

Group-4-Project-3 report

Yi Qian; Kailing Huang; Tarek Hatata; Shusen Wu; Bohui NI

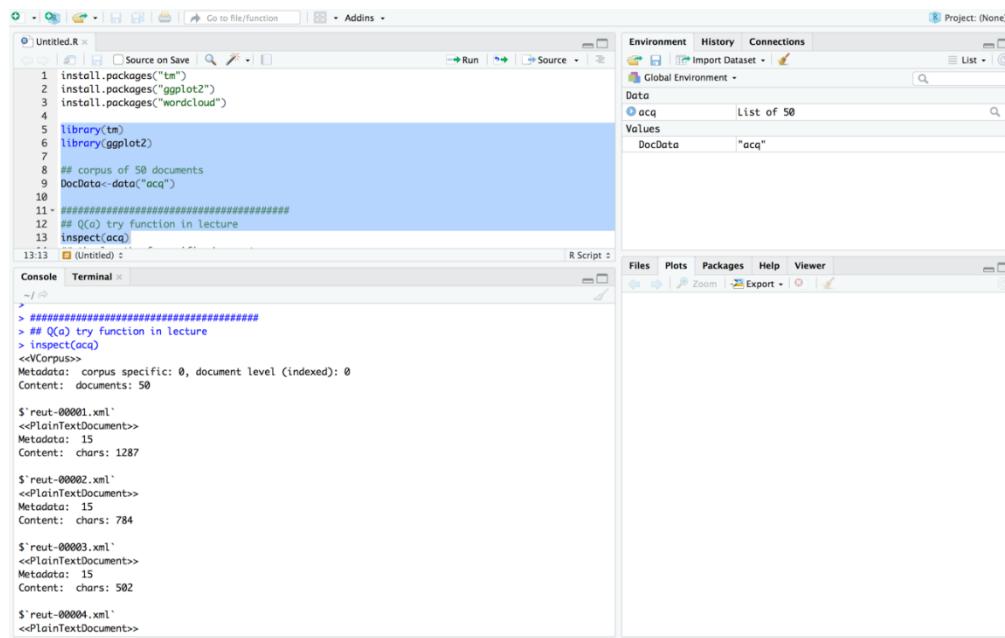
1. R functions:

Inspect()
DocumentTermMatrix()
termFreq()
as.data.frame()
tm_map()
content_transformer()
tm_map()
list()
findFreqTerms()
findAssocs()
rowSums()
as.matrix()
subset()
removeSparseTerms()
dist()
scale()
hclust()
sort()
brewer.pal()
wordcloud()
as.character()
nchar()
tokenize_words()
cbind()
tokenize_sentences()
sapply()

2. Presentation and discussion of results from the experiments that you run using the different functions from Lecture 7: parts (a) through (h):

- a. For the complete set of documents, try the functions in lecture 9. What happens? Does it yield anything understandable about the documents.

1. Inspect()



The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled "Untitled.R" containing the following code:

```

1 install.packages("tm")
2 install.packages("ggplot2")
3 install.packages("wordcloud")
4
5 library(tm)
6 library(ggplot2)
7
8 ## corpus of 50 documents
9 DocBeta<-data("acq")
10
11 #####
12 ## Q(c) try function in lecture
13 inspect(acq)

```

In the bottom-left pane, the "Console" tab is active, showing the output of the `inspect` command:

```

> #####
> ## Q(c) try function in lecture
> inspect(acq)
<--Corpus>
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 50

$'reut-00001.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 1287

$'reut-00002.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 784

$'reut-00003.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 502

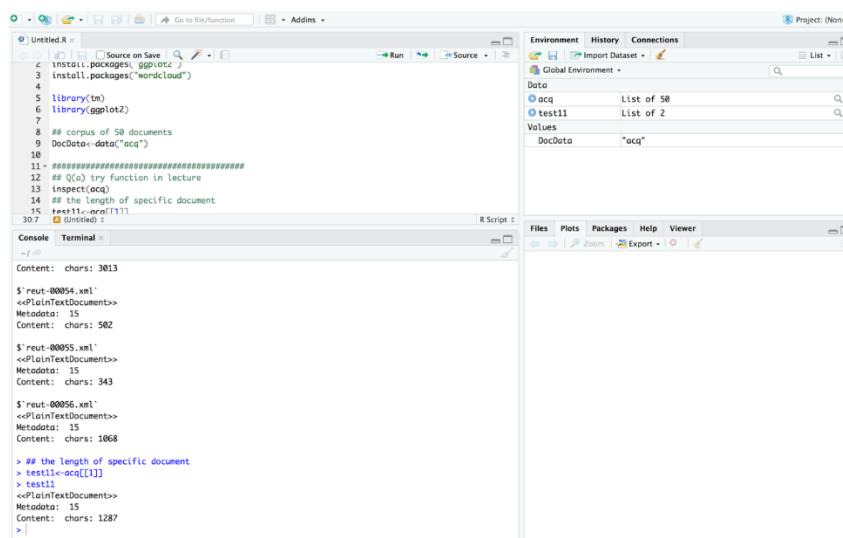
$'reut-00004.xml'
<>PlainTextDocument>>

```

The right side of the interface shows the "Environment" tab, where the variable `acq` is listed under "Data". The "Values" section shows `DocData` with the value "acq".

Yes, from this function, we can find the induction of the document.

2. We can find the length of specific document



The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled "Untitled.R" containing the following code:

```

1 install.packages("ggplot2")
2 install.packages("wordcloud")
3
4 library(tm)
5 library(ggplot2)
6
7 ## corpus of 50 documents
8 DocBeta<-data("acq")
9
10 #####
11 ## Q(c) try function in lecture
12 inspect(acq)
13
14 ## the length of specific document
15 test1<-acq[1]

```

In the bottom-left pane, the "Console" tab is active, showing the output of the code:

```

Content: chars: 3013

$'reut-00054.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 582

$'reut-00055.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 343

$'reut-00056.xml'
<>PlainTextDocument>>
Metadata: 15
Content: chars: 1068

> ## the length of specific document
> test1<-acq[1]
> test1
<>PlainTextDocument>>
Metadata: 15
Content: chars: 1287
>

```

The right side of the interface shows the "Environment" tab, where the variable `acq` is listed under "Data". The "Values" section shows `DocData` with the value "acq".

3. We can find sparsity or max length term

The screenshot shows the RStudio interface with the following details:

- Code Editor:** An R script titled "Untitled.R" containing code to inspect document length and term frequency.
- Console Output:**

```

<<PlainTextDocument>>
Metadata: 15
Content: chars: 343

$'reut-00056.xml'
<<PlainTextDocument>>
Metadata: 15
Content: chars: 1068

> ## the length of specific document
> test11<-acq[[1]]
> test11
<<PlainTextDocument>>
Metadata: 15
Content: chars: 1287

> ## sparsity/ Max length term
> ACQdtm<-DocumentTermMatrix(acq)
> ACQdtm
> ## inspect term
21 inspect(ACQdtm[1:15, 1:6])
22 ## frequency of term
23 test11f <- termFreq(test11)
24 test11f
25 ## convert to a DataFrame
26 test11df <- as.data.frame(test11f)
    
```
- Environment View:** Shows objects in the global environment: acq (List of 50), ACQdtm (List of 6), test11 (List of 2), and DocData ("acq").
- Files View:** Shows tabs for Files, Plots, Packages, Help, and Viewer.

4. We can inspect term

The screenshot shows the RStudio interface with the following details:

- Code Editor:** An R script titled "Untitled.R" containing code to inspect document length and term frequency.
- Console Output:**

```

Sparsity : 96%
Maximal term length: 21
Weighting : term frequency (tf)
> ## inspect term
> inspect(ACQdtm[1:15, 1:6])
> inspect(ACQdtm[1:15, 1:6])
<<DocumentTermMatrix (documents: 15, terms: 6)>>
Non-/sparse entries: 3/87
Sparsity : 97%
Maximal term length: 11
Weighting : term frequency (tf)
Sample :
  Terms
Docs "(american) "...that" "any" "bridge" "final" "it"
10      0      0      0      0      0      0
110     0      1      0      0      0      1
12      0      0      0      0      0      0
125     0      0      0      0      0      0
128     0      0      0      0      0      0
134     0      0      0      0      0      0
44      0      0      0      0      0      0
45      0      0      1      0      0      0
68      0      0      0      0      0      0
96      0      0      0      0      0      0
    
```
- Environment View:** Shows objects in the global environment: acq (List of 50), ACQdtm (List of 6), test11 (List of 2), and DocData ("acq").
- Files View:** Shows tabs for Files, Plots, Packages, Help, and Viewer.

5. We can find frequency of term

The screenshot shows the RStudio interface with the following code in the script pane:

```

19  ACQdtm
20 ## inspect term
21 inspect(ACQdtm[1:15, 1:6])
22 ## frequency of term
23 test1tf <- termFreq(test11)
24 test1tf
25 ## convert to a dataFrame
26 test1df <- as.data.frame(test1tf)
22.21 (Untitled) : 

```

The console pane displays the output of the termFreq() function:

```

> test1tf <- termFreq(test11)
> test1tf
   .125      1.50    200,000    50,000    <sedio    <woodco
   1          1        2          1        1          1
acquire    additional    also    and    any    are
   1          2        2          6        1          1
but,       buy    certain    change    circumstances    common
   1          1        1        1        1          3
company    company.    completed    computer    conditions    continue
   3          1        1        6        1          1
control    costs    current    delivery,    dirs,    dlr.s
   1          1        1        1        2          2
dot,       ensure    equal    exceed    exclusive    exercisable
   1          1        1        1        1          2
five,      for    forms,    future    generated    has
   4          1        1        1        1          2
help,      holdings    houston,    impact    improvements,    inc
   1          1        1        1        1          1
inc>,     including    increase    involving    its    labels,
   1          1        1        1        5          1
licensee    lugano,    makes    market    matrix    min
   1          1        1        1        1          1
moves,      n.v.>    not    occur    one    operation
   1          1        1        1        1          1
outstanding    part    pay    pct    per    plan
   1          1        1        2        2          1
price,      printers    product    purchase    reorganization    reuter
   3          1        1        1        1          1
right,      rights    said    sale    sedio    share.
   1          1        7        1        1          2

```

The environment pane shows the objects defined:

- acq
- ACQdtm
- test11
- test1tf

6. Convert to data frame

The screenshot shows the RStudio interface with the following code in the script pane:

```

24 test1tf
25 ## convert to a DataFrame
26 test1df <- as.data.frame(test1tf)
27 test1df
28 ## Convert the corpus to lower case
29 ACQlow<- tm_map(acq, content_transformer(tolower))
30 ACQlow
26.1 (Untitled) : 

```

The console pane displays the output of as.data.frame() function:

```

> test1df <- as.data.frame(test1tf)
> test1df
   test1tf
   .125      1
1.50      1
200,000    2
50,000      1
<sedio      1
<woodco    1
acquire     1
additional   2
also,       2
and,       6
any,       1
are,       1
but,       1
buy,       1
certain     1
change,    1
circumstances  1
common,    3
company,   3
company.,  1
completed, 1
computer,  6
conditions, 1
continue,  1
control,   1
costs,     1
current,   1
deliverv.  1

```

The environment pane shows the objects defined:

- acq
- ACQdtm
- test11
- test1df

7. Convert the corpus to lower case

The screenshot shows the RStudio interface with the following details:

- Code Editor:** The script pane contains R code for converting a corpus to lowercase. The highlighted line is `ACQlow <- tm_map(acq, content_transformer(tolower))`.
- Console:** The output shows the resulting document frequency matrix. The first few rows are:

Term	Document Frequency
stock	1
stock's	1
stock,	1
switzerland	1
systems	1
tags	1
technolgy	1
technology	1
technology,	1
terminal	4
terminal's	1
terminals.	1
tex.	1
the	15
ticket	1
time,	1
total	1
under	1
warrants	3
were	1
woodco.	1
worldwide	1
would	3
years	1
- Environment:** The global environment pane shows objects like `acq` (List of 50), `ACQdtm` (List of 6), and `ACQlow` (List of 50).

8. remove anything other than English letters or spaces

The screenshot shows the RStudio interface with the following details:

- Code Editor:** The script pane contains R code for removing non-English characters. The highlighted line is `removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)`.
- Console:** The output shows the resulting document frequency matrix. The first few rows are identical to the previous screenshot, indicating no change has been applied yet.
- Environment:** The global environment pane shows objects like `acq` (List of 50), `ACQcl` (List of 50), `ACQdtm` (List of 6), and `ACQlow` (List of 50).

9. remove stop words from the corpus

The screenshot shows the RStudio interface with the following components:

- Script Editor:** An R script titled "Untitled.R" is open, containing code for text processing. The code includes loading datasets, creating stop words, and generating term document matrices.
- Console:** The output of the R script is displayed, showing the frequency of words like "were", "woodco.", "worldwide", "would", and "years". It also shows the creation of a corpus and term document matrices.
- Environment Browser:** A list of objects in the global environment, including lists for ACQstop, ACQcl, ACQdtm, ACQlow, ACQtdm, and test11, along with their descriptions.
- Project:** A "Project: (None)" tab is visible at the top right.

10. find terms with a frequency of 5 or more

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays an R script named "Untitled.R" containing code related to document term matrices and word frequencies.
- Console:** Shows the output of the script, specifically the creation of a frequency matrix and the resulting term list.
- Environment:** Shows the global environment with various objects listed, such as ACQc1, ACQdtm, ACQlow, ACQstop, ACQtdm2, test11, and test1df.
- History:** Shows a history of previous sessions.
- Connections:** Shows connections to databases.
- Project:** Shows the current project is "(None)".

```
Untitled.R
Source on Save Run Source
37 inspect(ACQstop[1:2])
38 ## find terms with a frequency of 5 or more
39 ACQtdm2 <- TermDocumentMatrix(ACQstop, control = list(wordLengths = c(1, Inf)))
40 ACQtdm2
41 freq_terms <- findFreqTerms(ACQtdm2, lowfreq = 5)
42 freq_terms
43 ## find words associated with "states"
44 findfreqterms(ACQtdm2, "states", n = 25)
42:11 [1] "Untitled.R" R Script

Console Terminal
~/Desktop

> freq_terms <- findFreqTerms(ACQtdm2, lowfreq = 5)
> freq_terms
 [1] "ab"      "acquire"  "acquired" "acquisition" "acquisitions"
 [6] "added"   "agreed"   "agreement" "already"     "also"
[11] "american" "amusements" "analysts"  "another"    "approval"
[16] "around"   "arsenal"   "assets"    "b"          "bank"
[21] "barbara"  "bid"       "billion"   "board"      "bought"
[26] "brokerage" "burdett"   "business"  "buy"        "capital"
[31] "cash"     "certain"   "chemlown" "chief"      "circuit"
[36] "closed"   "co"        "commission" "common"     "companies"
[41] "company"  "companies" "completed" "completion" "computer"
[46] "considered" "considering" "consolidated" "control"    "corp"
[51] "courier"  "current"   "deal"      "debt"       "division"
[56] "dlr"      "dlrs"      "due"       "earlier"   "earnings"
[61] "ef"       "equity"    "esselte"   "exchange"  "expected"
[66] "express"  "february"  "filling"   "financial" "financing"
[71] "firm"     "first"     "five"      "four"      "friday"
[76] "gas"      "give"      "gold"      "government" "group"
[81] "growth"   "held"      "holding"   "holdings"   "hotel"
[86] "husky"    "button"    "ic"        "inc"       "increase"
[91] "industries" "interest"  "international" "investment" "issued"
[96] "lost"     "ltd"       "mode"      "management" "march"
[101] "market"   "match"     "may"       "meeting"   "merger"
[106] "mining"   "mln"       "multistep" "national"  "need"
[111] "net"      "new"       "now"       "offer"     "offered"
[116] "officer"  "one"       "operating" "operations" "option"
[121] "ordinary" "ounces"    "outstanding" "owned"     "owns"
[126] "part"     "pc"        "pct"      "penn"      "per"
[131] "pittston" "plan"      "plans"    "plc"       "position"
[136] "preferred" "president" "pretax"   "previously" "price"
```

11. find words associated with "states"

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Untitled.R contains R code to find words associated with "states".
- Console Output:**

```
[201] "wallenbergs" "warrants" "waste" "will" "worth"
[206] "wtc" "year" "years" "york"
> findAssocs(ACQtdm2, "states", .25)
```
- Data View:** A large table titled '\$states' is displayed, showing word frequencies. The table includes columns like 'areas', 'arranging', 'assurance', 'bankruptcy', 'bodies', 'charters', and 'continues'. Many words have a frequency of 0.70.
- Environment View:** Shows objects like ACQcl, ACQdtm, ACQlow, ACQstop, ACQtdm2, test11, and test1df.
- Files View:** Shows files like DocData, freq.terms, and myStopword.

12. term frequency

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Untitled.R contains R code to calculate term frequency.
- Console Output:**

```
> term.freq <- rowSums(as.matrix(ACQtdm2))
> term.freq <- subset(term.freq, term.freq >= 5)
> df <- data.frame(term = names(term.freq), freq = term.freq)
> term.freq
> df
```
- Data View:** A large table titled 'term.freq' is displayed, showing term frequencies. The table includes columns like 'ab', 'acquire', 'acquired', 'acquisition', 'acquisitions', 'added', and 'agreed'. Many terms have a frequency of 5 or higher.
- Environment View:** Shows objects like ACQstop, ACQtdm2, df, test11, and test1df.
- Files View:** Shows files like DocData, freq.terms, myStopword, and term.freq.

b. Find the 15 longest documents (in number of words).

```

## 15 largest document
## 50:1068, 47:3013, 44:1022, 42:1607, 36:1043, 34:1465, 29:3109, 25:3516, 22:1873, 20:1009,
## 19:2457, 18:871, 7:3635, 4:2308, 1:1287

```

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the R script for finding the 15 longest documents.
- Environment View:** Shows the global environment with variables like `hc` (List of 7), `tdm2` (List of 6), `test11` (List of 2), and `test1df` (111 obs. of 1 variable).
- Packages View:** Shows the installed packages: `grid`, `gridExtra`, `stringr`, `textreuse`, `tidyverse`, `tm`, `tokenizers`, `utf8`, `vegan`, `vinyldite`, `writr`, `wordcloud`, `wordnet`, and `yaml`.
- Console:** Displays the R command `## 15 largest document` and its output.
- Terminal:** Displays the R command `## 15 largest document` and its output.

c. For each document work through the examples given in Lecture 9 to display the dendrogram and the WordCloud.

For the following you will need to write R functions to help you compute the results.

Use the packages `textreuse`, `wordnet`, `zipfR`

```

## 15 largest document
## 50:1068, 47:3013, 44:1022, 42:1607, 36:1043, 34:1465, 29:3109, 25:3516, 22:1873, 20:1009,
## 19:2457, 18:871, 7:3635, 4:2308, 1:1287

```

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the R script for displaying a dendrogram and word cloud.
- Environment View:** Shows the global environment with variables like `hc` (List of 7), `tdm2` (List of 6), `test11` (List of 2), and `test1df` (111 obs. of 1 variable).
- Packages View:** Shows the installed packages: `grid`, `gridExtra`, `stringr`, `textreuse`, `tidyverse`, `tm`, `tokenizers`, `utf8`, `vegan`, `vinyldite`, `writr`, `wordcloud`, `wordnet`, and `yaml`.
- Console:** Displays the R command `## 15 largest document` and its output.
- Terminal:** Displays the R command `## 15 largest document` and its output.

c. Prior to removing the punctuation, find the longest word and longest sentence in each document from the 15 largest documents.

Longest word

The screenshot shows the RStudio interface with several panes:

- Code Editor:** An "Untitled.R" script containing R code to print various word statistics.
- Console:** Displays the output of the R code, including:
 - max_word1 through max_word15
 - max_sentence1 through max_sentence15
 - max_word1len through max_word14len
 - max_word1capitalization through max_word14appreciation
 - max_word1organization through max_word1manufacturing
- Environment:** Shows the global environment with objects like max_word1, max_sentence1, max_word1len, etc.
- Packages:** A list of installed packages including rvest, scales, selectr, snowballC, stringi, stringr, testthat, testreuse, tidyverse, dplyr, tm, tokenizers, usethis, vegan, viridisLite, whisker, withr, wordcloud, wordnet, xm2, and yaml.
- Help:** Provides help for selected packages like rvest and scales.
- Viewer:** Shows the results of the printed objects as text in the console.

Longest sentence

d. Print a table of the length of each sentence in each of the 15 documents.

RStudio

File Edit View Plots Session Build Debug Profile Tools Help

Untitled.R

Source on Tree Run Source

```
535 print(max_sentences0)
536 print(max_sentence1)
537 print(max_sentence2)
538 print(max_sentence3)
539 print(max_sentence4)
540 print(max_sentence5)
541 print(max_sentence6)
542 ## (c) #####
543 # draw a table showing the length of longest sentence
544 length_data <- c(max_sentence_len,max_sentence_len2,max_sentence_len3,max_sentence_len4,max_sentence_len5,max_sentence_len6,max_sentence_len7,max_sentence_len8,max_sentence_len9,max_sentence_len10,max_sentence_len11,max_sentence_len12,max_sentence_len13,max_sentence_len14,max_sentence_len15)
545 ## change length header
546 length_data <- data.frame(len = length_array[1:2])
547 mytable <- cbind/sites ~ c("file 7", "file 50", "file 47", "file 44", "file 42", "file 36", "file 34", "file 29", "file 25", "file 22", "file 20", "file 18", "file 4", "file 1"), length_data[1:2], ## change file name here
548 warning.message <- "Number of rows of result is not a multiple of vector length (arg 2)"
549 rownames(mytable) <- c("No1", "No2", "No3", "No4", "No5", "No6", "No7", "No8", "No9", "No10", "No11", "No12", "No13", "No14", "No15")
550 mytable
551 sites
No1 "file 7" "14"
No2 "file 50" "14"
No3 "file 47" "14"
No4 "file 44" "13"
No5 "file 42" "13"
No6 "file 36" "13"
No7 "file 34" "14"
No8 "file 29" "14"
No9 "file 25" "14"
No10 "file 22" "13"
No11 "file 20" "13"
No12 "file 18" "13"
No13 "file 17" "13"
No14 "file 16" "14"
No15 "file 1" "14"
> |
```

Environment History Connections

Global Environment

Name	Type	Value
length_data	list	[1:15] 1 150 4 4 2 1 2 7 1 2 1 ...
mytable	list	[1:15, 1:2] "file 7" "file 50" "file 47" "file 44" ...
testdf	list	[1:6]
tokensdf	list	[1:11] obs. of 1 variable
values	list	"charoc" "Computer Terminal Systems Inc said it has completed the ...
charoc	list	"Class 'dist' atomic [1:2] 5.98 8.27 8.48 7.12 6.93 ..."
dd	list	"acq" "acq"
ddcdata	list	"chr" [1:209] "ab" "acquire" "acquired" "acquisition" "acqu..."
freq_terms	list	"num" [1:15] 14 13 14 14 15 14 13 13 16 13 ..."
length_array	list	"line" "reuter"
max_sentence	list	"cash" which might be the only reason to sell a part of a ..."

PLOTS

Install Update

Name	Description	Version
project	Finding Files in Project Subdirectories	1.3-2
rstudioapi	Easily Access the RStudio API	0.7
rvest	Easy Harvest (Scrap) Web Pages	0.3.2
scales	Scale Functions for Visualization	0.5.0
selectr	Translate CSS Selectors to XPath Expressions	0.4-1
slam	Sparse Lightweight Arrays and Matrices	0.1-43
SnowballC	Snowball stemmers based on the C libstemmer UTF-8 library	0.5.1
stringr	Character String Processing Facilities	1.1.7
stringi	String Processing Wrappers for Common String Operations	1.5.0
tidyselect	Select From a Set of Strings	0.2.4
tidyverse	Easily Install and Load the 'Tidyverse'	1.2.1
tm	Text Mining Package	0.7-3
tokenizers	Fast, Comprehensive Tokenization of Natural Language Text	0.2.1
ubiquitin	Ubiquitin Processing	1.1.3
vegan	Community Ecology Package	2.5-1
vinistyle	Default Color Maps from 'matplotlib' (Lite Version)	0.3.0
whisker	[(im)atch] for R, logics templating	0.3-2
width	Run Code With 'Temporarily Modified Global State'	2.1.2
wordcloud	Word Clouds	2.5
wordnet	WordNet Interface	0.1-14
xmld	Parse XML	1.2.0
yaml	Methods to Convert R Data to YAML and Back	2.1.19

f. For each sentence of the longest document, remove the punctuation. Display the sentences in the largest document (just show results/screenshot for the longest document).

The screenshot shows the RStudio interface with two main panes. The left pane contains an R script titled 'Untitled.R' with code for sentiment analysis, including functions for removing punctuation and tokenizing sentences. The right pane shows a terminal window with the command 'dist' being run, displaying the results as a distance matrix.

```
543 # draw a table show the length of longest sentence
544 length_array <- c(max_sentence_len,max_sentence_len2,max_sentence_len3,max_sentence_len4,max_sentence_len5,max_sentence_len6,max_sentence_len7)
545 length_data <- data.frame(len = length_array[1:2])
546 mytable <- bindtiffs(c("file 50","file 51","file 42","file 44","file 42","file 36","file 34","file 29","file 25","file 22","file 21"))
547 rowsnum<(mytable) <- c("No1","No2","No3","No4","No5","No7","No8","No9","No10","No11","No12","No13","No14","No15")
548
549 #####
550 ## QC
551 ## remove punctuation
552 fileNopunct <- tm_map(acq, content_transformer(removeNumPunct))
553 docNopunct <- fileNopunct[[7]]
554 tokenize_sentences(as.character(DocINOpunct))
555
556 #####
557 ## QC
558 ## remove part of speech of every word
559 <
560 <-- Untitled.R
```

Console Terminal

```
-/-
> fileNopunct <- tm_map(acq, content_transformer(removeNumPunct))
> docNopunct <- fileNopunct[[7]]
> tokenize_sentences(as.character(DocINOpunct))
[1] "american can express co remained silent on"
[2] "tehman brothers inc but some analysts said the company may be"
[3] "value of its stock"
[4] "the market could give a partially public shearsom may command a"
[5] "experts the rumor also was accompanied by talk the financial"
[6] "american express closed on the new york stock exchange at"
[7] "american express would not comment on the rumors or its"
[8] "analysts said the company at an analysts"
[9] "yesterday of management changes"
[10] "stock is undervalued and does not fully reflect the performance"
[11] "yesterday shearsom said it was elevating its chief"
[12] "program which had been vacant it also created four new"
[13] "analysts speculated a partial spinoff would make most"
[14] "spinoff"
[15] "shearsom would be good since it is a strong profit center for"
[16] "yesterday"
[17] "going to sell shearsom said perin long of lipper analytical"
[18] "profitable securities firm"
[19] "cash which might be the only reason to sell a part of a strong"
[20] "relationship between the two companies"
[21] "rumors suggested selling about pct of it in the market"
[22] "believes american express could have considered a partial"
[23] "shearsom being as profitable as it was would have fetcched a"
[24] "higher price than what it would have in the market place would"
[25] "capitalization said eckfelder"
[26] "plans to expand globally"
[27] "capital you want your stock to reflect realistic valuations to"
[28] "the market place said michael lewis"
[29] "in the future which is heavily into the international"
[30] "divestitures along the way he said"
[31] "Brokerage business by selling part of shearsom its stock might"
[32] "be a good business"
[33] "to break up the value of the other components could command a"
[34] "the total operating earnings of the company he said"
[35] "operating earnings up about mln drs in"
"market rumors it would spinoff all or part of its shearsom"
"considering such a move because it is unhappy with the market"
"american express stock got a lift from the rumor as the"
"shearsom spinoff thereby boosting the total value of american"
"services firm would split its stock and boost its dividend"
"ups on heavy volume"
"stock activity"
"seemingly helped fuel the rumors as did an announcement"
"at the meeting company officials said american express"
"of shearsom according to analysts"
"operating officer jeffery lane to the added position of"
"senior executive chairman it will also divest"
"some contrary to one variation on market rumors of a total"
"some analysts however disagreed that any spinoff of"
"american express contributing about pct of earnings last"
"year is as high unlikely that american express is"
"the question is whether would be a better investment than a very"
"several analysts said american express is not in need of"
"asset"
"including the option of spinning off part of shearsom and one"
"larry eckfelder of prudential bacchus securities said he"
"spinoff in the past"
"big premium in the market place shearsom book value is in"
"some analysts said american express could use capital since"
"they have enormous internal growth plans that takes"
"enhance your ability to make all kinds of endeavors down the"
"theyve been doing well in the market place"
"lewis said lewis that does not preclude acquisitions and"
"lewis said if american express reduced its exposure to the"
"better reflect other assets such as the travel related"
"they could find a true market with a lesser exposure"
"higher profile because they constitute a higher percentage of"
"lewis said shearsom contributed mln aftertax"
```

Environment History Connections

Global Environment

- DocINOpunct List of 2
- filenopunct List of 50
- hc List of 7
- Length_data 2 obs. of 1 variable
- m1 num [1:1, 1:50] 4.4 2.3 2.1 2.1 2.1 ...
- mytable [1:15, 1:2] "file 50" "file 47" "file 44" ...
- tdm2 List of 6
- test11 List of 2
- test1df 111 obs. of 1 variable

Values

- charDoc Class 'dist' atomic [1:21] 9.58 8.27 8.48 7.12 6.93 ...
- cont
- DocData

Files Plot Packages Help Viewer

- Install Update
- Name Description Version
- rprojroot Finding Files in Project Subdirectories 1.3-2
- rstudioapi Safely Access the RStudio API 0.9.2
- rvtest Easily Harvest (Scope) Web Pages 0.3.2
- scales Scale Functions for Visualization 0.5.0
- selectr Translate CSS Selectors to XPath Expressions 0.4-1
- slam Sparse Lightweight Arrays and Matrices 0.1-43
- SnowballC Snowball stemmers based on the libstemmer UTf-8 library 0.5.1
- stringr Character String Processing Facilities 1.1.7
- stringr Simple, Consistent Wrappers for Common String Operations 1.3.0
- testthat Unit Testing for R 2.0.0
- textrouze Detect Text Reuse and Document Similarity 0.1.4
- tidyverse Simple Data Frames 1.4.2
- tidytidy Easily Tidy Data with 'spread()' and 'gather()' Functions 0.8.0
- tidyselect Select From a Set of Strings 0.2.4
- tidyverse Easily Install and Load the 'Tidyverse' 1.2.1
- Text Mining Package 0.7-3
- tm Fast, Consistent Tokenization of Natural Language Text 0.2.1
- tokenizers Unicode Text Processing 1.1.3
- ut8 Community Ecology Package 2.5-1
- vegan Default Color Maps from matplotlib (Lite Version) 0.3.0
- viridisLite (mustache) for R, legible templating 0.3-2
- wirth Run Code With Temporarily Modified Global State 2.1.2
- wordcloud Word Clouds 2.5
- wordnet WordNet Interface 0.1-14
- xml2 Parse XML 1.2.0
- yaml Methods to Convert R Data to YAML and Back 2.1.19

g. For each word print its part of speech using the Wordnet package in the 15 largest documents (just show results/screenshot for the longest document).

The screenshot shows the RStudio interface with the following details:

- Editor:** Contains R code for processing a dataset. The code includes functions like `remove_punctuation`, `tm_map`, `DocTokenizer`, `tokenize_sentences`, and `sentences_words`. It also uses `dplyr` for filtering and `stringr` for extracting parts of speech.
- Console:** Displays the output of the R code, which includes various market rumors and financial statements from American Express. Some examples include:
 - "market rumors it would spinoff all or part of its searshon"
 - "considering such a move because it is unhappy with the market"
 - "american express stock got a lift from the rumor as the market closed on tuesday after the company's value rose american"
 - "the rumor was also accompanied by talk of the firm's financials"
 - "american express closed on the new york stock exchange at \$115.14, up 1.14% from its previous close of \$114.00."
 - "analysts said comments by the company and its analysts"
 - "yesterday of management changes"
 - "the market's reaction did not fully reflect the performance"
 - "yesterday searshon said it was elevating its chief president which had been vacant"
 - "possibly due to the company's reorganizing divisions"
 - "seem contrary to one variation on market rumors of a total"
 - "some analysts however disagreed that any spinoff of the company's travel business was unlikely given earnings last year"
 - "i think it is highly unlikely that american express is"
 - "the questioned what would be a better investment than a very well known analyst said american express is not in need of an asset"
 - "considering the option of spinning off part of searshon and one bank to acknowledge the prudential bacne securities said he"
 - "spinoff in the past"
 - "big premium in the market place"
 - "the market has been very strong"
 - "probably be worth three to 3.5 billion dils in terms of market"
 - "some analysts said american express could use capital since they have been investing heavily in their business"
 - "you want your stock to reflect realistic valuations to"
 - "they have confirmed the fact that they are investing heavily in arena seal lewis"
 - "arena seal lewis"
 - "arenas are being built in the us and abroad"
 - "divestitures along the way he said"
 - "arenas have built in the us and abroad"
- Environment:** Shows the global environment with objects like `chardoc` (atomic), `count` (list), `dd` (list), `document` (list), `frequencies` (list), `length_array` (list), `max_sentence` (list), `max_sentence_1en` (list), and `max_sentence_1ent` (list).
- Files:** Shows open files including `Untitled.Rnw` and `Untitled.R`.
- Plots:** No plots are currently displayed.
- Package:** Shows installed packages: `gridExtra` (0.9.1), `grid` (3.3.2), `gridBase` (0.4.1), `gridExtra` (0.9.1), `grid` (3.3.2), and `gridBase` (0.4.1).
- Help:** Shows help documentation for various R functions.
- Viewer:** Shows the results of the R code execution, including the market rumors and financial statements.

h. Analyze word frequency using functions from package zipfR in the 15 largest documents (just show results/screenshot for the longest document).

The screenshot shows the RStudio interface with the following details:

- Source Tab:** Displays the R script `Untitled.R` containing code for tokenizing sentences and calculating term frequency.
- Console Tab:** Shows the execution of the script, resulting in a large data frame `testFRE` with columns representing words and their frequencies across various documents.
- Environment Tab:** Shows the global environment with objects like `max_word1` through `max_word9`, `stopword`, `pal`, `sentences`, `term.freq`, `testfref`, and `testfrf`.
- Plots Tab:** Shows a scatter plot of word frequency.
- Packages Tab:** Shows available packages: `rpr`, `rstudioapi`, `rvest`, `scales`, `selectr`, `slam`, `SnowballC`, `stringi`, `testthat`, `textreuse`, `tidyverse`, `tidyselect`, `tidyverse`, `tm`, `tokenizers`, `utf8`, `vegan`, `viridisLite`, `whisker`, `withr`, `wordcloud`, `wordNetInterface`, `xmll`, and `yaml`.

b. Write an R function to search through the documents to find a specific word or phrase. Print the document number, line number, and word index in the sentence. Demonstrate with three examples. Use words of 6 characters or more as your test cases. 3 points.

Example1:

The screenshot shows the RStudio interface with a script named 'charFile.R' open in the editor. The code in the script is as follows:

```
> library(tidyverse)
> library(stringr)
> target = "capital"
> file_index = 1
> find = 0
> result_index = 1
> for(i in 1:50){
+   file <- acf(file_index)
+   print(str_sub(file, character(file)))
+   line_index = str_index_in_file(file)
+   for(line in tokenize_sentences(charFile)){
+     word_index = 1
+     for(word in tokenize_words(line)){
+       if(word == target){
+         print(paste(paste("No.", " ", as.character(result_index)), "result"))
+         print(paste(paste("No.", " ", as.character(file_index)), "file"))
+         print(paste(paste("No.", " ", as.character(line_index)), "line"))
+         print(paste(paste("No.", " ", as.character(word_index)), "word"))
+         find = 1
+         result_index = result_index + 1
+         print("-----")
+       }
+       word_index = word_index + 1
+     }
+     line_index = line_index + 1
+   }
+   file_index = file_index + 1
+ }
+ result_index
```

The 'Project Navigator' panel on the right lists several packages and files:

File	Path	Package	Description	Version
charFile.R	charFile.R		Finding Files in Project Subdirectories	1.3.2
curl	curl		Safely Access the RStudio API	0.7
htmltools	htmltools		Easy Harvest (Script) Web Pages	0.3.2
scales	scales		Scale Functions for Visualization	0.5.0
selectr	selectr		Translate CSS Selectors to XPath Expressions	0.4-1
SnowballC	SnowballC		Snowball stemmers based on the libstemmer-UTF-8 library	0.5.1
stringr	stringr		Character Processing Facilities	1.3.7
stringi	stringi		Simple, Convenient Wrappers for Common String Operations	1.3.0
testthat	testthat		Unit Testing for R	2.0.0
textrm	textrm		Detect Tex Reserve and Document Similarity	0.1.4
tidyverse	tidyverse		Simple Data Frames	1.4.2
tidyb	tidyb		Easy Tidy Data with 'spread()' and 'gather()' Functions	0.8.0
tidyselect	tidyselect		Select from a Set of Seings	0.2.4
tidyverse	tidyverse		Easily Install and Load the 'Tidyverse'	1.2.1
tm	tm		Text Mining	0.7-3
tmdbR	tmdbR		Federated Curation of Natural Language Test	0.2.1
utf8	utf8		Unicode Text Processing	1.1.3
viridis	viridis		Community Ecology Package	2.5-1
viridisLite	viridisLite		Default Color Maps from 'matplotlib' (Lite Version)	0.3.0
whisker	whisker		[im]match(es) for R, logicless templating	0.3-2
withr	withr		Run Code 'With' Temporally Modified Global State	2.1.2
wordcloud	wordcloud		Word Clouds	2.5
wordnet	wordnet		WordNet Interface	0.1-14
xmll	xmll		Parse XML	1.2.0
yaml	yaml		Methods to Convert R Data to YAML and Back	2.1.19

Example 2:

The screenshot shows the RStudio interface with the following details:

- Source Editor:** Displays the `Untitled.R` script. The code implements a search function that iterates through a file's content to find a specific word. It uses `grep` to search for words and `strsplit` to handle punctuation.
- Console:** Shows the output of the script execution, including the search results for the word "word".
- Environment View:** Shows the global environment with various objects defined, such as `max_word4`, `max_word5`, `max_word6`, `max_word7`, `max_word8`, `max_word9`, `myStopword`, `pal`, `result_index`, `sentences`, `target`, `term.freq`, `textrcf`, `textre`, `word`, and `word_index`.

Example 3:

```

#> library(textr
#> target = "International"
#> file_index = 1
#> result_index = 1
#> for(i in 1:50){
+   file_index = aq[file_index]
+   print(as.character(file))
+   time_index = 1
+   for(lin in tokenize_sentences(charfile)){
+
+     word_index = 1
+     for(wrd in tokenize_words(lin)){
+       if(wrd == target){
+         print(paste(paste("No.", as.character(result_index)), "result"))
+         print(paste(paste("No.", as.character(file_index)), "file"))
+         print(paste(paste("No.", as.character(time_index)), "time"))
+         print(paste(paste("No.", as.character(word_index)), "word"))
+         result_index = result_index + 1
+         print("-----")
+       }
+       word_index = word_index + 1
+     }
+     time_index = time_index + 1
+   }
+   file_index = file_index + 1
+ }
#> "No. 1 result"
#> "No. 1 file"
#> "No. 63 time"
#> "No. 9 word"
#> "-----"
#> "No. 2 result"
#> "No. 2 file"
#> "No. 2 time"
#> "No. 12 word"
#> "-----"

```

The screenshot shows the RStudio interface with the following components:

- File menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Untitled.R:** A script file containing R code for processing text files and counting specific words.
- Environment tab:** Shows the global environment with variables like max_word4, max_word5, max_word6, max_word7, max_word8, max_word9, result_index, pal, and word_index.
- Console tab:** Shows the execution of the script and its output, including the results of the word count.
- Terminal tab:** Shows the command-line interface where the script is run and its output is displayed.
- Project tab:** Shows a list of packages installed in the project, including tidyverse, scales, selectr, slam, SnowballC, stringr, testthat, tidyverse, tm, tokenizers, utf8, vegem, viridisLite, whisker, purrr, wordcloud, wordnet, xml2, and yaml.

d. Analysis of what this project helped you learn about data science, e.g., the exploration of data which is what you have been doing: 3 points

First, we learn some new function and how to use it. Based on those function, we can get some graph and data what we need for analyses. Then the most important things is know how to analysis those text, we can find the word or sentence or detail about document. We have those data, then we can create some new data for someone who need to know the introduction of those text file. Base on the function inspect(), we can read the file, then we can use some function such as termFreq(), tm_map(), content_transformer(), findFreqTerms(), findAssocs(), subset(), removeSparseTerms(), wordcloud(), nchar(), tokenize_words(), to get some key to solve the problem.