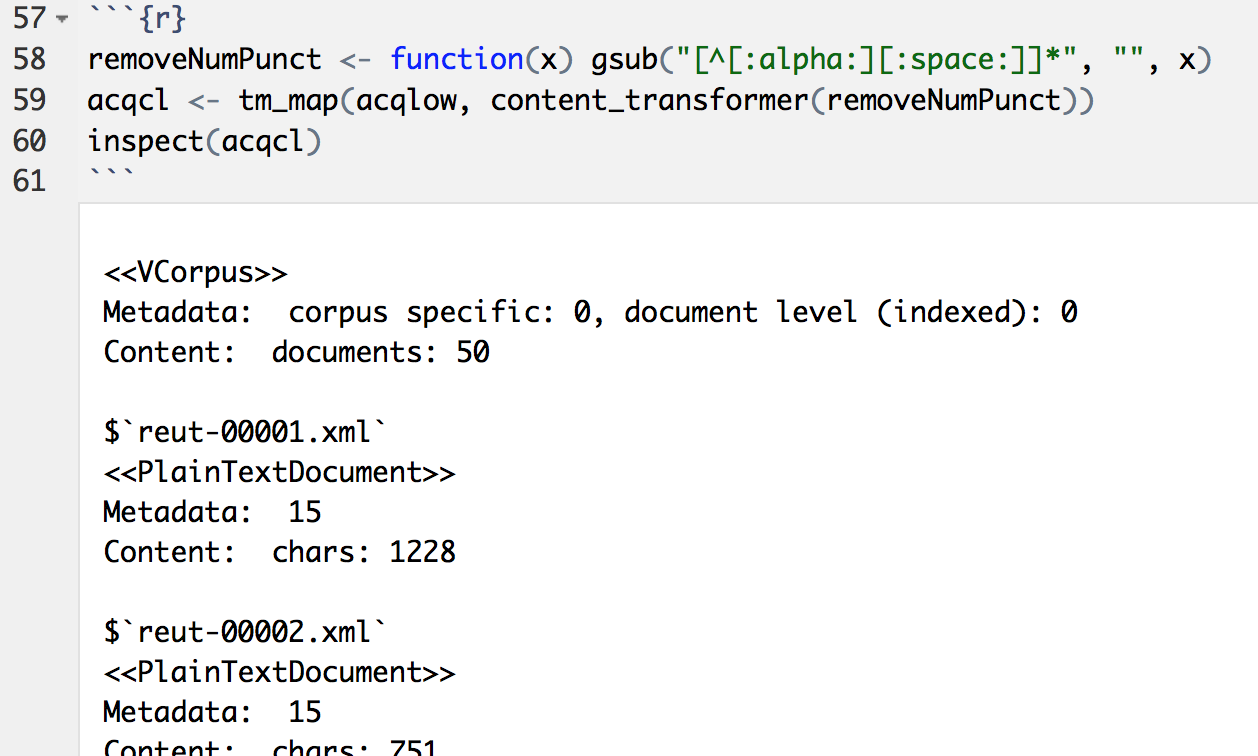


Until this moment, we can see that the data we have include numbers and special characters that are not valuable for our analysis.

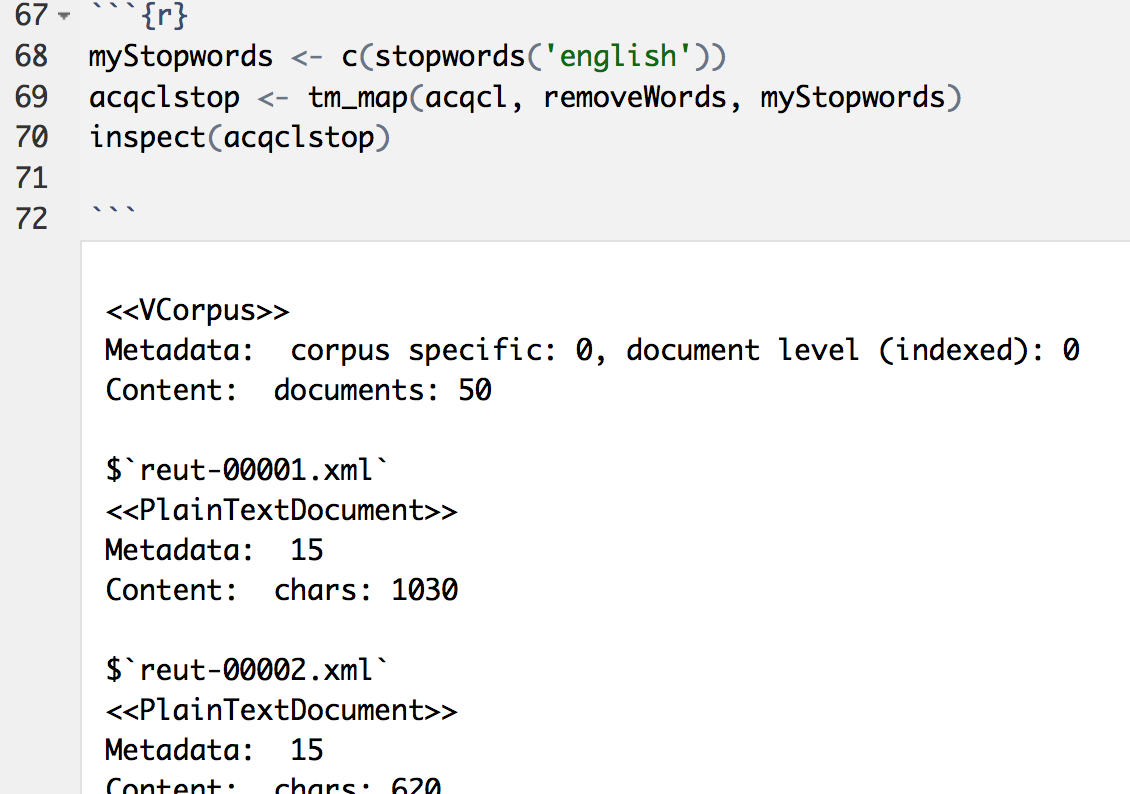
First, we want to convert the text of all the documents in acq into lowercase, so the word count won't matter if it is in the beginning of a sentence or not. We may use the following commands on acq, including the function content\_transformer(x) with the parameter ‘tolower’.



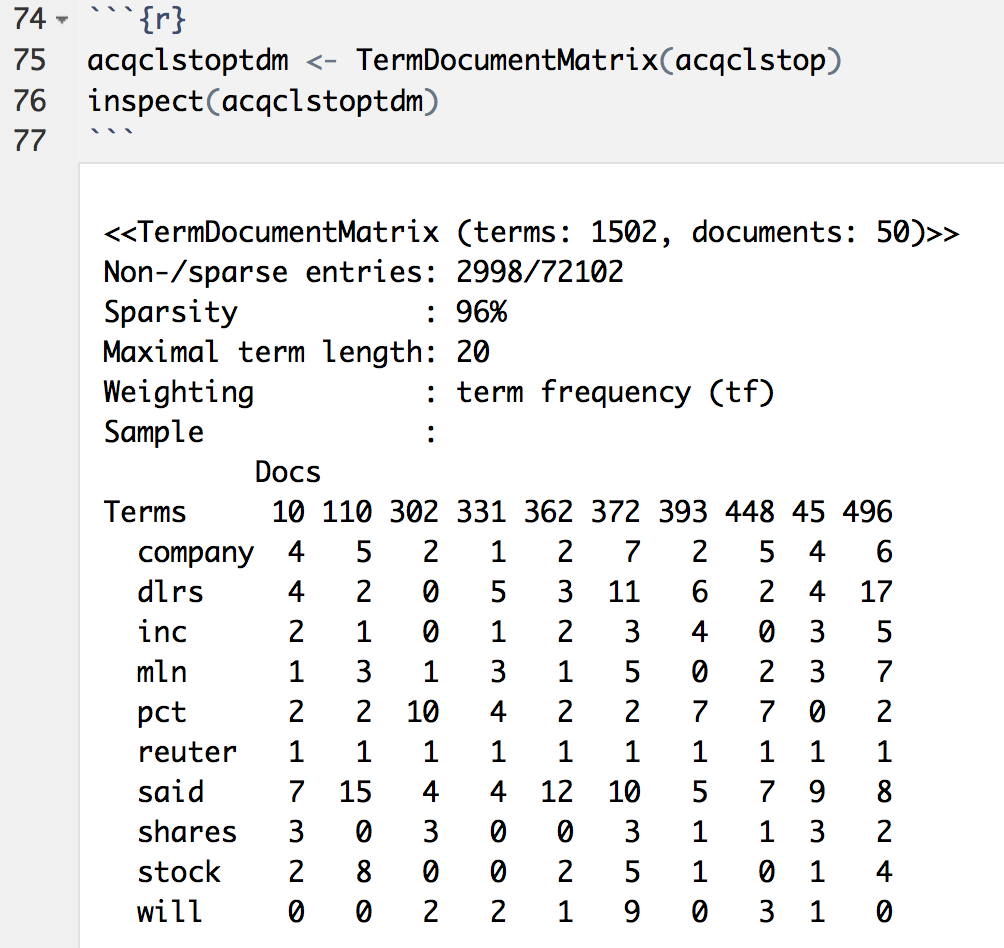
Then, we use the following commands including the function tm\_map(x…) and the function content\_transformer(x) with the parameters to remove numbers, punctuations and anything other than letters and spaces.



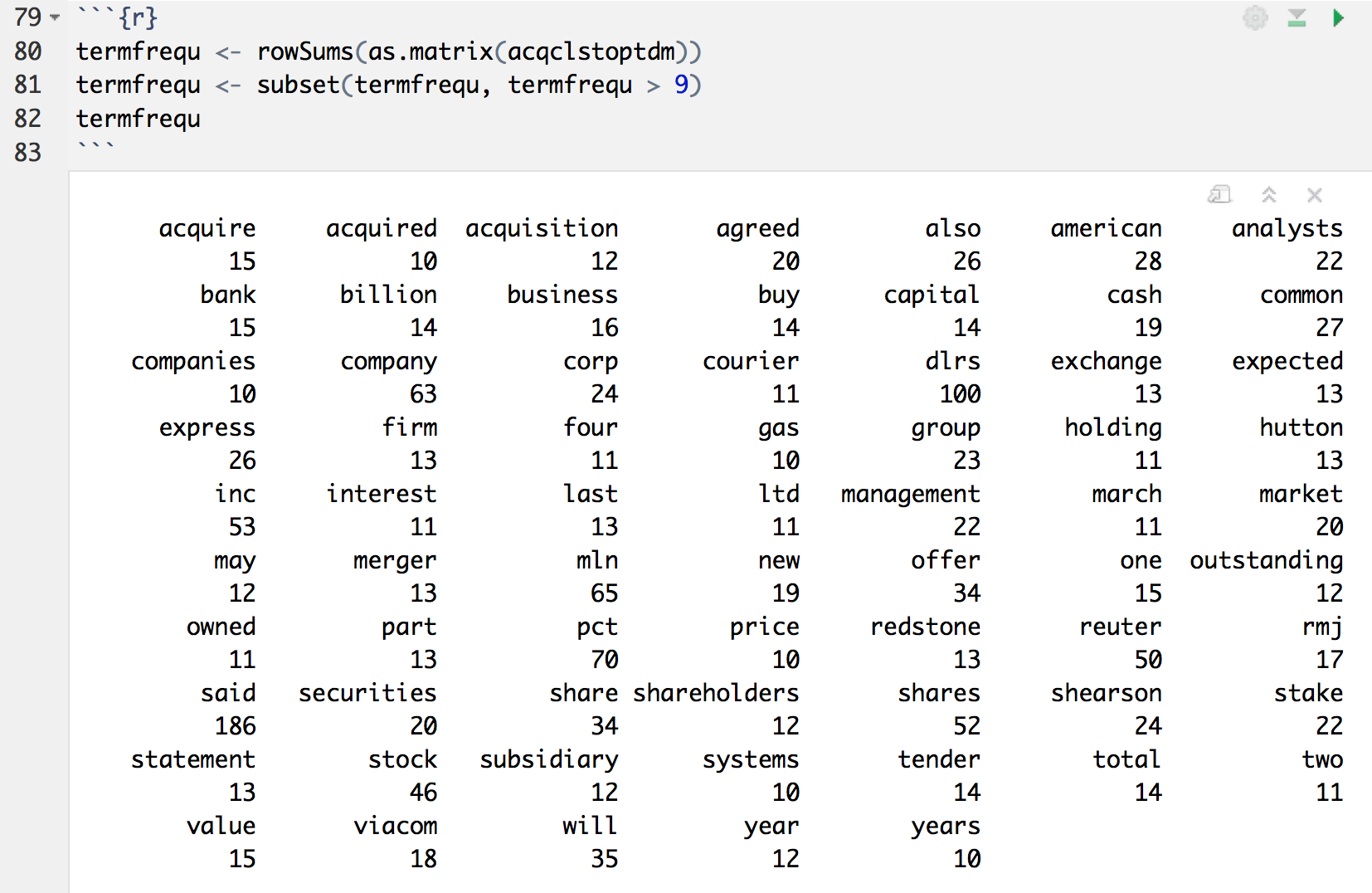
Then, we want to remove the words that we don’t want them to be counted in our analysis because they are not important such as I, they, so, are, is and all stop-words that NLP researchers came up with. The function stopwords(x) returns those words. Then we used the function tm\_map(x…) with their parameters to remove the words.



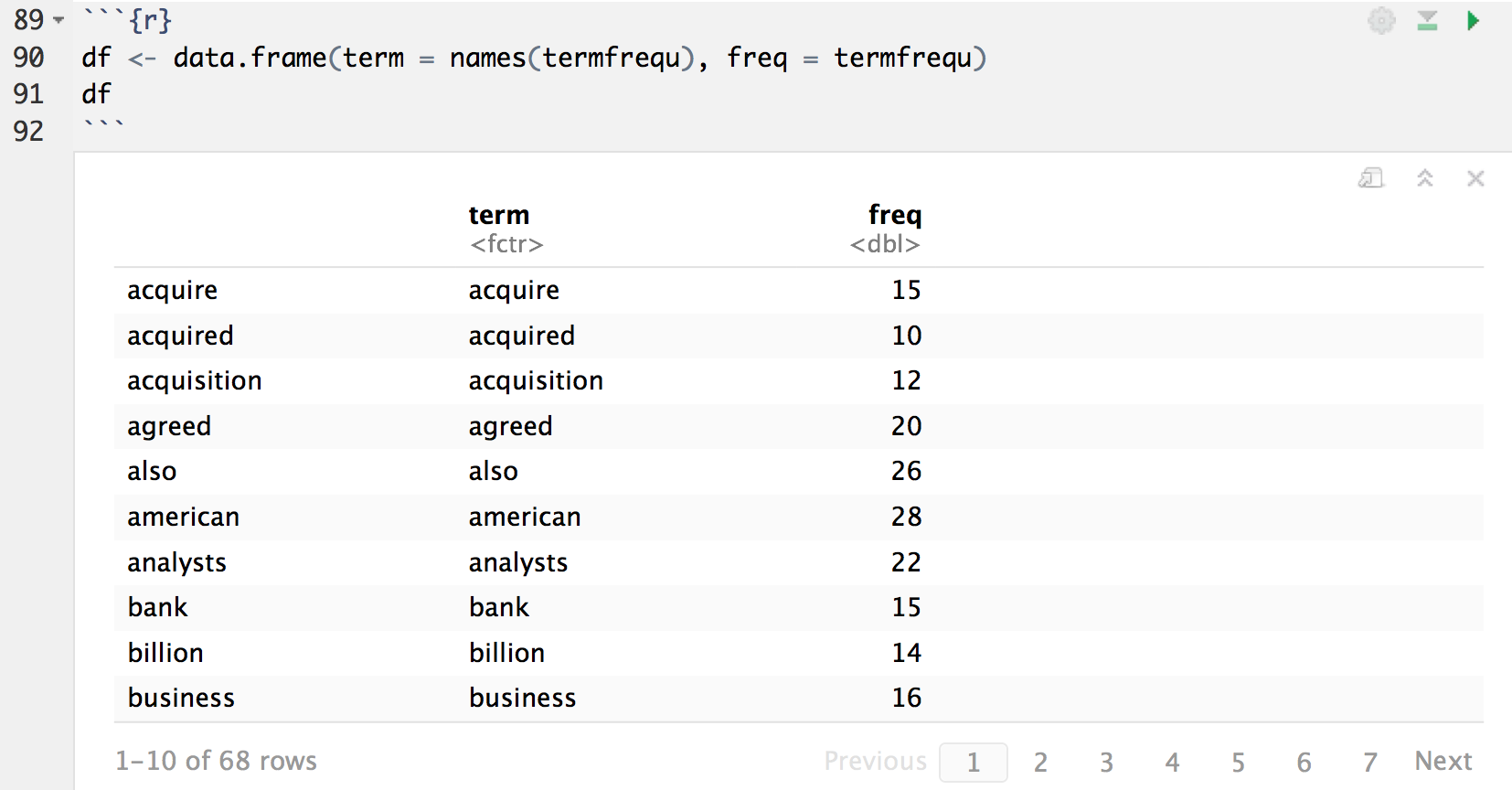
Now we inspect the dataset after removing the noise from it. We find that our data has 1502 terms with 20 maximal term length. While the original dataset had 2103 terms and 21 maximal term length.



After deleting unwanted words, we want to calculate the term frequency in the entire dataset. We do that by computing the summation of the rows of the term-document matrix. Then we focus on the words with high frequency. We test with the condition for the words that has more than 9 frequency. The output shows that ‘reuter’ appears 50 time, ‘dlrs’ appears 100 times, and ‘shares’ appears 52 times in the dataset.



After removing the noise from the data, we also want to convert the term frequency to data frame by utilizing the function data.frame(x…) with its parameters as follows.

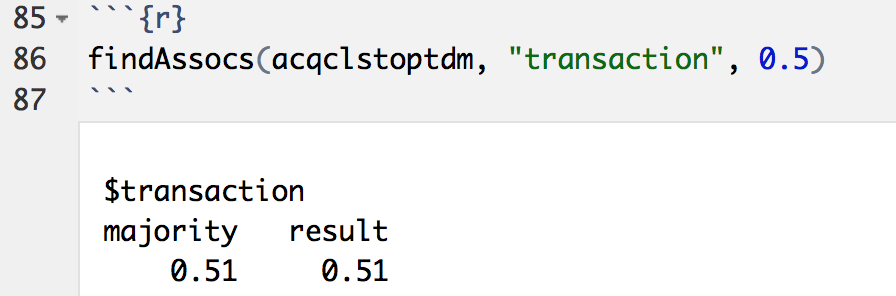


Another way to collect the collect only the terms with hih frequency by using the function FindFreqTerms. We passed the number 10 as parameter to collect the words with 10 appearances or more in our analysed data

.

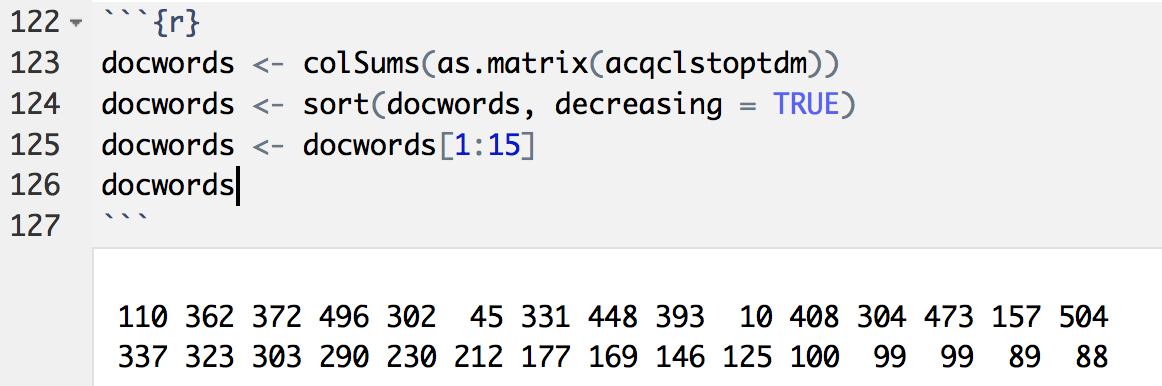


To do more testing with specific words, we may test the associations of words with the word ‘transaction’ with medium correlation limit. We do this by using the function findAssocs(x…)



B.

To find the longest 15 documents we do the following three lines of commands. In the first line, the function colSum(x) calculate the summation of words in every document because we already have the term-document matrix. The second line sorts the documents based on the number of words in decreasing order. The third line extracts only the first 15 documents. Because the documents are sorted decreasingly, we get the the longest 15 documents. The output shows the the address of the 15 documents in the first row, and the number of words every document has in the second row.



**C.**

The code of wordcloud:

dtm2 <- TermDocumentMatrix(revs[22])

m <- as.matrix(dtm2)

v <- sort(rowSums(m),decreasing=TRUE)

d <- data.frame(word = names(v),freq=v)

set.seed(1234)

wordcloud(words = d$word, freq = d$freq, min.freq = 1,max.words=200, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))

The code of dendrogram:

tdm1<- TermDocumentMatrix(revs[7])

distMatrix<-dist(scale(tdm1))

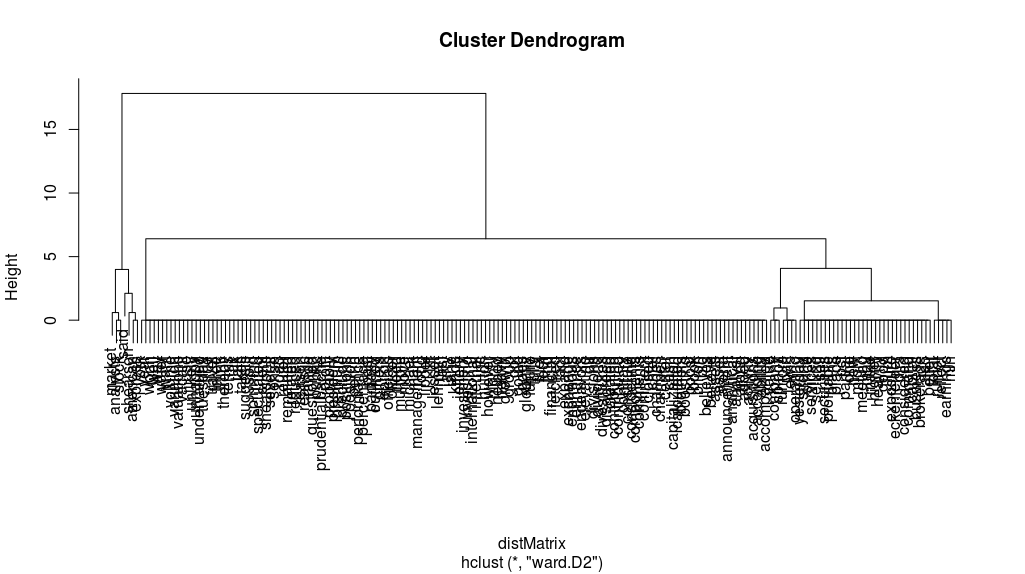
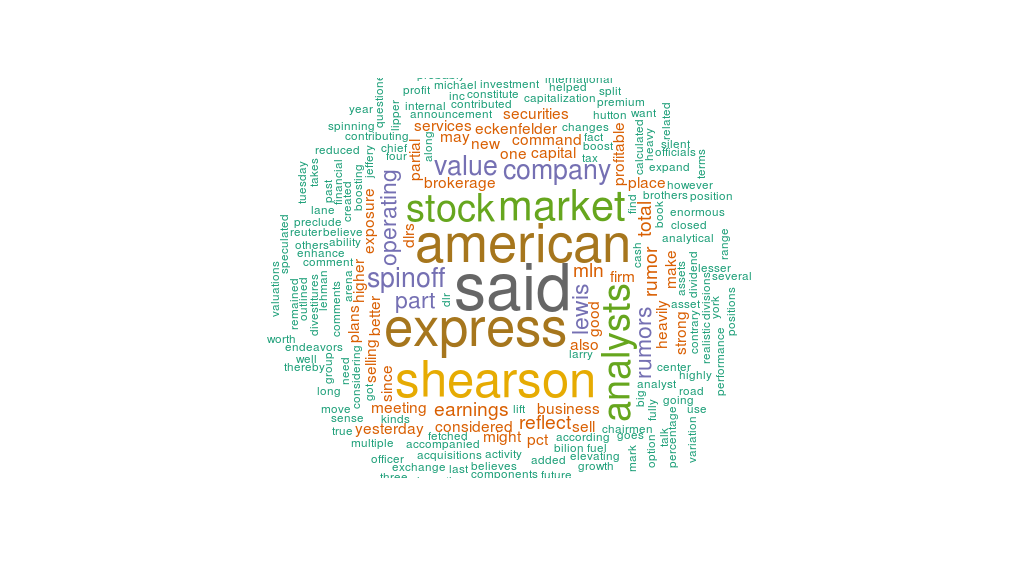
fit<-hclust(distMatrix,method="ward.D2")

plot(fit)

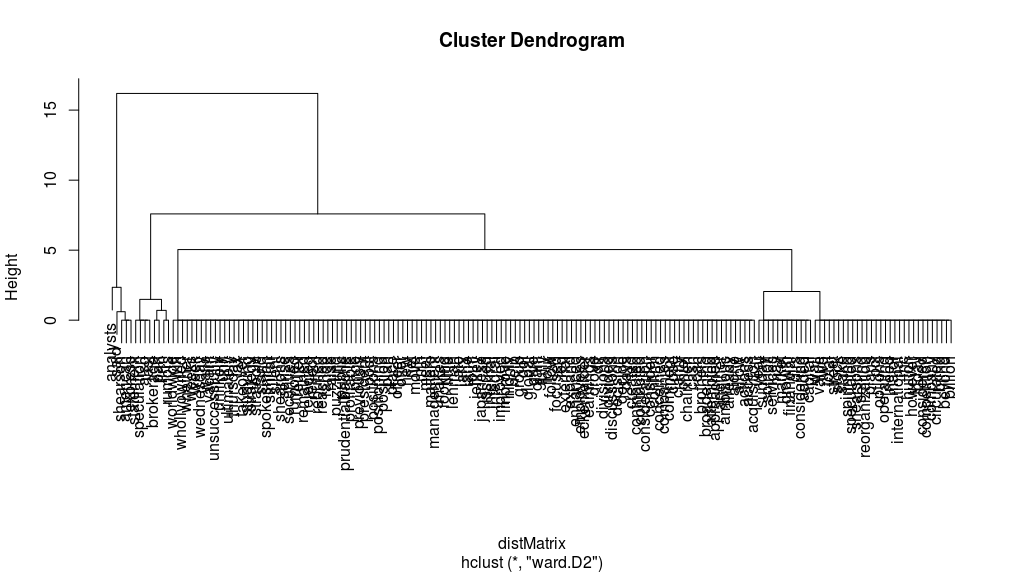
Because of too many word in the document, the prior dendrograms maybe too dense. You can check latter graphs if you want to look at each word clearly.

The graph of 15 longest documents:

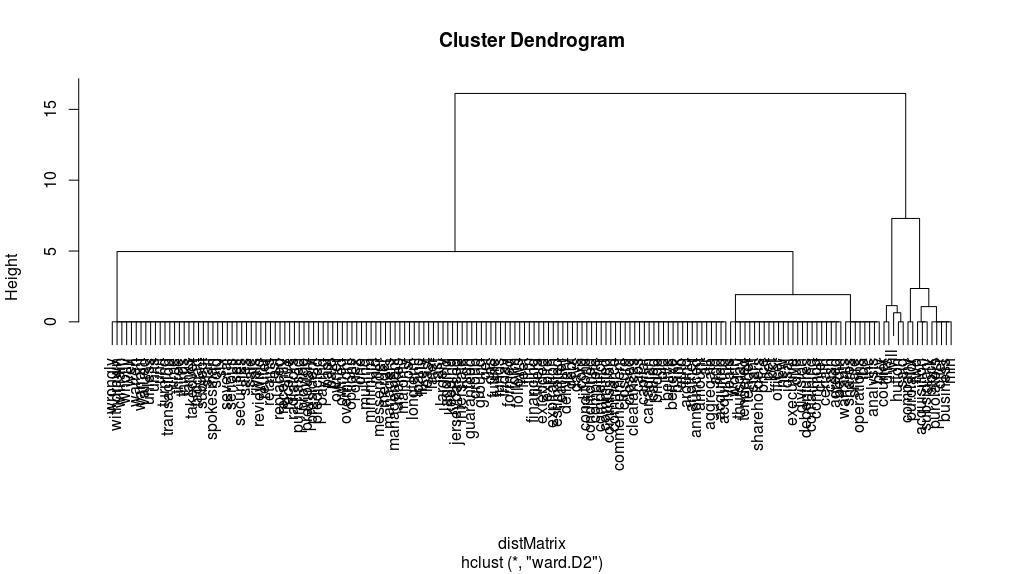
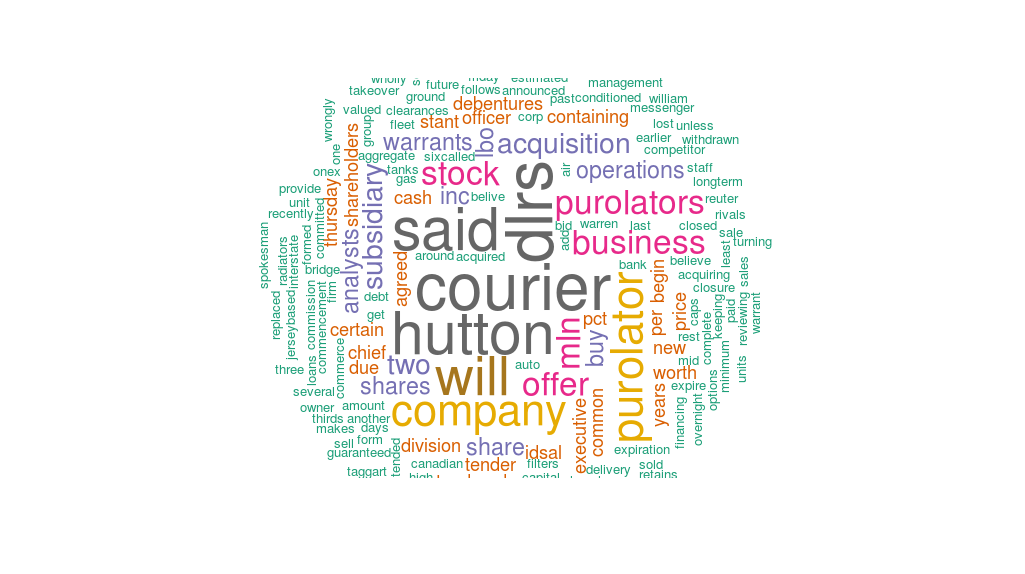
**110:**

****

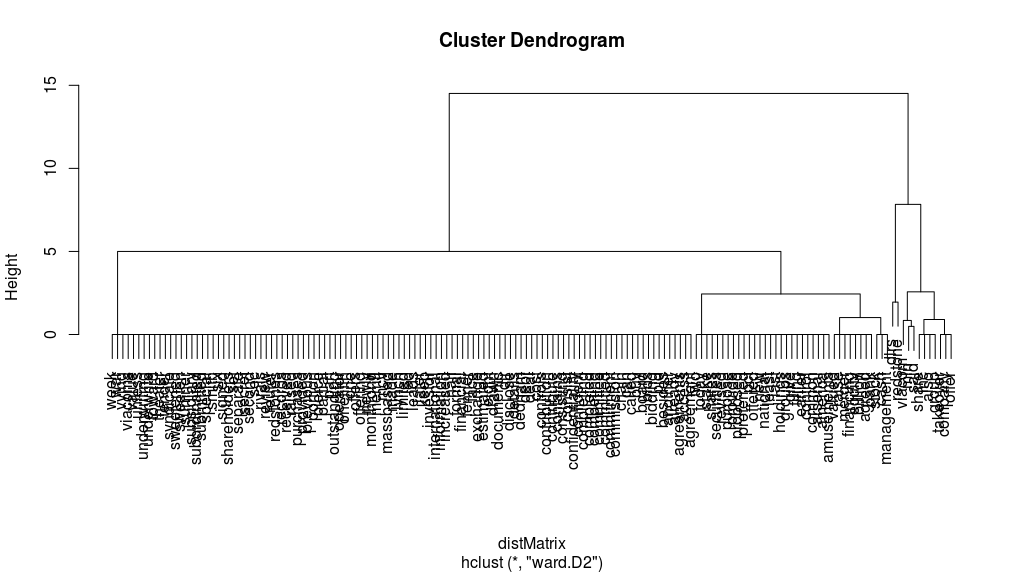
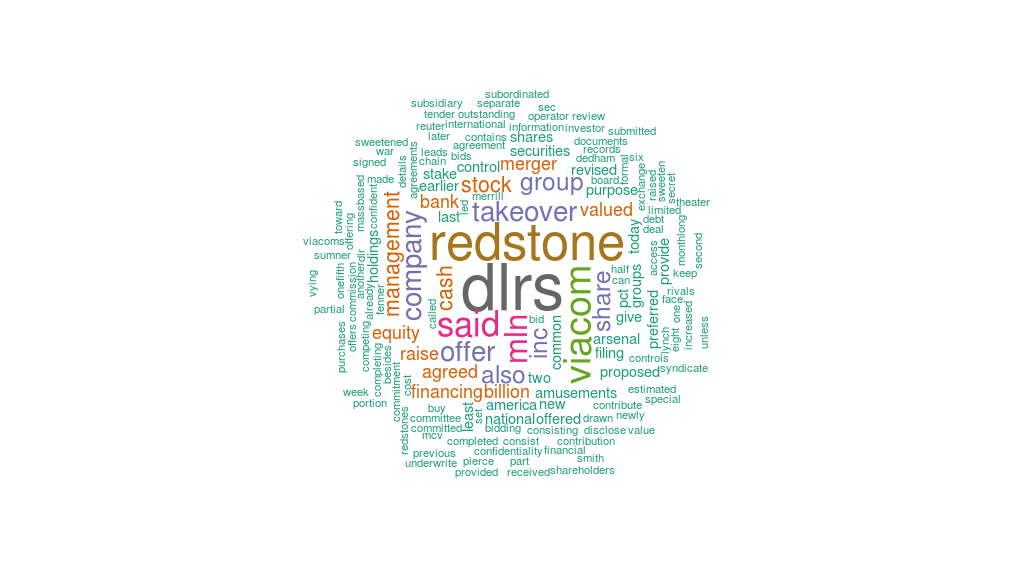
**362:**

****

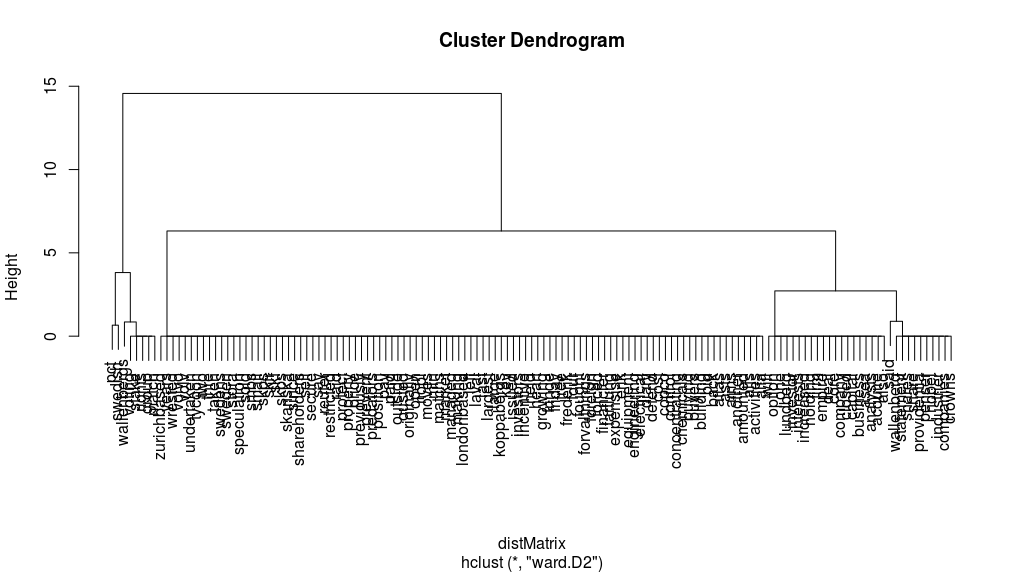
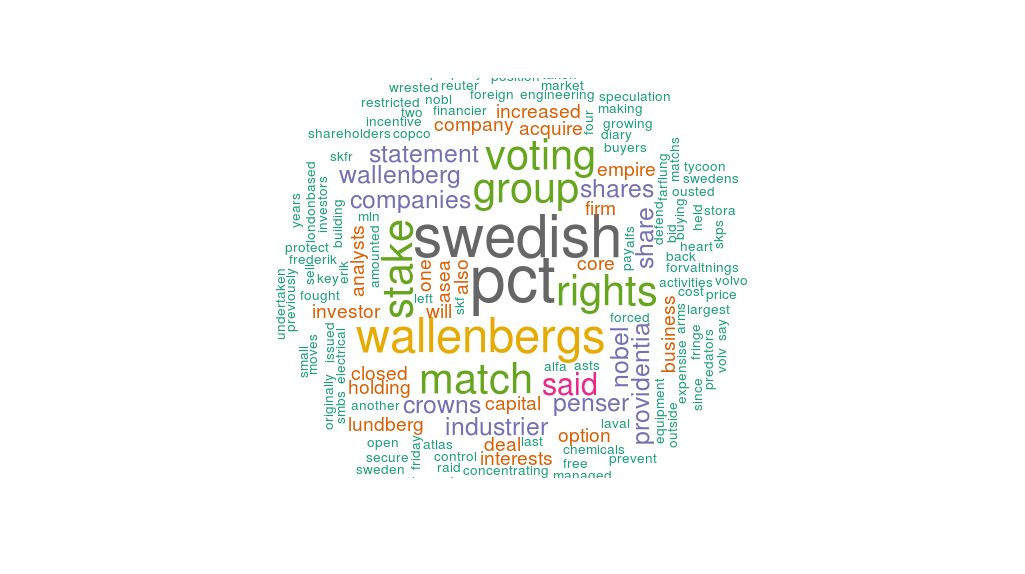
**372:**

****

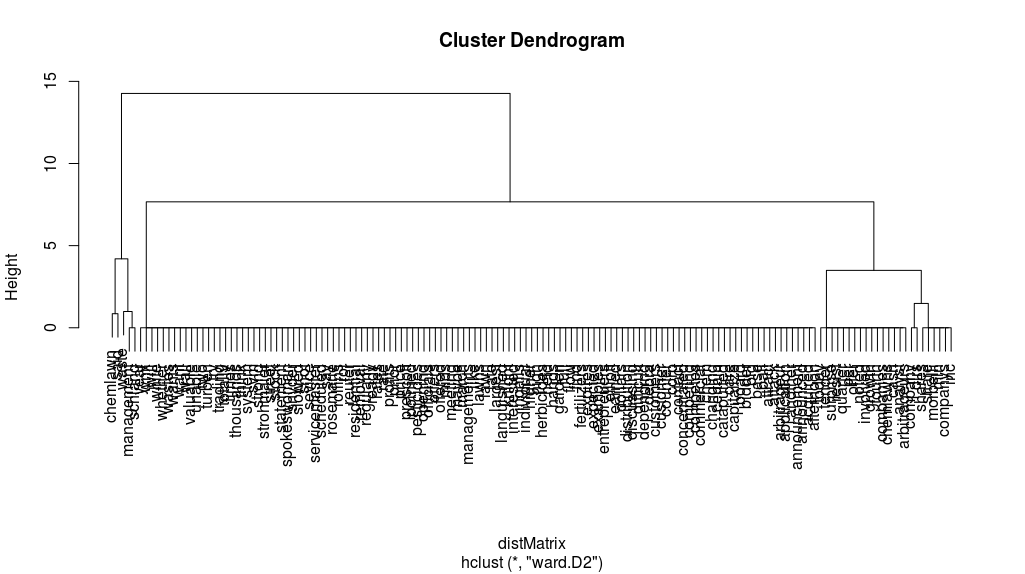
**496:**

****

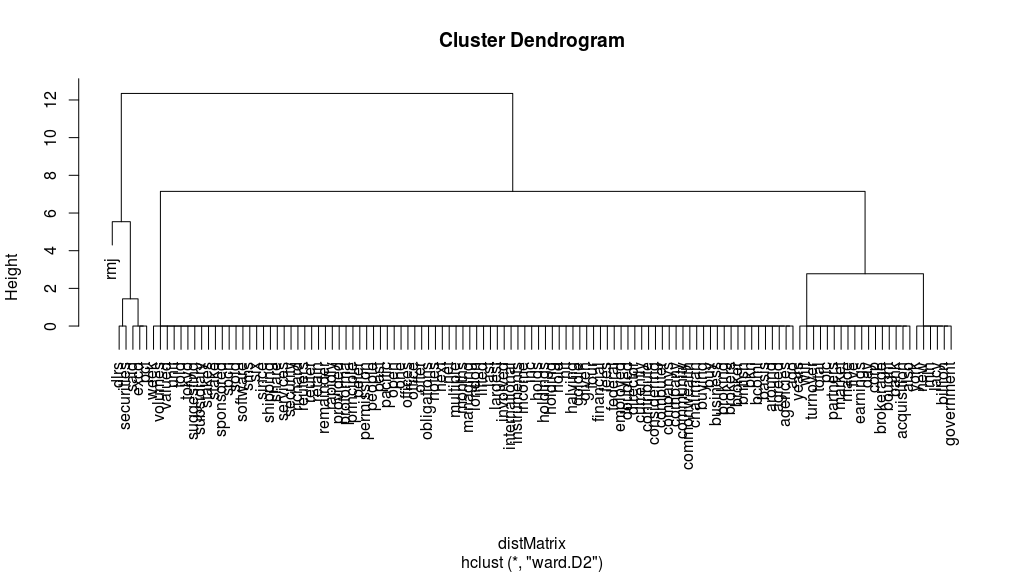
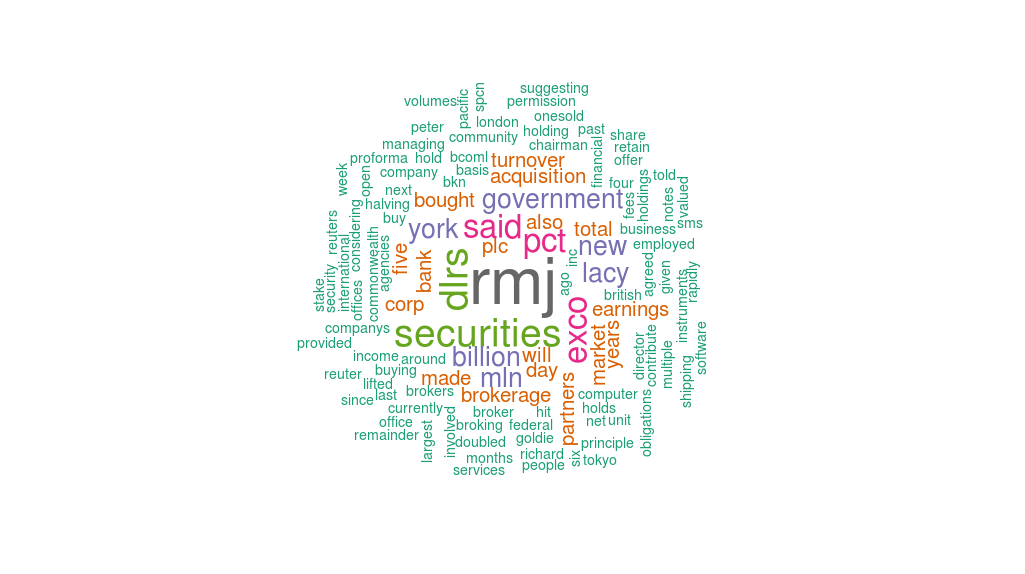
**302:**

****

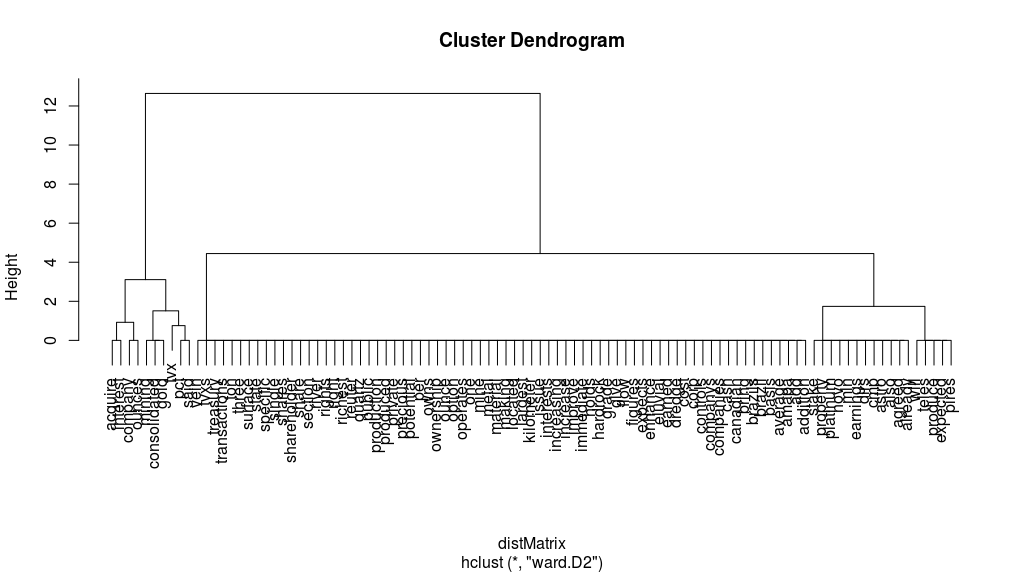
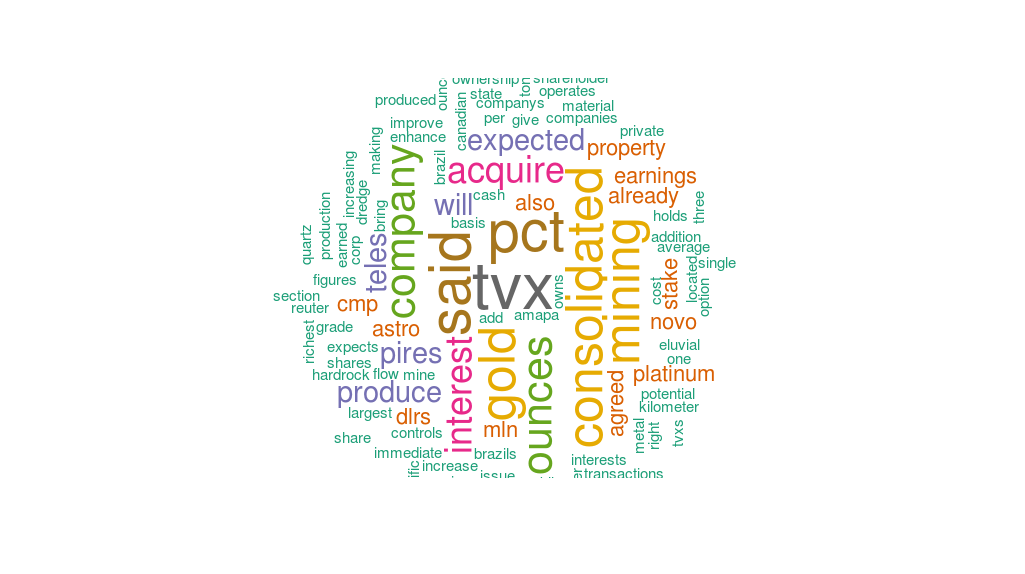
**45:**

****

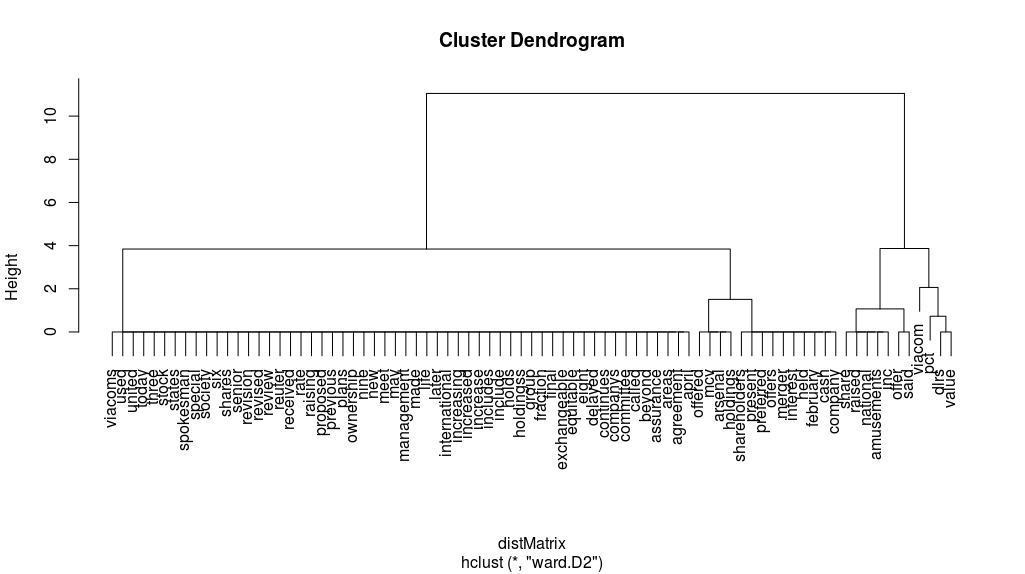
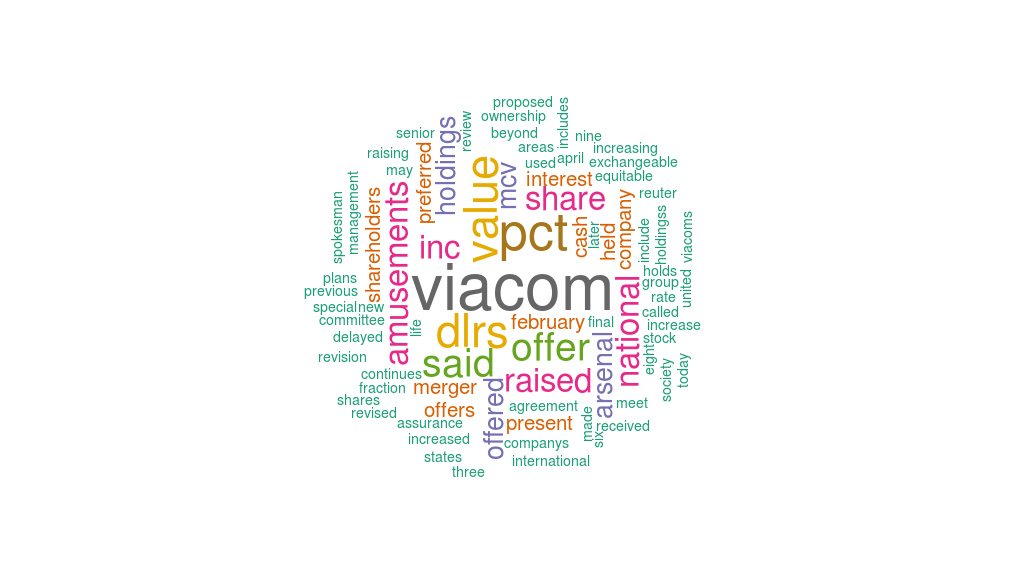
**331:**

****

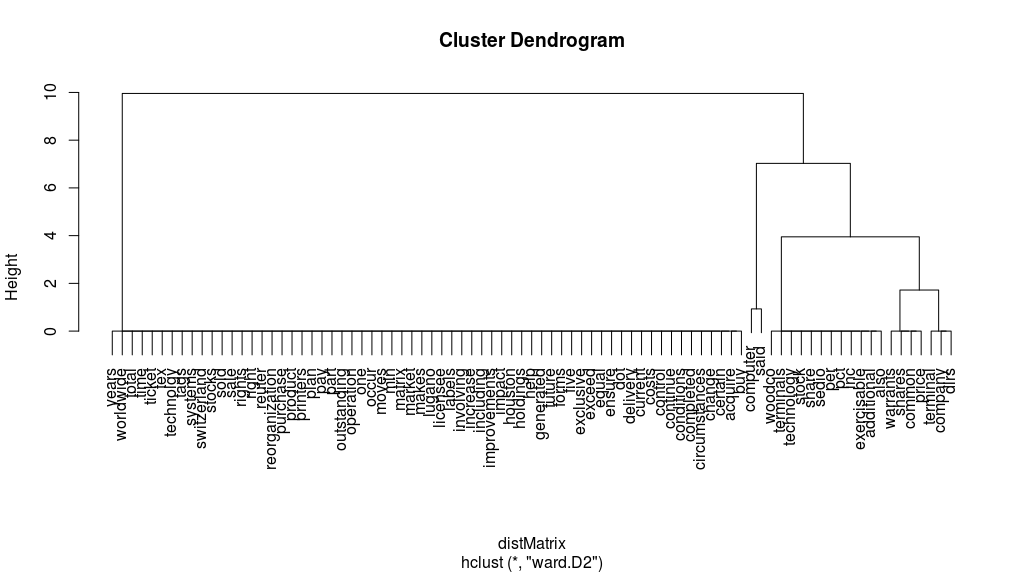
**448:**

****

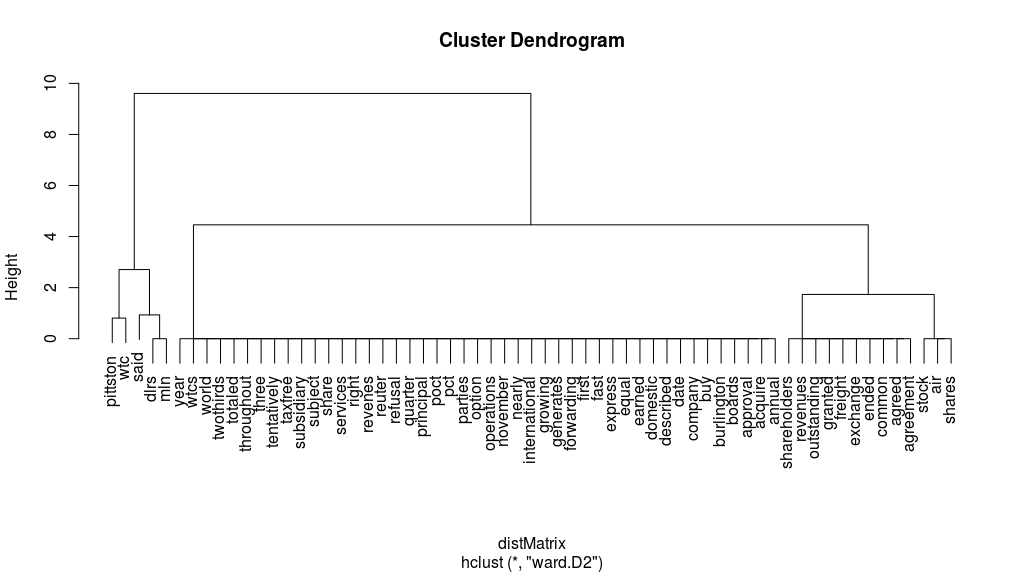
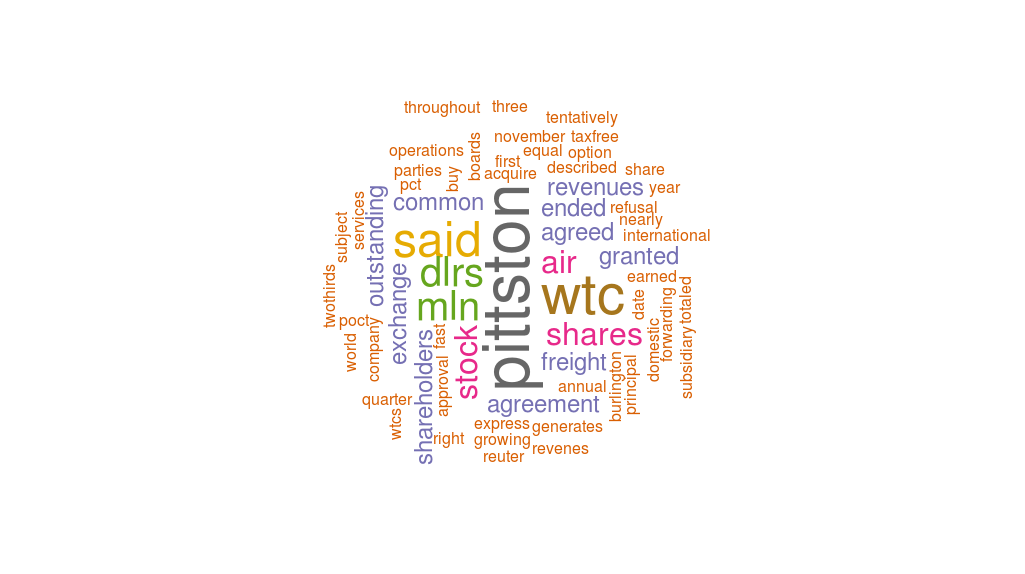
**393:**

****

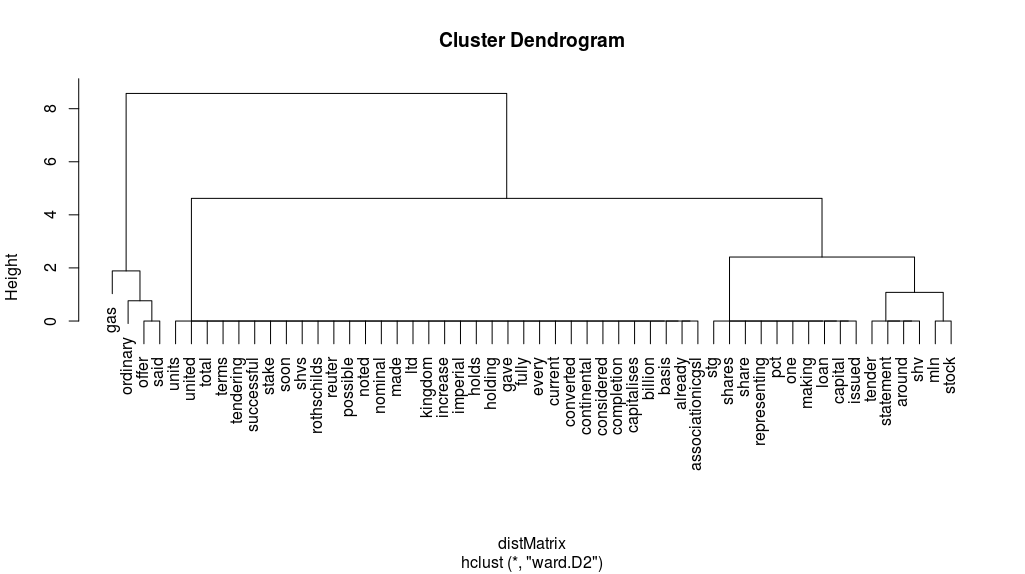
**10:**

****

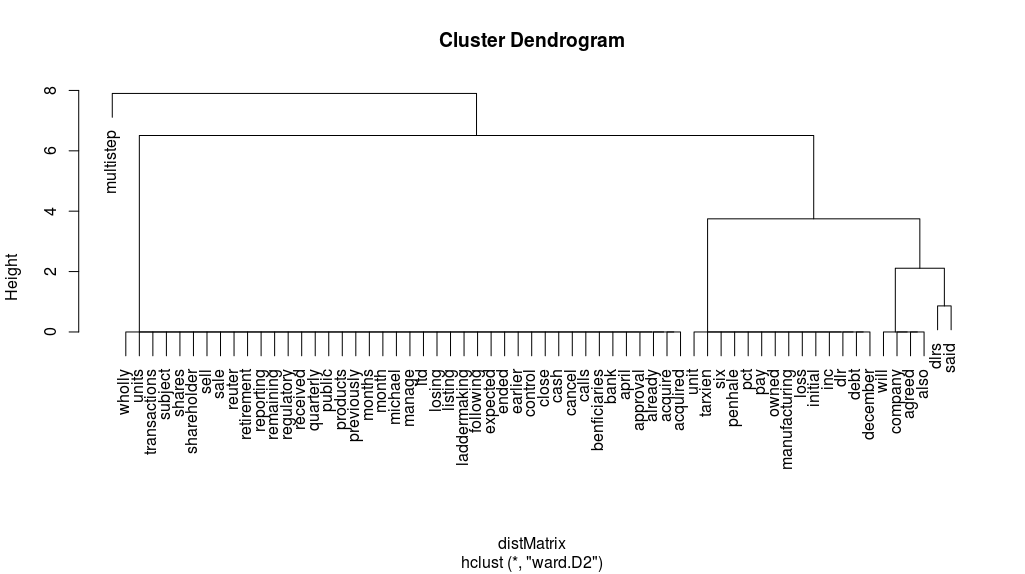
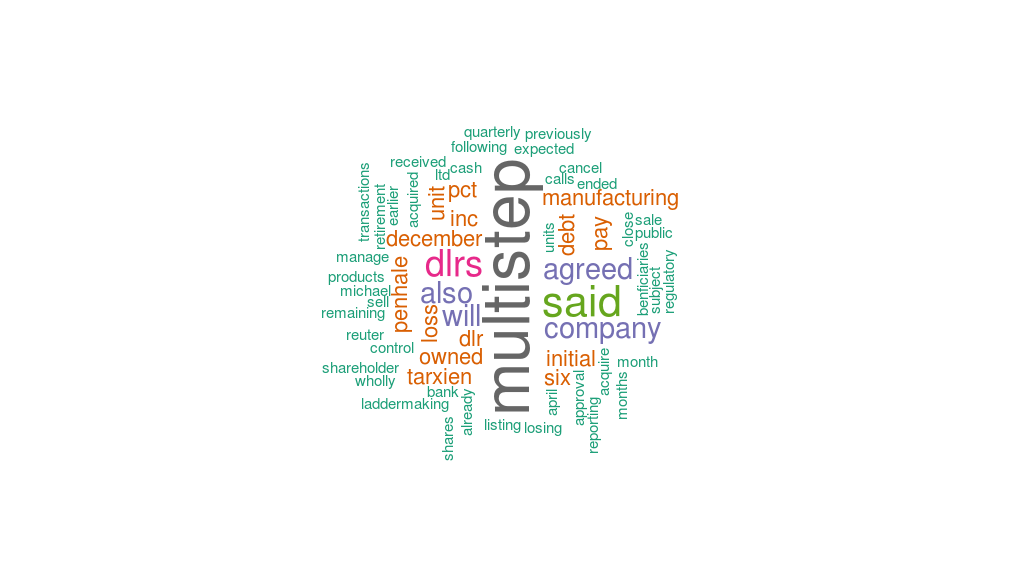
**408:**

****

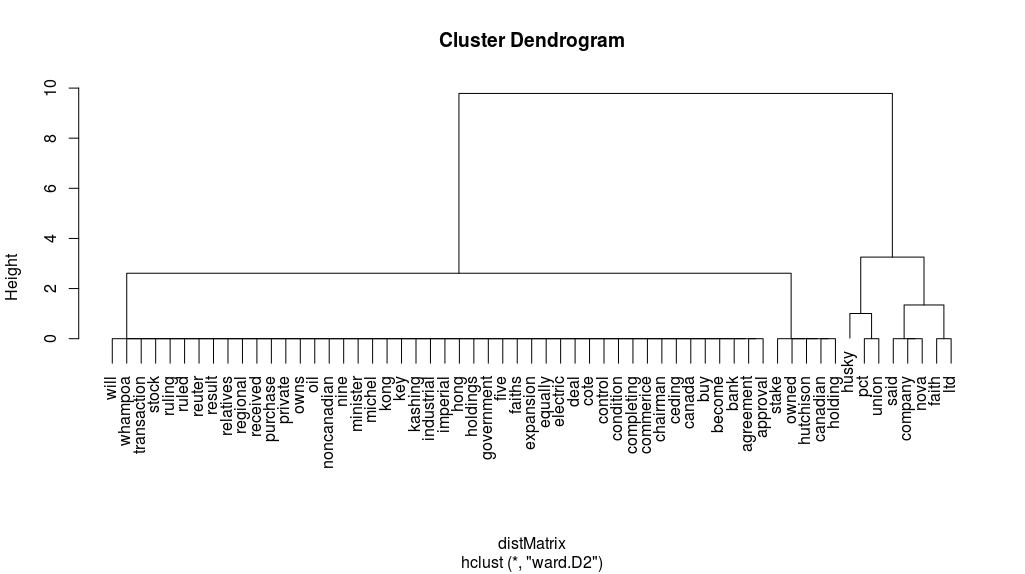
**304:**

****

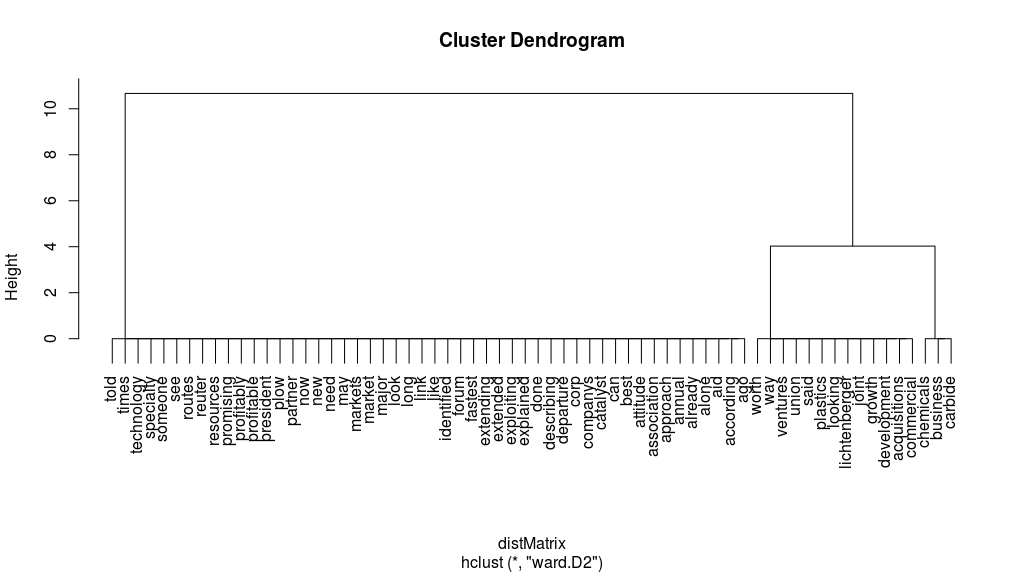
**473:**

****

**157:**

****

**504:**

****

Question d

**Longest Words**

**Program**

v = c(7,25,29,47,19,4,22,42,34,1,36,20,44,13,50)

for(n in v){

d = inspect(revs[[n]])

d = d[[1]]

a = tokenize\_words(d)

b = which.max(nchar(a))

c = a[b]

print(c)

}

**Result**

[1] "capitalization"

PlainTextDocument

Metadata: 15

Content: chars: 2591

[1] "reorganization"

PlainTextDocument

Metadata: 15

Content: chars: 2323

[1] "shareholders"

PlainTextDocument

Metadata: 15

Content: chars: 2176

[1] "confidentiality"

PlainTextDocument

Metadata: 15

Content: chars: 1806

[1] "concentrating"

PlainTextDocument

Metadata: 15

Content: chars: 1692

[1] "servicemaster"

PlainTextDocument

Metadata: 15

Content: chars: 1294

[1] "international"

PlainTextDocument

Metadata: 15

Content: chars: 1225

[1] "consolidated"

PlainTextDocument

Metadata: 15

Content: chars: 1079

[1] "international"

PlainTextDocument

Metadata: 15

Content: chars: 961

[1] "reorganization"

PlainTextDocument

Metadata: 15

Content: chars: 765

[1] "international"

PlainTextDocument

Metadata: 15

Content: chars: 735

[1] "representing"

PlainTextDocument

Metadata: 15

Content: chars: 762

[1] "manufacturing"

PlainTextDocument

Metadata: 15

Content: chars: 635

[1] "transaction"

PlainTextDocument

Metadata: 15

Content: chars: 710

[1] "lichtenberger"

**Longest sentence**

**Program**

**//**Must make sure your data not use **lowercase** and **remove punctuation**

1.We are not sure about the definition of the longest sentence. We choose the sentence **has most characters** as longest sentence in first result.

v = c(7,25,29,47,19,4,22,42,34,1,36,20,44,13,50)

for(n in v){

d = data.frame(text=unlist(sapply(myCorpus[n], `[`, "content")), stringsAsFactors=F)

d = d[[1]]

a = tokenize\_sentences(d)

b = which.max(nchar (a))

c = a[b]

print(c)

}

**Result**

[1] "american express co remained silent on market rumors it would spinoff all or part of its shearson lehman brothers inc but some analysts said the company may be considering such a move because it is unhappy with the market value of its stock"

[1] "in a joint statement american express and shearson said the actions under consideration are an integral part of american express worldwide financial services strategy and that the two companies have been having both internal and external discussions on the matters"

[1] "if all the shares of purolator are tendered shareholders would receive for each share 29 dlrs cash six dlrs in debentures and a warrant to buy shares in a subsidiary of pc acquisition containing the u s courier operations"

[1] "the redstone group which has a 19 5 pct stake in viacom and the management group which has a 5 4 pct stake have both agreed not to buy more shares of the company until a merger is completed unless the purchases are part of a tender offer for at least half of the outstanding stock"

[1] "but analysts say the wallenbergs position in the electrical engineering firm asea ab asea st is also too small at 12 6 pct of the voting rights and there has been growing speculation that the group will be forced to sell off fringe interests to protect its core activities"

[1] "both schlang and morbelli noted that high growth rates had catapulted chemlawn s share price into the mid 30 s in 1983 but the stock languished as the rate of growth slowed"

[1] "exco international plc a subsidiary of british and commonwealth shipping co plc bcom l said it had agreed in principle to buy an 80 pct stake in rmj holdings corp for about 79 mln dlrs"

[1] "consolidated tvx mining corp said it agreed to issue 7 8 mln treasury shares to acquire interests in three gold mining companies in brazil and an option to increase the company s interest in a platinum property"

[1] "viacom said mcv holdings a group which includes the company s senior management and the equitable life assurance society of the united states raised the value of its offer by increasing the value of the preferred being offered to 8 50 dlrs from 8 00 dlrs a share and raising the ownership in the new company to be held by present viacom shareholders to 45 pct from 25 pct"

[1] "computer terminal said sedio also has the right to buy additional shares and increase its total holdings up to 40 pct of the computer terminal s outstanding common stock under certain circumstances involving change of control at the company"

[1] "its revenues totaled nearly 200 mln dlrs in the year ended november 30 and for the quarter ended on that date it earned 1 3 mln dlrs on revenues of 55 8 mln dlrs"

[1] "successful completion of the offer would increase shv s stake in ic gas to 39 8 mln shares representing around 27 9 pct of issued share capital it said"

[1] "multi step products inc earlier reporting an initial six month loss said it agreed to sell wholly owned multi step manufacturing inc for 100 000 dlrs cash subject to shareholder and regulatory approval"

[1] "nova the canadian company that owns 56 pct of husky oil ltd said it received government approval for a transaction under which union faith canada holding ltd would buy a 43 pct stake in husky"

[1] "describing this as a major departure in the company s approach to commercial development he told the annual new business forum of the commercial development association we are looking to acquisitions and joint ventures when they look like the fastest and most promising routes to the growth markets we ve identified"

2.If we use the number of words in a sentence as the length of sentence

**Program**

v = c(7,25,29,47,19,4,22,42,34,1,36,20,44,13,50)

list = c()

for(n in v){

d = data.frame(text=unlist(sapply(myCorpus[n], `[`, "content")), stringsAsFactors=F)

d = d[[1]]

a = tokenize\_sentences(d)

for (i in a){

word = tokenize\_words(i)

list = c(list, length(word))

}

b = which.max(list)

c = a[b]

print(c)

list = c()

}

**Result**

[1] "american express co remained silent on market rumors it would spinoff all or part of its shearson lehman brothers inc but some analysts said the company may be considering such a move because it is unhappy with the market value of its stock"

[1] "american express co rumored to be considering a spinoff of part of shearson lehman brothers inc said it is studying a range of options for its brokerage unit that could improve shearon s access to capital and help it meet broadening international competition"

[1] "if all the shares of purolator are tendered shareholders would receive for each share 29 dlrs cash six dlrs in debentures and a warrant to buy shares in a subsidiary of pc acquisition containing the u s courier operations"

[1] "the redstone group which has a 19 5 pct stake in viacom and the management group which has a 5 4 pct stake have both agreed not to buy more shares of the company until a merger is completed unless the purchases are part of a tender offer for at least half of the outstanding stock"

[1] "but analysts say the wallenbergs position in the electrical engineering firm asea ab asea st is also too small at 12 6 pct of the voting rights and there has been growing speculation that the group will be forced to sell off fringe interests to protect its core activities"

[1] "shares of chemlawn shot up 11 5 8 to 29 3 8 in over the counter trading with 3 8 mln of the company s 10 1 mln shares changing hands by late afternoon"

[1] "exco international plc a subsidiary of british and commonwealth shipping co plc bcom l said it had agreed in principle to buy an 80 pct stake in rmj holdings corp for about 79 mln dlrs"

[1] "consolidated tvx mining corp said it agreed to issue 7 8 mln treasury shares to acquire interests in three gold mining companies in brazil and an option to increase the company s interest in a platinum property"

[1] "viacom said mcv holdings a group which includes the company s senior management and the equitable life assurance society of the united states raised the value of its offer by increasing the value of the preferred being offered to 8 50 dlrs from 8 00 dlrs a share and raising the ownership in the new company to be held by present viacom shareholders to 45 pct from 25 pct"

[1] "computer terminal said sedio also has the right to buy additional shares and increase its total holdings up to 40 pct of the computer terminal s outstanding common stock under certain circumstances involving change of control at the company"

[1] "its revenues totaled nearly 200 mln dlrs in the year ended november 30 and for the quarter ended on that date it earned 1 3 mln dlrs on revenues of 55 8 mln dlrs"

[1] "it said in a statement the offer was on the basis of 700p for each ic gas ordinary and 252p for every one stg nominal of ic gas loan stock"

[1] "multi step products inc earlier reporting an initial six month loss said it agreed to sell wholly owned multi step manufacturing inc for 100 000 dlrs cash subject to shareholder and regulatory approval"

[1] "nova the canadian company that owns 56 pct of husky oil ltd said it received government approval for a transaction under which union faith canada holding ltd would buy a 43 pct stake in husky"

[1] "describing this as a major departure in the company s approach to commercial development he told the annual new business forum of the commercial development association we are looking to acquisitions and joint ventures when they look like the fastest and most promising routes to the growth markets we ve identified"

Question e

1. If we use the number of characters as the length of sentence

**Program**

v = c(7,25,29,47,19,4,22,42,34,1,36,20,44,13,50)

list = c()

for(n in v){

d = data.frame(text=unlist(sapply(myCorpus[n], `[`, "content")), stringsAsFactors=F)

d = d[[1]]

a = tokenize\_sentences(d)

for (i in a){

c = nchar(i)

list = c(list, c)

}

print(list)

list = c()

}

**Result**

[1] 240 186 111 89 70 146 152 138 74 121 178 115 86 125 168 134 90 49 128 87 59 129 34 129 0 74 194 69 164 106 6

[2] 258 264 164 205 124 154 149 169 103 193 178 82 143 92 54 132 84 86 173 101 233 51 171 6

[3] 143 71 65 131 43 54 153 161 221 205 78 55 116 47 131 89 66 168 3 73 124 87 139 71 46 67 183 172 6

[4] 199 233 100 224 241 134 280 148 163 235 222 93 154 170 226 61 6

[5] 202 224 172 78 109 236 108 236 218 170 165 164 274 6

[6] 142 150 141 83 72 113 79 105 88 53 96 106 81 78 138 49 158 172 73 97 112 6

[7] 190 184 119 67 29 87 16 48 140 40 142 120 179 73 86 145 112 6

[8] 212 206 41 167 166 149 149 167 114 169 6

[9] 117 86 151 52 239 158 134 371 103 6

[10] 205 102 240 182 174 97 134 94 6

[11] 107 116 106 69 96 72 113 161 155 6

[12] 142 8 138 117 151 54 129 41 11 92 61 6

[13] 203 146 83 70 59 78 113 113 103 6

[14] 193 191 62 93 131 138 6

[15] 123 49 315 106 70 87 137 129 6

2.If we use the number of words in a sentence as the length of sentence

**Program**

v = c(7,25,29,47,19,4,22,42,34,1,36,20,44,13,50)

list = c()

for(n in v){

d = data.frame(text=unlist(sapply(myCorpus[n], `[`, "content")), stringsAsFactors=F)

d = d[[1]]

a = tokenize\_sentences(d)

for (i in a){

word = tokenize\_words(i)

list = c(list, length(word))

}

print(list)

list = c()

}

**Result**

[1] 43 31 19 20 12 23 23 22 12 20 30 20 14 25 31 20 17 12 22 14 9 24 5 22 0 12 31 13 27 19 1

[2] 43 40 27 35 20 24 26 35 17 36 30 15 23 17 10 22 12 13 27 16 36 11 27 1

[3] 27 13 12 20 9 10 29 28 39 36 15 11 24 8 23 14 12 33 2 13 23 14 25 14 8 11 33 26 1

[4] 38 41 20 37 40 24 56 22 31 41 36 16 30 28 44 12 1

[5] 37 38 31 15 20 48 19 41 38 29 28 27 49 1

[6] 24 34 26 13 11 17 16 18 16 10 17 20 12 13 22 8 24 32 12 21 16 1

[7] 35 35 24 14 3 15 2 10 27 9 27 23 30 14 16 27 22 1

[8] 37 33 7 30 33 30 32 32 21 32 1

[9] 17 15 25 10 48 33 22 69 17 1

[10] 38 19 39 37 29 17 22 13 1

[11] 20 22 19 12 19 12 17 34 25 1

[12] 25 2 30 22 29 10 25 9 1 18 12 1

[13] 33 28 11 15 9 14 23 25 19 1

[14] 35 33 12 15 24 26 1

[15] 21 6 51 22 14 16 28 22 1

**Functions:**

Install.packages(“tm”)

library(“tm”)

Install.packages(“textreuse”)

Install.packages(“wordnet”)

data(acq)

tm = DocumentTermMatrix(acq)

myCorpus = tm\_map(acq, content\_transformer(tolower))

myCorpus = tm\_map(myCorpus, removeWords, stopwords("english"))

myCorpus = tm\_map(myCorpus, removePunctuation)

myCorpus = tm\_map(myCorpus, removeNumbers)

myCorpus = tm\_map(myCorpus, stripWhitespace)

d<-data.frame(text=unlist(sapply(myCorpus[1], `[`, "content")), stringsAsFactors=F)

a = d[sapply(d, function(x) max(nchar(x)))>1]

sapply(d, as.character)

install.packages("tokenizers")

library("tokenizers")

data.frame(text=unlist(sapply(d, `[`, "content")), stringsAsFactors=F)

tokenize\_words(e)

**f.**

data(acq)

tm = DocumentTermMatrix(acq)

myCorpus = tm\_map(acq, content\_transformer(tolower))

myCorpus = tm\_map(myCorpus, removeWords, stopwords("english"))

myCorpus = tm\_map(myCorpus, removePunctuation)

myCorpus = tm\_map(myCorpus, removeNumbers)

myCorpus = tm\_map(myCorpus, stripWhitespace)

d<-data.frame(text=unlist(sapply(myCorpus[1], `[`, "content")), stringsAsFactors=F)

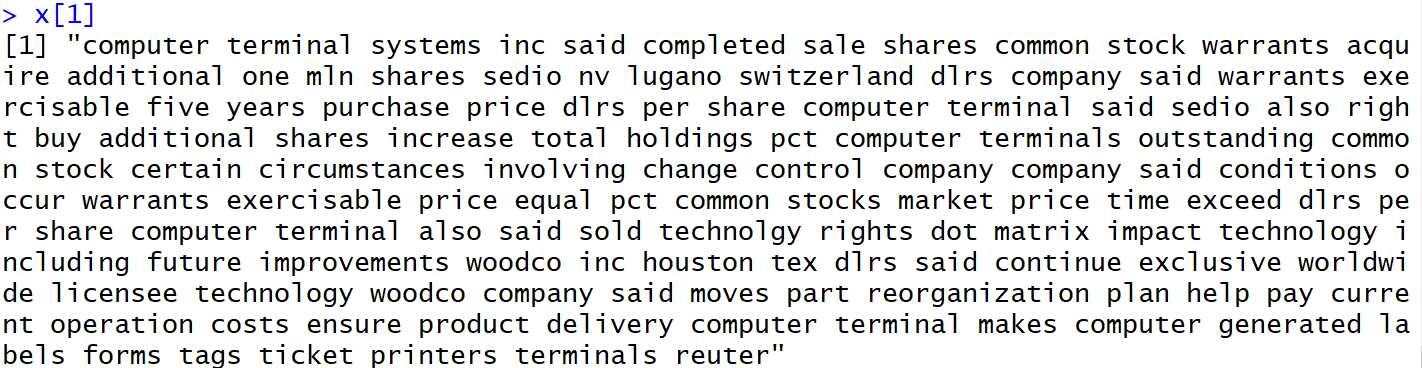
tmp <- sapply(d, as.character) # a character vector

m <- matrix(c(tmp, sapply(d, sub, pattern = "(.)(.)", replacement = "\\2\\1")),

2, byrow = TRUE)

x = unlist(m)

The display result after removing the punctuation is:

****

**g.**

install.packages("wordnet")

setDict("C:/Program Files (x86)/WordNet/2.1/dict")

Sys.setenv(WNHOME = "C:/Program Files (x86)/WordNet/2.1")

library(wordnet)

install.packages("tokenizers")

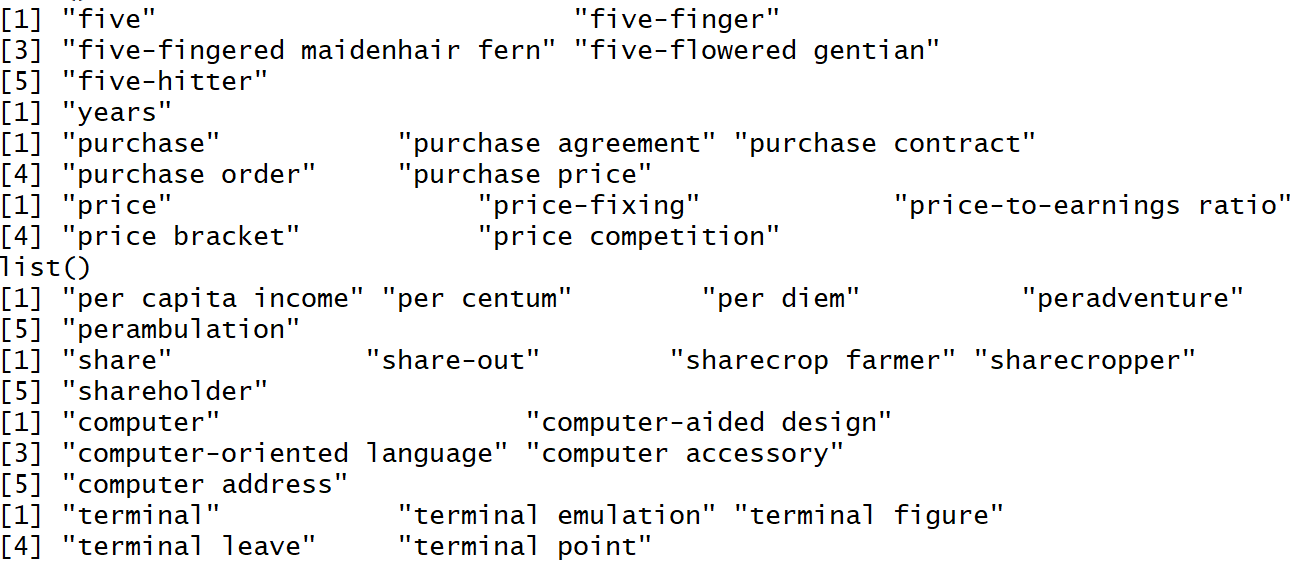
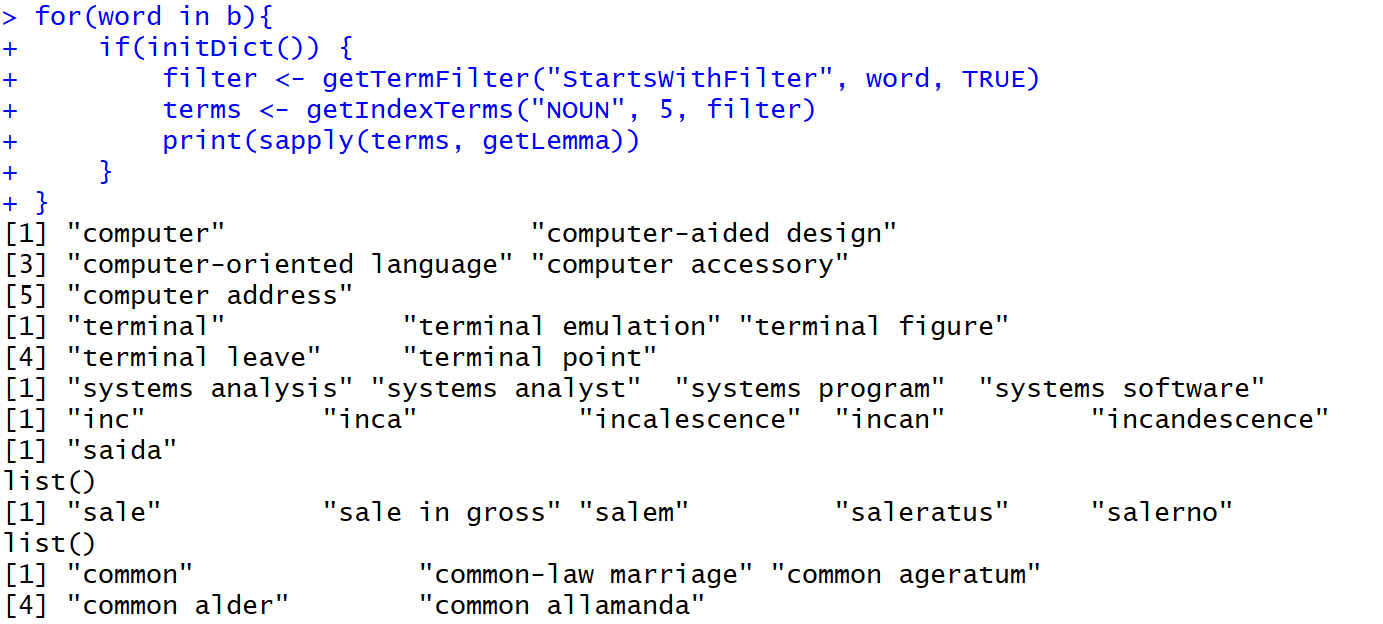
library("tokenizers")

e<-data.frame(text=unlist(sapply(d, `[`, "content")), stringsAsFactors=F)

a = tokenize\_words(e)

b = a[[1]][1]

The part of speech for each word print using the **Wordnet** package is:



H：

install.packages("zipfR", repos="http://R-Forge.R-project.org")

install.packages("tm")

library(“tm”)

library(“zipfR”)

revs <- tm\_map(acq, content\_transformer(tolower))

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

revs <- tm\_map(revs, removePunctuation)

revs <- tm\_map(revs, removeNumbers)

revs <- tm\_map(revs, stripWhitespace)

myTdm <- as.matrix(TermDocumentMatrix(revs))

FreqMat <- data.frame(ST = rownames(myTdm), Freq = rowSums(myTdm), row.names = NULL)

FreqMat <- spc(Vm=FreqMat$Freq, m=FreqMat$ST)

plot(FreqMat, log="x")

