Text Mining

Sentiment Analysis in R

Dr Akhter Raza

Text Mining

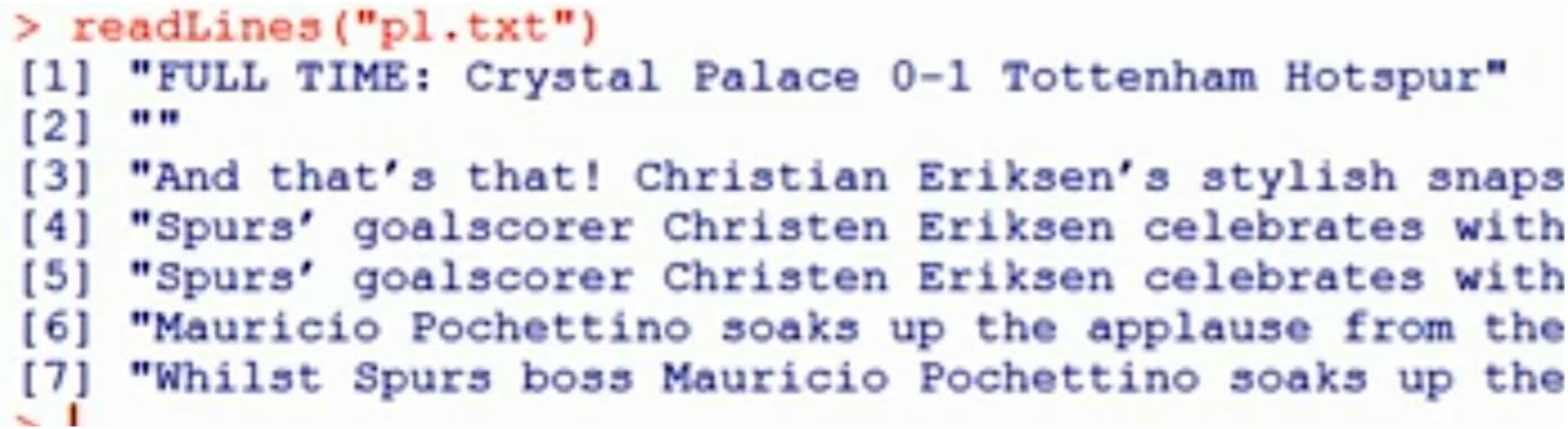
This analysis comprised on 8 segments
First segment is started from here

- create a text file in notepad and save it in the default directory using .txt extension
- I saved this file with name "pl.txt"

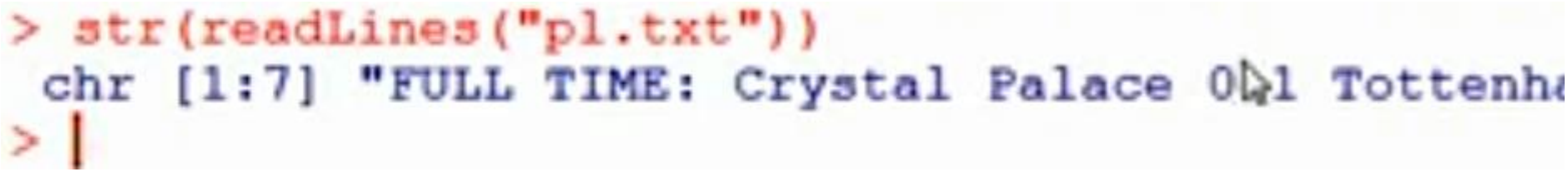
Getting text into workspace

```
# current directory?
    getwd()
#reading a text file into R workspace
#readLines("filename")
   text<-readLines("pl.txt")
   str(readLines ("pl.txt"))
```

readline() will read line by line



```
> readLines("pl.txt")
[1] "FULL TIME: Crystal Palace 0-1 Tottenham Hotspur"
[2] ""
[3] "And that's that! Christian Eriksen's stylish snaps
[4] "Spurs' goalscorer Christen Eriksen celebrates with
[5] "Spurs' goalscorer Christen Eriksen celebrates with
[6] "Mauricio Pochettino soaks up the applause from the
[7] "Whilst Spurs boss Mauricio Pochettino soaks up the
>
```



```
> str(readLines("pl.txt"))
 chr [1:7] "FULL TIME: Crystal Palace 0-1 Tottenh
>
```

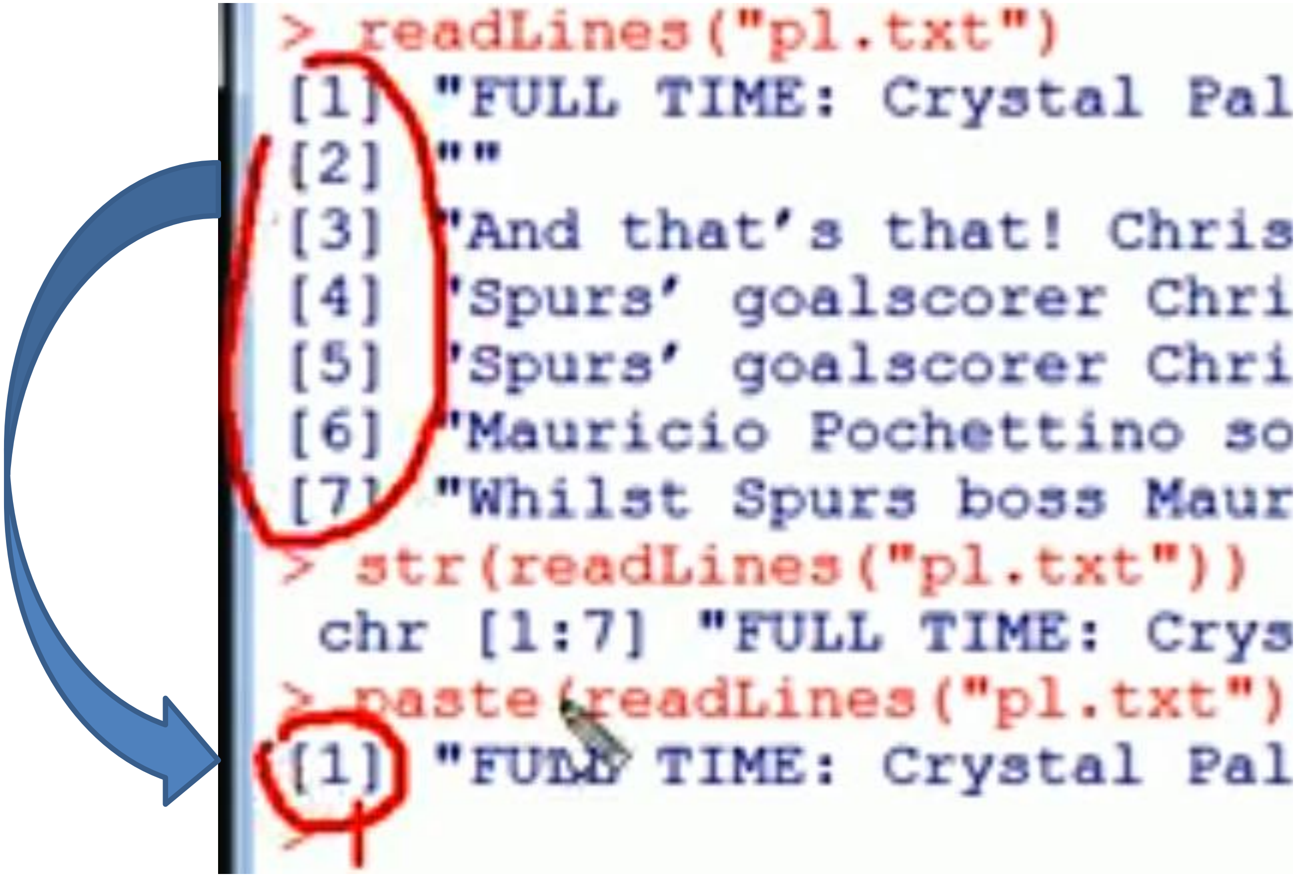
Collapse text in one line

We don’t want to keep text in separate lines. collapse all lines into one line using paste function with collapse option

```
paste(readLines("pl.txt"),collapse = " ")
```

Collapse text in one line

paste(readLines("pl.txt"),collapse = " ")



```
> readLines("pl.txt")
[1] "FULL TIME: Crystal Pal
[2] ""
[3] "And that's that! Chris
[4] "Spurs' goalscorer Chri
[5] "Spurs' goalscorer Chri
[6] "Mauricio Pochettino so
[7] "Whilst Spurs boss Maur
> str(readLines("pl.txt"))
 chr [1:7] "FULL TIME: Crys
> paste(readLines("pl.txt")
[1] "FULL TIME: Crystal Pal
>
```

Collapse another example

create a vector with 6 elements and then try to collapse it using paste

```
helo<-c("name", "of","my","country","is", "pakistan")
```

```
[1] "name" "of" "my" "country“ "is“  "pakistan"
```

Collapse text in one line

```
paste(helo, collapse = " ")
[1] "name  of  my  country  is  pakistan"


# collapse and separate with commas
paste(helo,collapse = ", ")
[1] "name,  of,  my,  country,  is,  pakistan"
```

Corpus of whole text

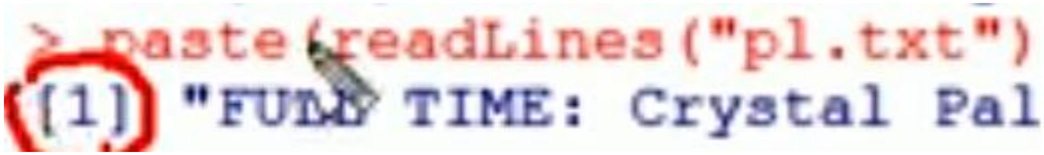
The purpose of collapse is to form a Corpus of whole text and cleaning will be done collectively in the whole document then we will separate the words after cleaning

Collapse text in one line

2nd segment

first collapse lines into one

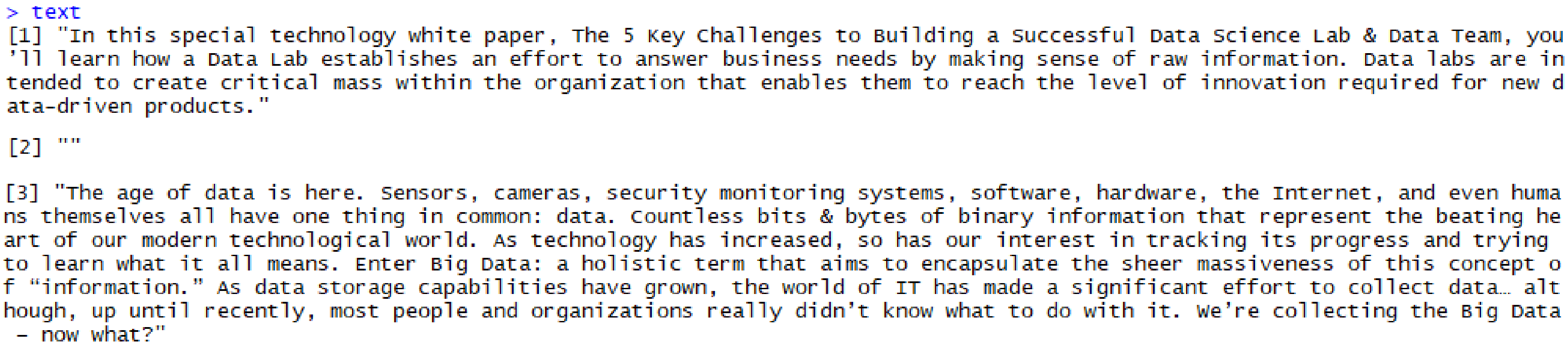
text <-paste(readLines("pl.txt"),collapse = " ")



```
> paste(readLines("pl.txt")
[1] "FULL TIME: Crystal Pal
```

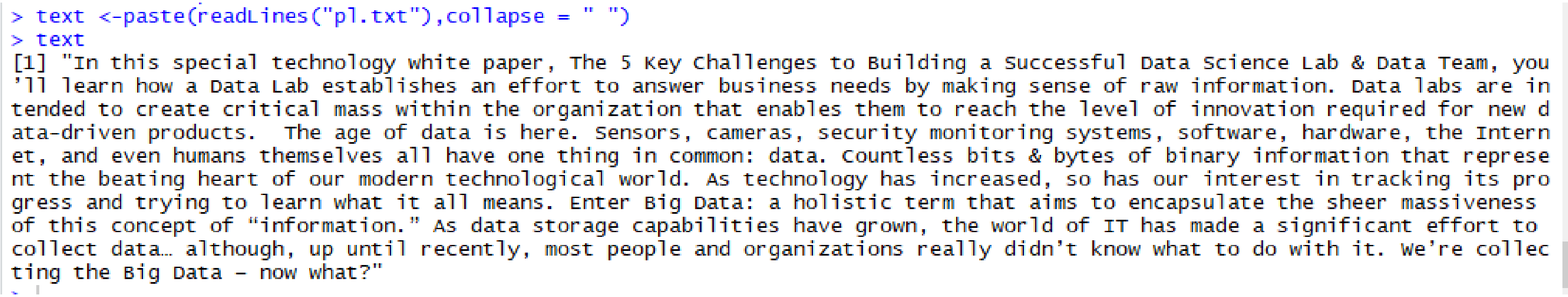
text before and after collapse

Text before collapse comprised on three elements in a list



```
> text
[1] "In this special technology white paper, The 5 Key Challenges to Building a Successful Data Science Lab & Data Team, you
'll learn how a Data Lab establishes an effort to answer business needs by making sense of raw information. Data labs are in
tended to create critical mass within the organization that enables them to reach the level of innovation required for new d
ata-driven products."

[2] ""

[3] "The age of data is here. Sensors, cameras, security monitoring systems, software, hardware, the Internet, and even huma
ns themselves all have one thing in common: data. Countless bits & bytes of binary information that represent the beating he
art of our modern technological world. As technology has increased, so has our interest in tracking its progress and trying
 to learn what it all means. Enter Big Data: a holistic term that aims to encapsulate the sheer massiveness of this concept o
f “information.” As data storage capabilities have grown, the world of IT has made a significant effort to collect data… alt
hough, up until recently, most people and organizations really didn’t know what to do with it. We’re collecting the Big Data
 – now what?"
```

Single line Text after collapse



```
> text <-paste(readLines("pl.txt"),collapse = " ")
> text
[1] "In this special technology white paper, The 5 Key Challenges to Building a Successful Data Science Lab & Data Team, you
'll learn how a Data Lab establishes an effort to answer business needs by making sense of raw information. Data labs are in
tended to create critical mass within the organization that enables them to reach the level of innovation required for new d
ata-driven products.  The age of data is here. Sensors, cameras, security monitoring systems, software, hardware, the Intern
et, and even humans themselves all have one thing in common: data. Countless bits & bytes of binary information that represe
nt the beating heart of our modern technological world. As technology has increased, so has our interest in tracking its pro
gress and trying to learn what it all means. Enter Big Data: a holistic term that aims to encapsulate the sheer massiveness
of this concept of “information.” As data storage capabilities have grown, the world of IT has made a significant effort to
collect data… although, up until recently, most people and organizations really didn’t know what to do with it. We’re collec
ting the Big Data – now what?"
```

Remove punctuations by gsub()

\\W is for replacing punctuations with space

```
text2<-gsub(pattern = "\\W",replace=" ",text)
```

Remove digits by using “\\d” in gsub()

Replace digits using \\d in gsub() with spaces

```
text3  <- gsub(pattern= "\\d",replace=" ",text2)
```

lowercase

convert into lower cases

```
text4 <- tolower(text3)
```

Install tm package

installing text mining package tm

```
install.packages("tm")
```

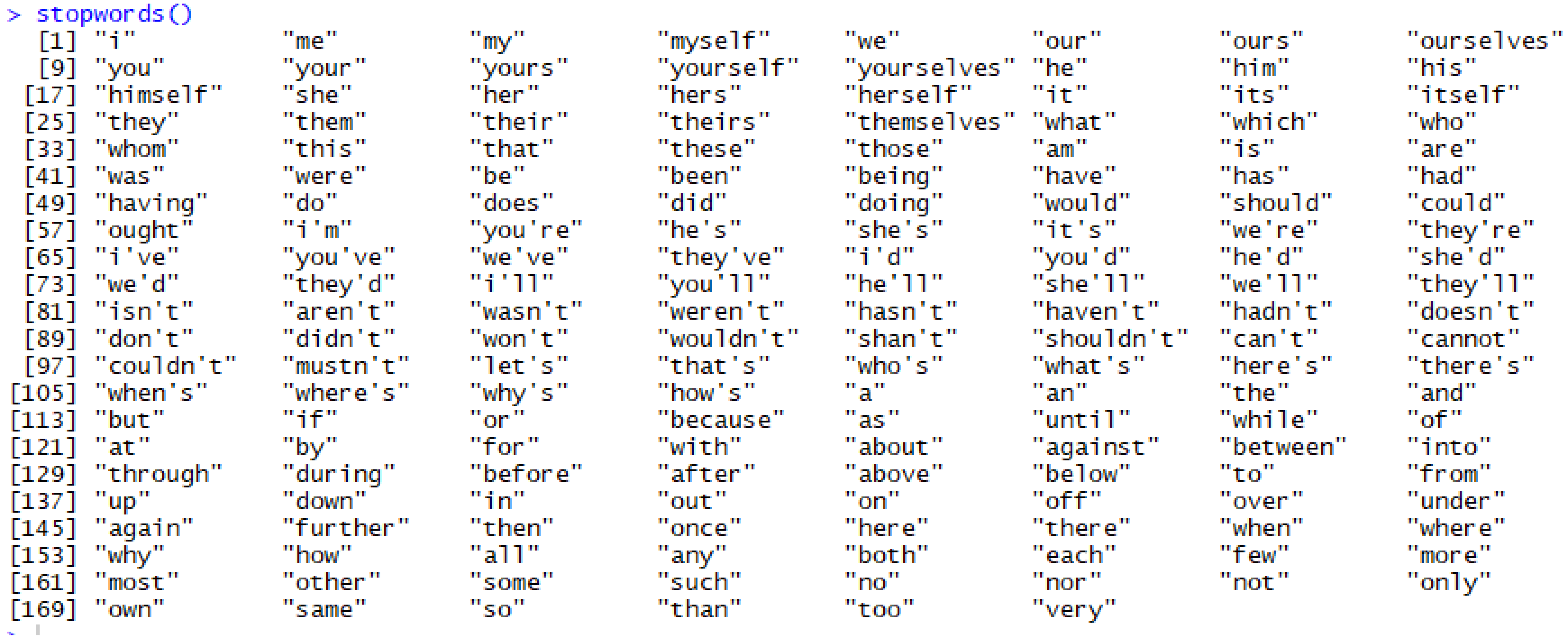
load the package tm

```
library("tm")
```

List of stopwords

check the list of all stop words

stopwords()



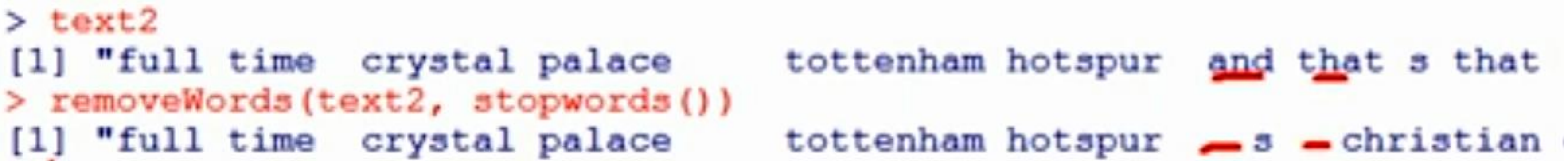
```
> stopwords()
  [1] "i"          "me"         "my"         "myself"     "we"         "our"        "ours"       "ourselves" 
  [9] "you"        "your"       "yours"      "yourself"   "yourselves" "he"         "him"        "his"       
 [17] "himself"    "she"        "her"        "hers"       "herself"    "it"         "its"        "itself"    
 [25] "they"       "them"       "their"      "theirs"     "themselves" "what"       "which"      "who"       
 [33] "whom"       "this"       "that"       "these"      "those"      "am"         "is"         "are"       
 [41] "was"        "were"       "be"         "been"       "being"      "have"       "has"        "had"       
 [49] "having"     "do"         "does"       "did"        "doing"      "would"      "should"     "could"     
 [57] "ought"      "i'm"        "you're"     "he's"       "she's"      "it's"       "we're"      "they're"   
 [65] "i've"       "you've"     "we've"      "they've"    "i'd"        "you'd"      "he'd"       "she'd"     
 [73] "we'd"       "they'd"     "i'll"       "you'll"     "he'll"      "she'll"     "we'll"      "they'll"   
 [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"     "haven't"    "hadn't"     "doesn't"   
 [89] "don't"      "didn't"     "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"    
 [97] "couldn't"   "mustn't"    "let's"      "that's"     "who's"      "what's"     "here's"     "there's"   
[105] "when's"     "where's"    "why's"      "how's"      "a"          "an"         "the"        "and"       
[113] "but"        "if"         "or"         "because"    "as"         "until"      "while"      "of"        
[121] "at"         "by"         "for"        "with"       "about"      "against"    "between"    "into"      
[129] "through"    "during"     "before"     "after"      "above"      "below"      "to"         "from"      
[137] "up"         "down"       "in"         "out"        "on"         "off"        "over"       "under"     
[145] "again"      "further"    "then"       "once"       "here"       "there"      "when"       "where"     
[153] "why"        "how"        "all"        "any"        "both"       "each"       "few"        "more"      
[161] "most"       "other"      "some"       "such"       "no"         "nor"        "not"        "only"      
[169] "own"        "same"       "so"         "than"       "too"        "very"      
```

List of stopwords

Removing helping words like and, or, is etc.
these words are called stopwords

removeWords(text4, c("and","or"))

removeWords(text4, stopwords())



```
> text2
[1] "full time  crystal palace     tottenham hotspur  and that s that
> removeWords(text2, stopwords())
[1] "full time  crystal palace     tottenham hotspur   s  christian
```

Removing specific words

\\b mean start with letter given after that

\\bs all words starts with s will be removed

do not use the following command

```
gsub(pattern =\\bs )
```

delete words like Success, source, side, suggest

Removing words of any size

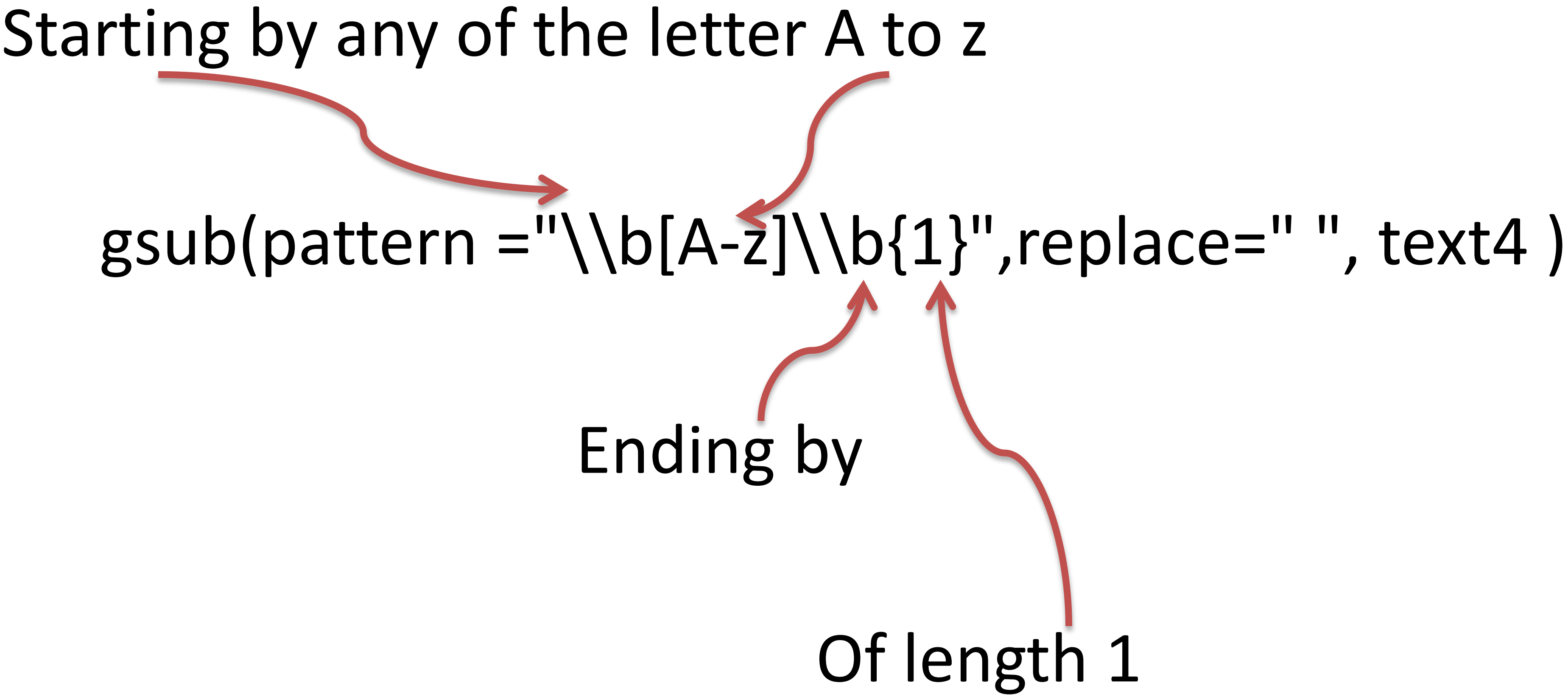
\\b[A-z] remove all words starting A to z

If we use \\b again shows end with

{1} with of size 1

```
gsub(pattern="\\b[A-z]\\b{1}",replace="",text4 )
```

Removing words of any size



Starting by any of the letter A to z

```
gsub(pattern ="\\b[A-z]\\b{1}",replace=" ", text4 )
```

Ending by

Of length 1

Removing whitespaces

```
#remove all extra white spaces from text


      stripWhitespace(text4)
```

Using *stringr* and *wordcloud* package

```
#Text Mining Part 3
        install.packages("stringr")
        install.packages("wordcloud")
        library("stringr")
        library("wordcloud")
```

Splitting string into list of words

splitting string into individual words which are separated using single space

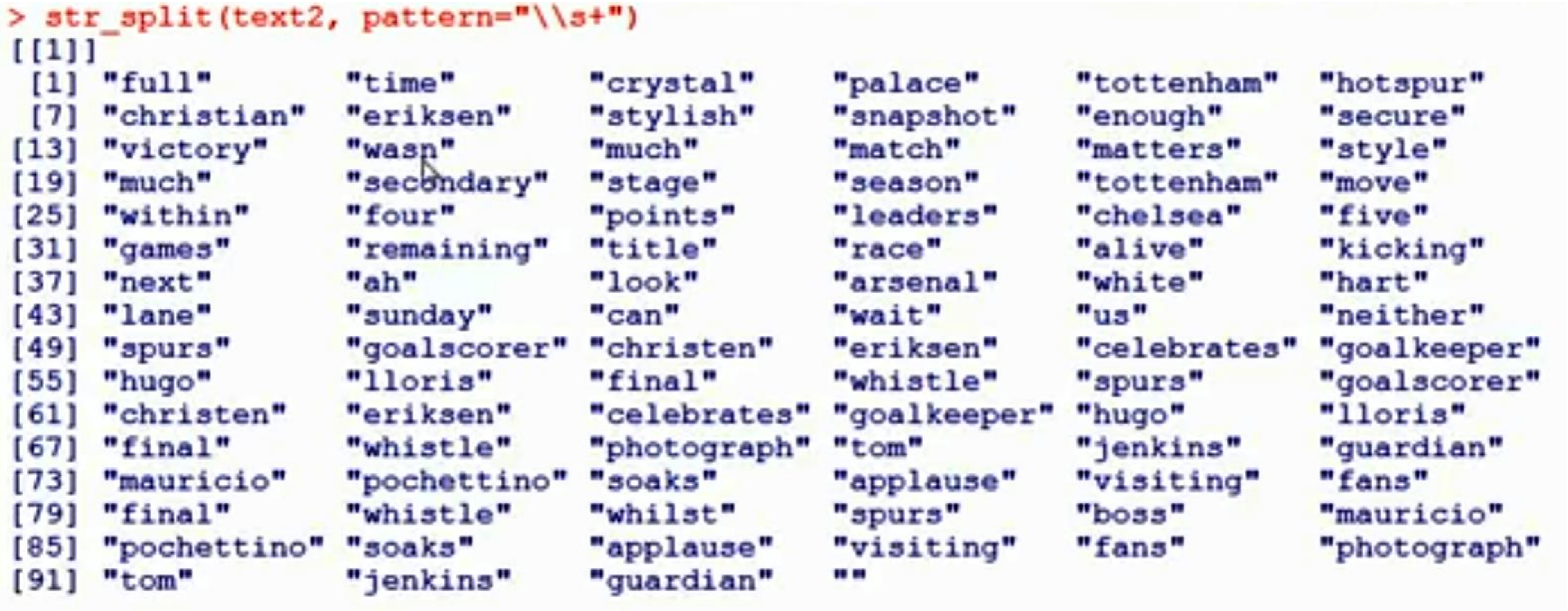
```
str_split(text, pattern = " ")
```

Separated by any number of spaces

\\s+   s means space + means any number of spaces

```
wordBag <- str_split(text, pattern = "\\s+")
```

wordBag created



```
> str_split(text2, pattern="\\s+")
[[1]]
 [1] "full"       "time"       "crystal"    "palace"     "tottenham"  "hotspur"   
 [7] "christian"  "eriksen"    "stylish"    "snapshot"   "enough"     "secure"    
[13] "victory"    "wasn"       "much"       "match"      "matters"    "style"     
[19] "much"       "secondary"  "stage"      "season"     "tottenham"  "move"      
[25] "within"     "four"       "points"     "leaders"    "chelsea"    "five"      
[31] "games"      "remaining"  "title"      "race"       "alive"      "kicking"   
[37] "next"       "ah"         "look"       "arsenal"    "white"      "hart"      
[43] "lane"       "sunday"     "can"        "wait"       "us"         "neither"   
[49] "spurs"      "goalscorer" "christen"   "eriksen"    "celebrates" "goalkeeper"
[55] "hugo"       "lloris"     "final"      "whistle"    "spurs"      "goalscorer"
[61] "christen"   "eriksen"    "celebrates" "goalkeeper" "hugo"       "lloris"    
[67] "final"      "whistle"    "photograph" "tom"        "jenkins"    "guardian"  
[73] "mauricio"   "pochettino" "soaks"      "applause"   "visiting"   "fans"      
[79] "final"      "whistle"    "whilst"     "spurs"      "boss"       "mauricio"  
[85] "pochettino" "soaks"      "applause"   "visiting"   "fans"       "photograph"
[91] "tom"        "jenkins"    "guardian"   ""          
```

Unlist wordbag

What is the class of wordBag?

```
class(wordBag)
```

is a list now we transform it into char

```
wordBag <- unlist(wordBag)
class(wordBag)
```

Now it is character

Web links for +ve and –ve words

Link for Positive words

http://ptrckprry.com/course/ssd/data/positive-words.txt

Link for Negative words

http://ptrckprry.com/course/ssd/data/negative-words.txt

Web links for +ve and –ve words

Copy all the positive and negative terms from the above two links and place them in two separate text files in notepad and save these files in your default current directory

Web links for +ve and –ve words

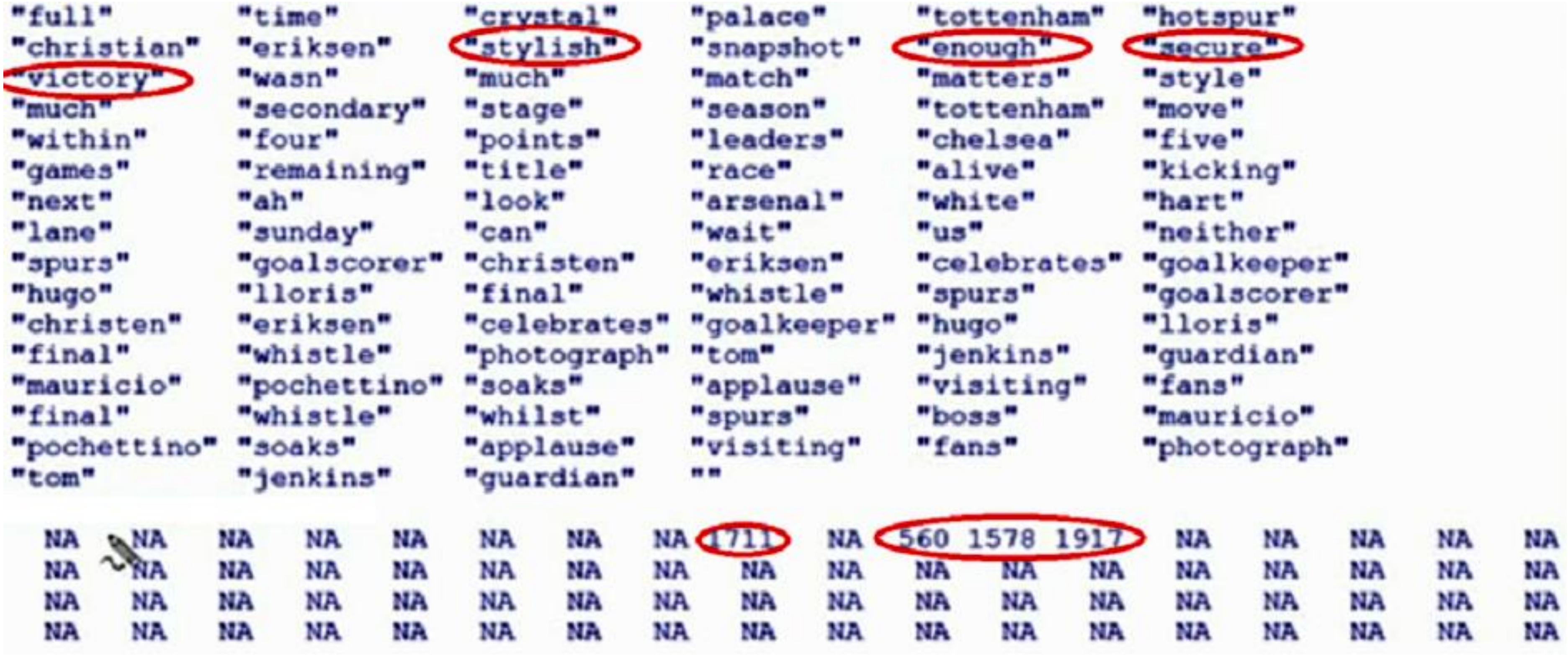
Now we have following three objects

wordBag	→ actual document
negwords	→ standarized -ve words
poswords	→ standarized +ve words

Finding positive words

```
# now we have to find which of the words
are positive and which are negative

    match(wordBag, poswords)
```

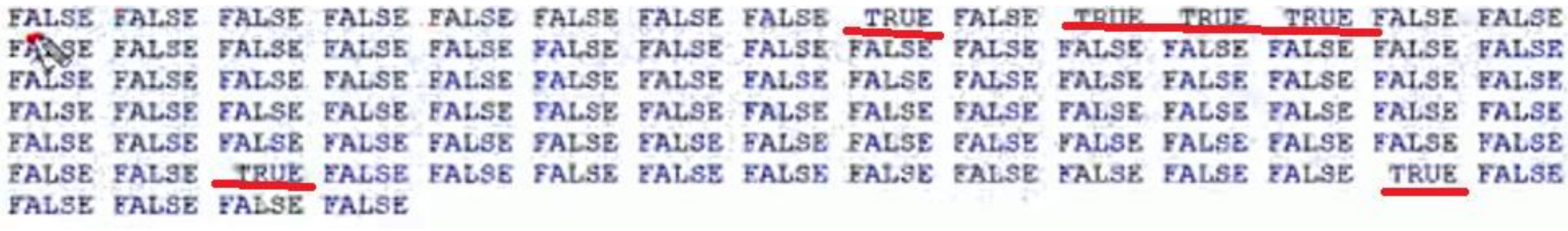
Finding positive words



```
"full"       "time"       "crystal"    "palace"     "tottenham"  "hotspur"
"christian"  "eriksen"    "stylish"    "snapshot"   "enough"     "secure"
"victory"    "wasn"       "much"       "match"      "matters"    "style"
"much"       "secondary"  "stage"      "season"     "tottenham"  "move"
"within"     "four"       "points"     "leaders"    "chelsea"    "five"
"games"      "remaining"  "title"      "race"       "alive"      "kicking"
"next"       "ah"         "look"       "arsenal"    "white"      "hart"
"lane"       "sunday"     "can"        "wait"       "us"         "neither"
"spurs"      "goalscorer" "christen"   "eriksen"    "celebrates" "goalkeeper"
"hugo"       "lloris"     "final"      "whistle"    "spurs"      "goalscorer"
"christen"   "eriksen"    "celebrates" "goalkeeper" "hugo"       "lloris"
"final"      "whistle"    "photograph" "tom"        "jenkins"    "guardian"
"mauricio"   "pochettino" "soaks"      "applause"   "visiting"   "fans"
"final"      "whistle"    "whilst"     "spurs"      "boss"       "mauricio"
"pochettino" "soaks"      "applause"   "visiting"   "fans"       "photograph"
"tom"        "jenkins"    "guardian"   ""

  NA   NA   NA   NA   NA   NA   NA   NA 1711   NA  560 1578 1917   NA   NA   NA   NA   NA
  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

Finding positive words

!is.na(match(wordBag, poswords))



```
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE FALSE FALSE FALSE
```

total positive words

```
sum(!is.na(match(wordBag, poswords)))
```

Answer is 6

total negative words

```
sum(!is.na(match(wordBag, negwords)))
```

Answer is 0

Sentiment score

```
score <-
sum(!is.na(match(wordBag, poswords)))
-
sum(!is.na(match(wordBag, negwords)))
```

Answer is 6

Sentiment score

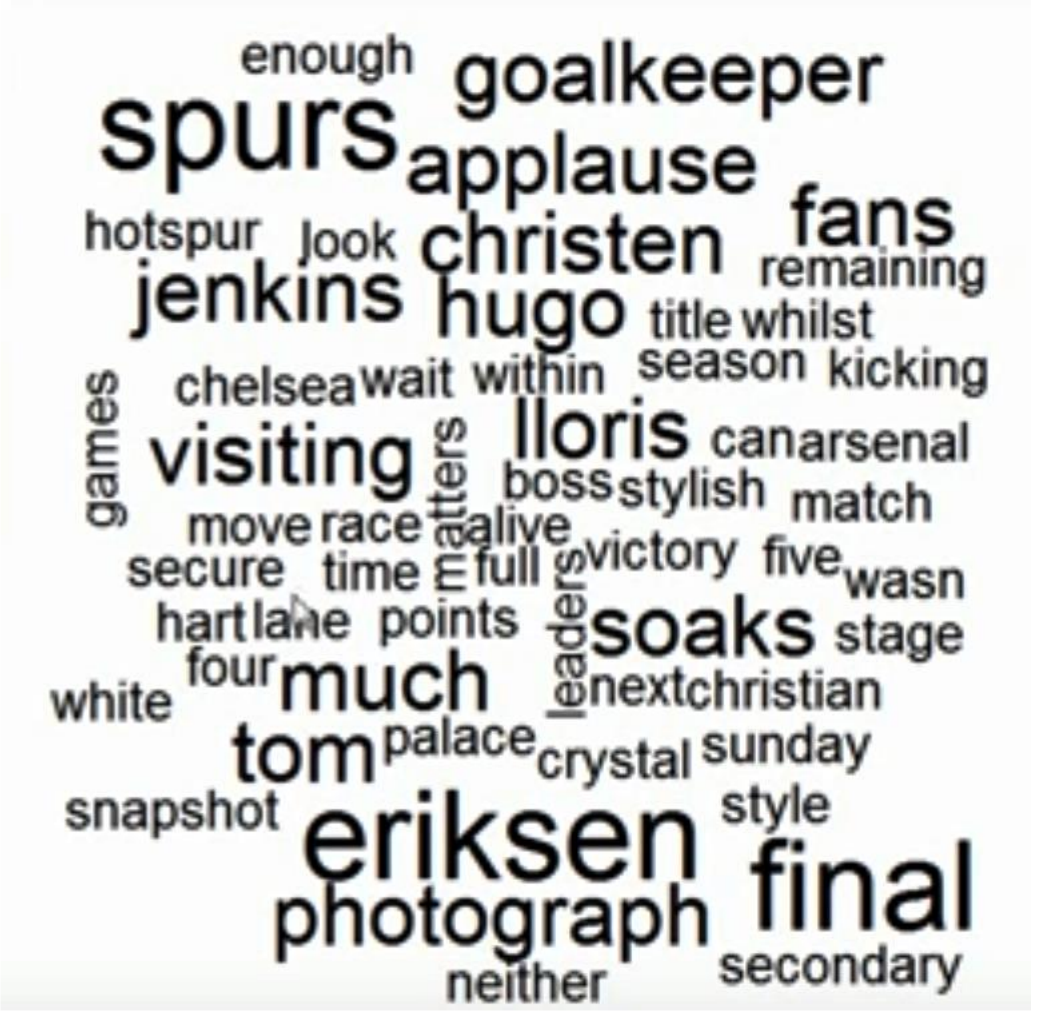
If we have thousands of documents in the Corpus then sentiment score for each document is a vector.

We can find mean, sd of score

We can also construct the hist(score) to show the distribution of sentiment analysis

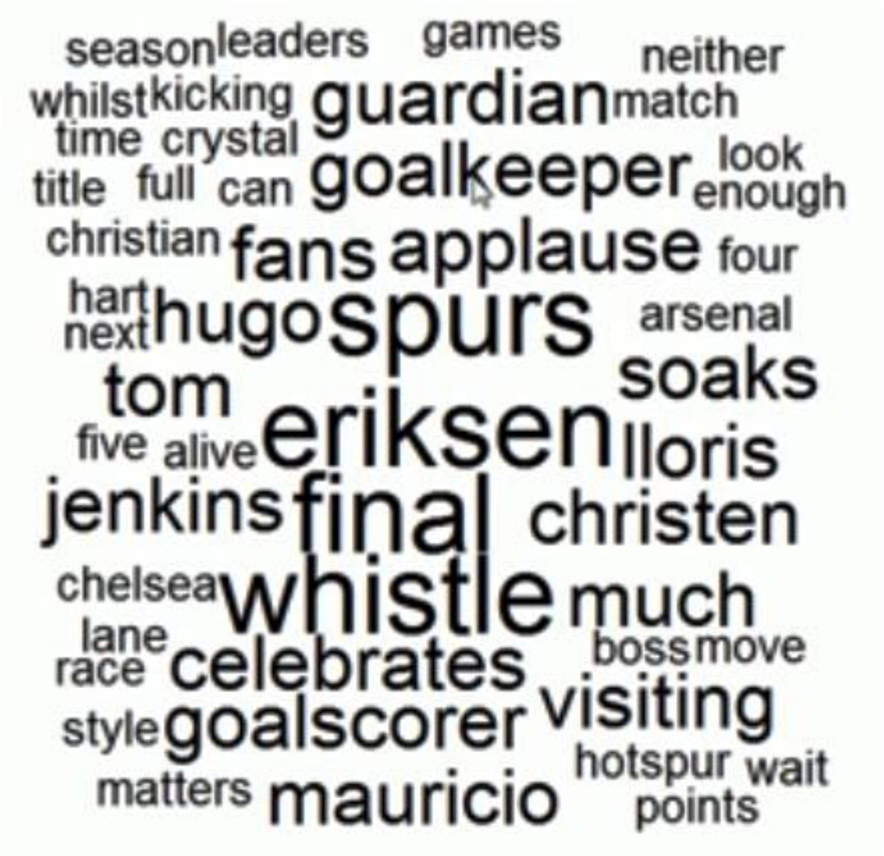
wordCloud with min freq

```
wordcloud(wordBag, min.freq = 4)
```



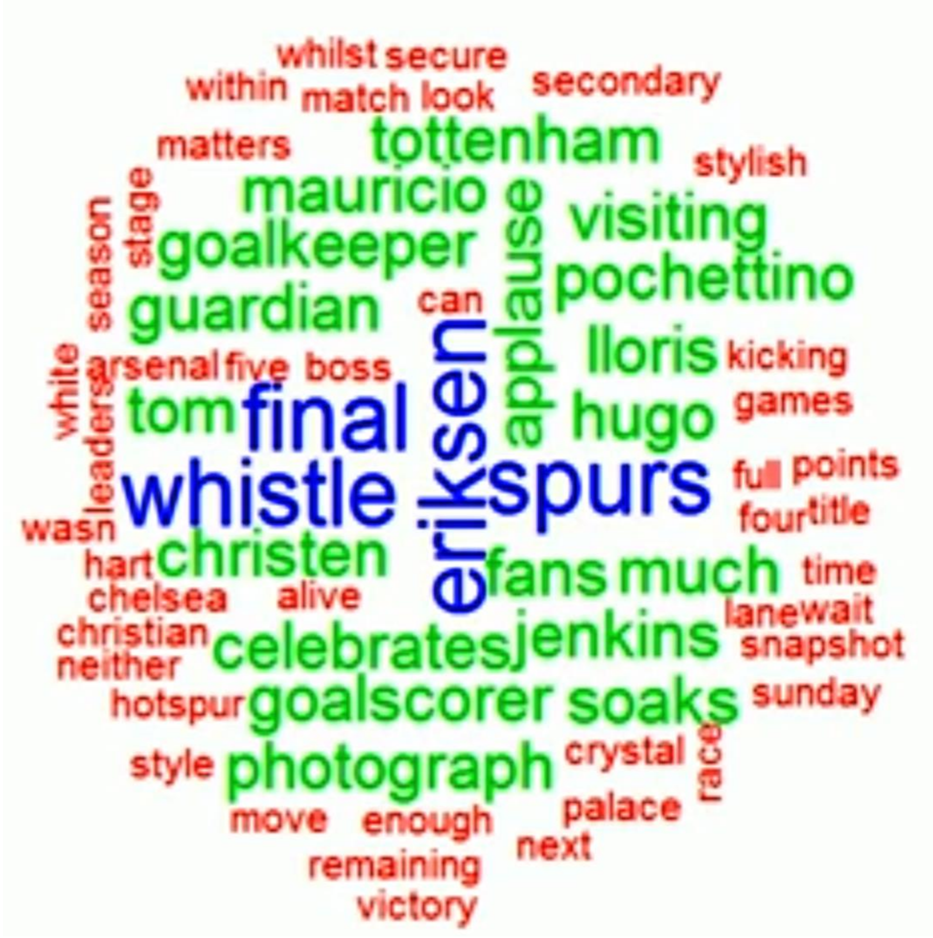
wordCloud with non random order

```
wordcloud(wordBag, min.freq = 4, random.order= FALSE)
```



wordCloud with rainbow color

```
wordcloud(wordBag, min.freq = 4, random.order= FALSE,
scale=c(3, 0.5),color=rainbow(3))
```



Working on multiple documents

5th Segment

Download a courpus of documents into R

Three packages are needed

tm

wordcloud

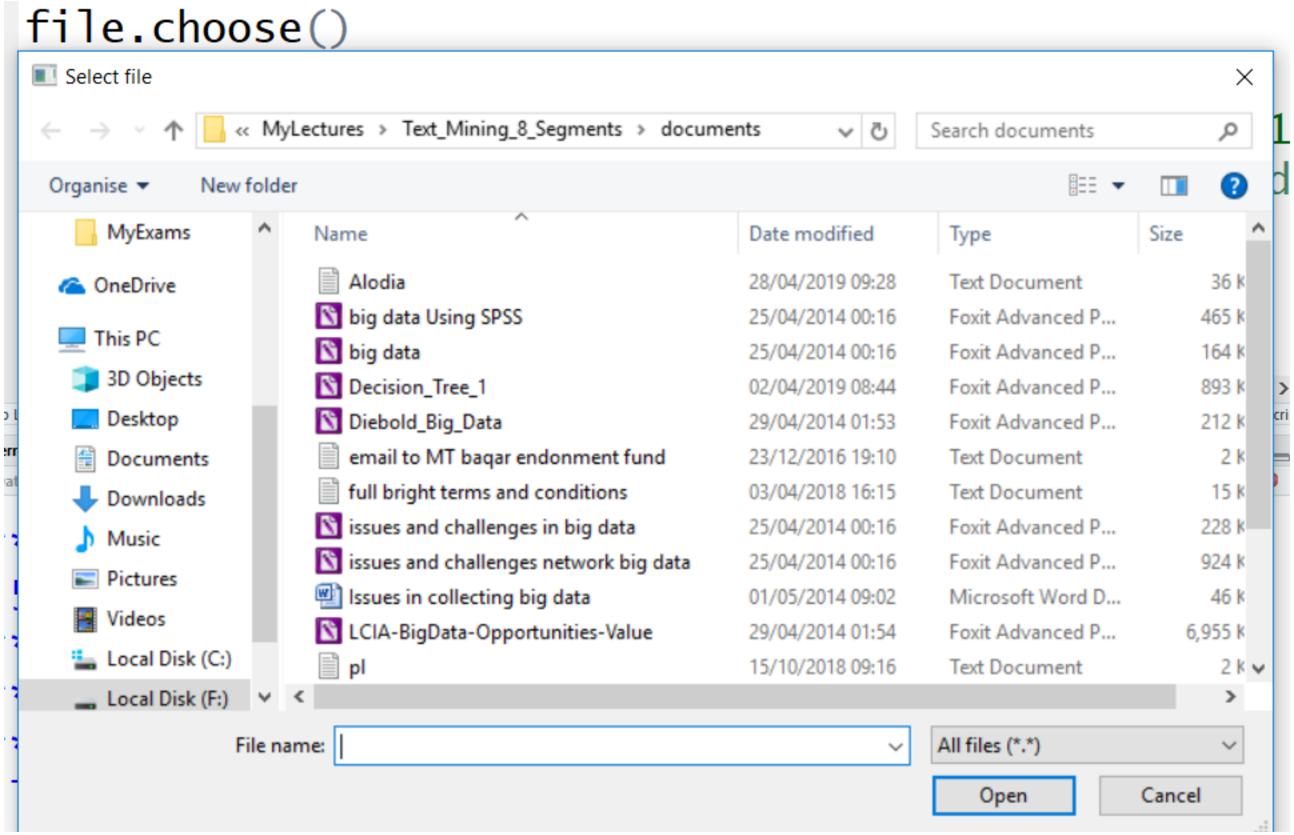
stringr

Working on multiple documents

Save some text document into a folder

file.choose() function will help us to to locate folder in which we have these files

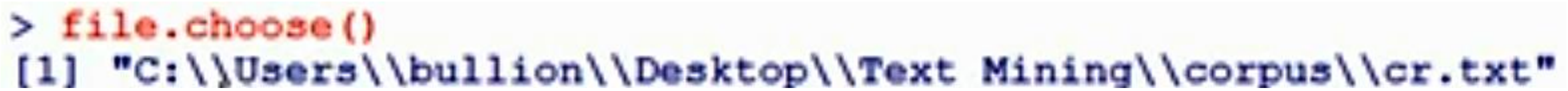
file.choose()



```
file.choose()
```

Name	Date modified	Type	Size
Alodia	28/04/2019 09:28	Text Document	36 K
big data Using SPSS	25/04/2014 00:16	Foxit Advanced P...	465 K
big data	25/04/2014 00:16	Foxit Advanced P...	164 K
Decision_Tree_1	02/04/2019 08:44	Foxit Advanced P...	893 K
Diebold_Big_Data	29/04/2014 01:53	Foxit Advanced P...	212 K
email to MT baqar endonment fund	23/12/2016 19:10	Text Document	2 K
full bright terms and conditions	03/04/2018 16:15	Text Document	15 K
issues and challenges in big data	25/04/2014 00:16	Foxit Advanced P...	228 K
issues and challenges network big data	25/04/2014 00:16	Foxit Advanced P...	924 K
Issues in collecting big data	01/05/2014 09:02	Microsoft Word D...	46 K
LCIA-BigData-Opportunities-Value	29/04/2014 01:54	Foxit Advanced P...	6,955 K
pl	15/10/2018 09:16	Text Document	2 K

File name:



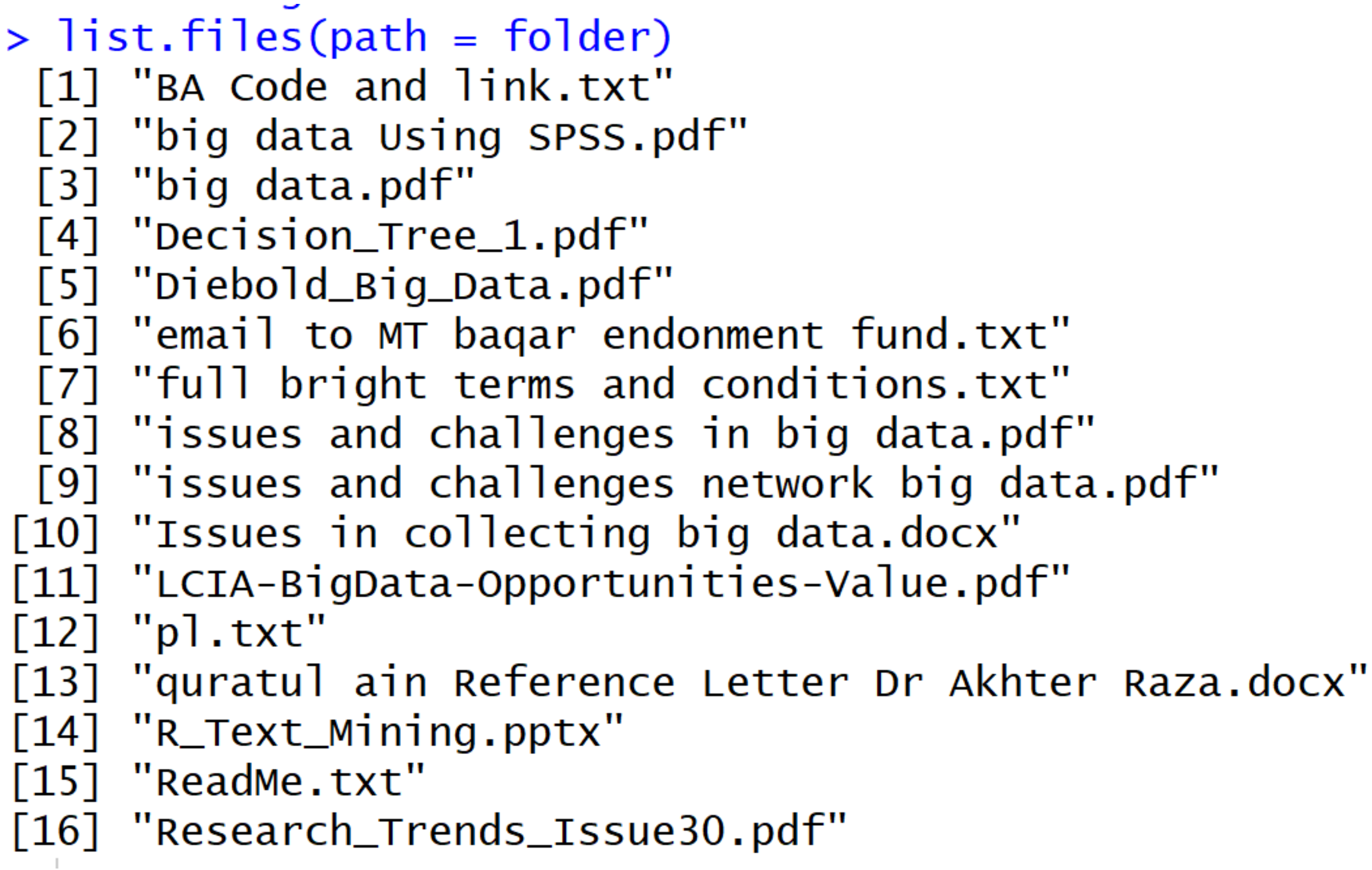
```
> file.choose()
[1] "C:\\Users\\bullion\\Desktop\\Text Mining\\corpus\\cr.txt"
```

Save the file path

```
folder <-"F:\\Users\\bullion\\Desktop\\Text Mining\\corpus"
```

List of files in folder

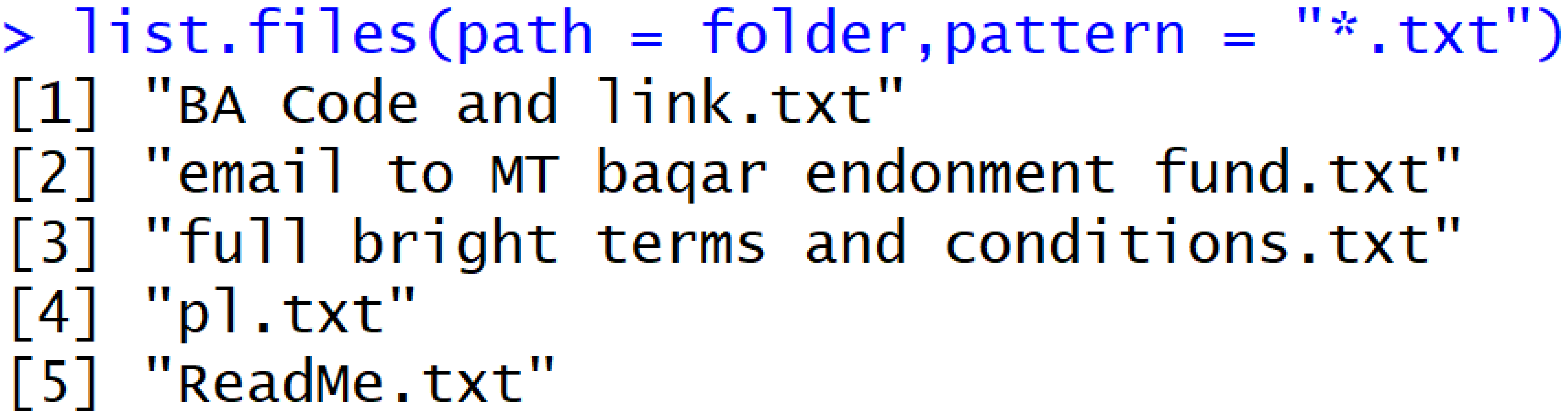
list.files(path = folder)



```
> list.files(path = folder)
 [1] "BA Code and link.txt"
 [2] "big data Using SPSS.pdf"
 [3] "big data.pdf"
 [4] "Decision_Tree_1.pdf"
 [5] "Diebold_Big_Data.pdf"
 [6] "email to MT baqar endonment fund.txt"
 [7] "full bright terms and conditions.txt"
 [8] "issues and challenges in big data.pdf"
 [9] "issues and challenges network big data.pdf"
[10] "Issues in collecting big data.docx"
[11] "LCIA-BigData-Opportunities-Value.pdf"
[12] "pl.txt"
[13] "quratul ain Reference Letter Dr Akhter Raza.docx"
[14] "R_Text_Mining.pptx"
[15] "ReadMe.txt"
[16] "Research_Trends_Issue30.pdf"
```

Only txt files

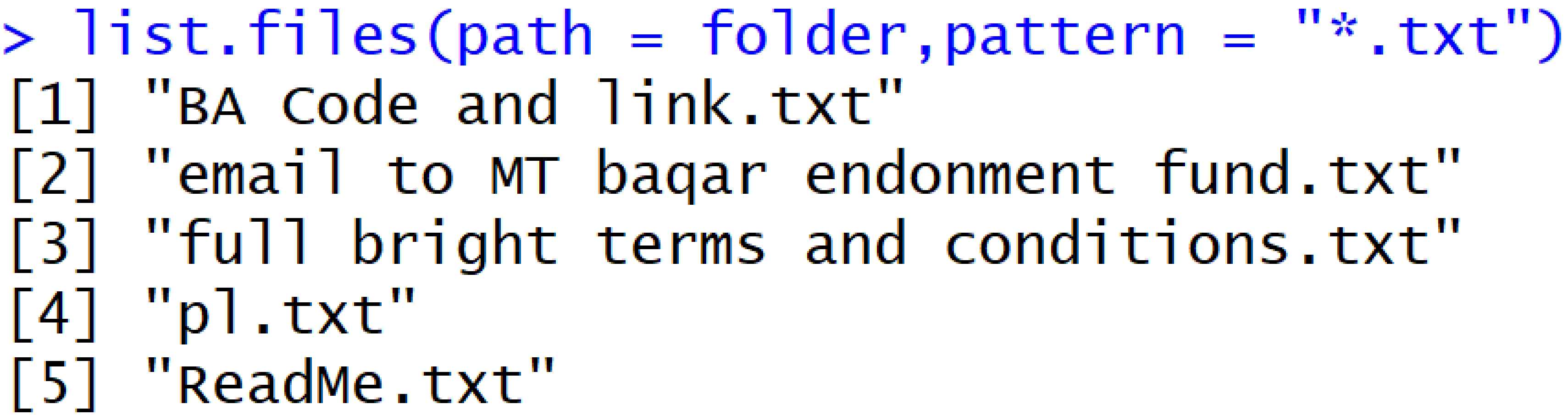
list.files(path = folder,pattern = "*.txt")



```
> list.files(path = folder,pattern = "*.txt")
[1] "BA Code and link.txt"
[2] "email to MT baqar endonment fund.txt"
[3] "full bright terms and conditions.txt"
[4] "pl.txt"
[5] "ReadMe.txt"
```

File list stored in a sepearte variable

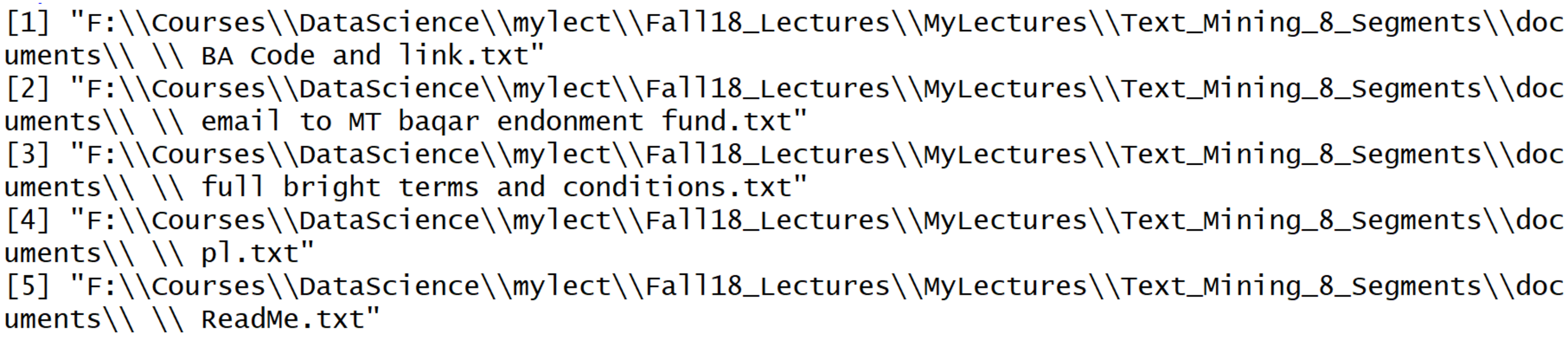
```
filelist<- list.files(path = folder,pattern = "*.txt")
filelist
```



```
> list.files(path = folder,pattern = "*.txt")
[1] "BA Code and link.txt"
[2] "email to MT baqar endonment fund.txt"
[3] "full bright terms and conditions.txt"
[4] "pl.txt"
[5] "ReadMe.txt"
```

Paste folder name with filename

paste(folder, "\\",filelist)

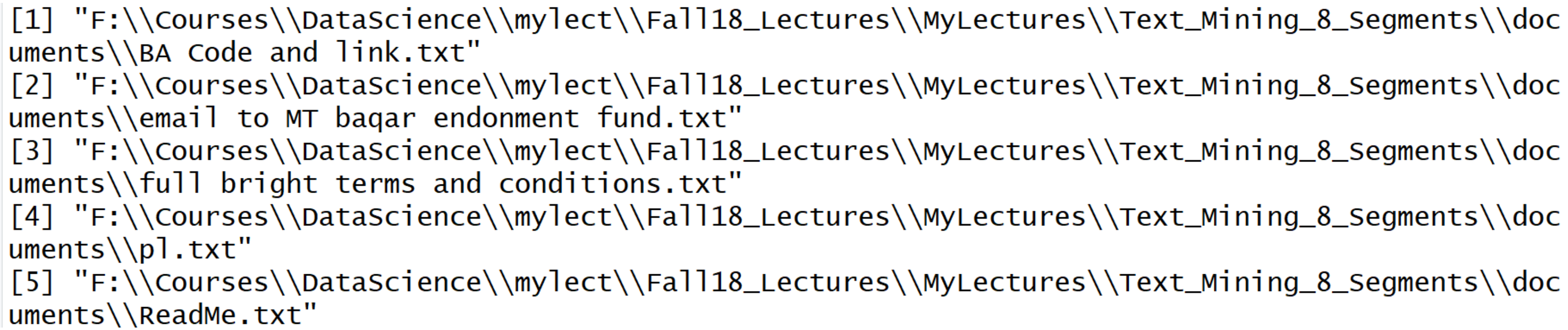


```
[1] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ \\ BA Code and link.txt"
[2] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ \\ email to MT baqar endonment fund.txt"
[3] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ \\ full bright terms and conditions.txt"
[4] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ \\ pl.txt"
[5] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ \\ ReadMe.txt"
```

Paste folder name with filename

Removing spaces from filenames

```
filelist<-paste(folder, filelist, sep =“”)
```

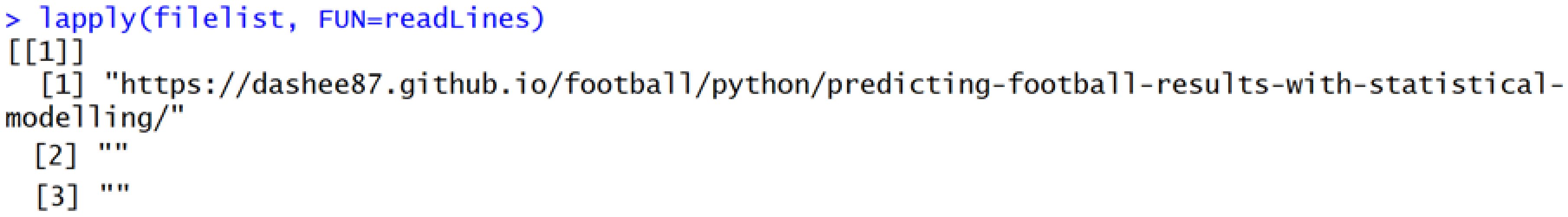


```
[1] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\BA Code and link.txt"
[2] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\email to MT baqar endonment fund.txt"
[3] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\full bright terms and conditions.txt"
[4] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\pl.txt"
[5] "F:\\Courses\\DataScience\\mylect\\Fall18_Lectures\\MyLectures\\Text_Mining_8_Segments\\doc
uments\\ReadMe.txt"
```

Reading lines from all of these docs

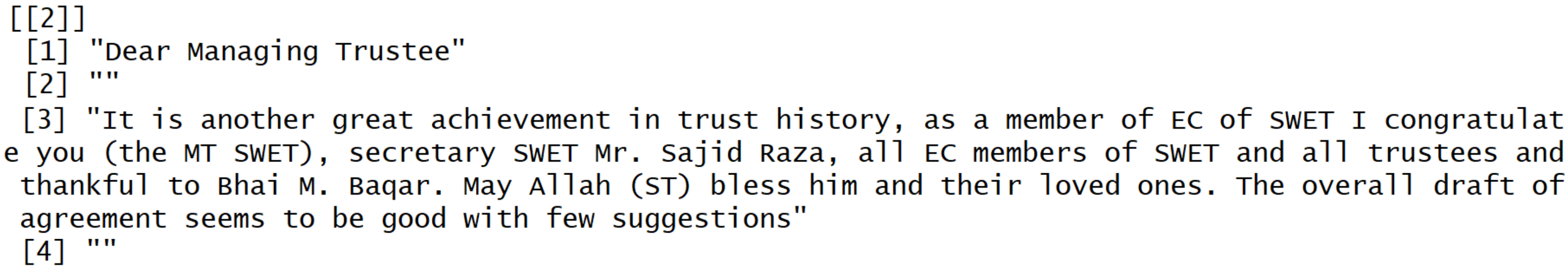
lapply(filelist, FUN=readLines)

First line from document 1



```
> lapply(filelist, FUN=readLines)
[[1]]
  [1] "https://dashee87.github.io/football/python/predicting-football-results-with-statistical-
modelling/"
  [2] ""
  [3] ""
```

First line from document 1

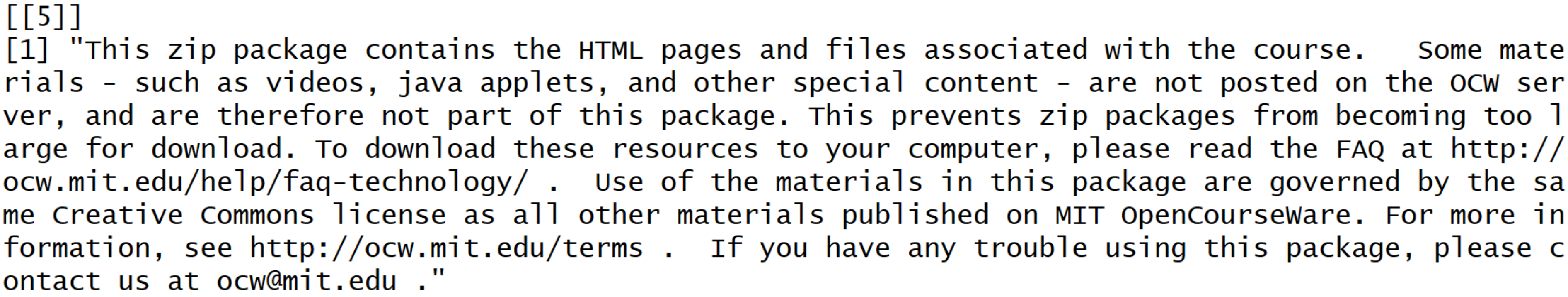


```
 [[2]]
  [1] "Dear Managing Trustee"
  [2] ""
  [3] "It is another great achievement in trust history, as a member of EC of SWET I congratulat
 e you (the MT SWET), secretary SWET Mr. Sajid Raza, all EC members of SWET and all trustees and
  thankful to Bhai M. Baqar. May Allah (ST) bless him and their loved ones. The overall draft of
  agreement seems to be good with few suggestions"
  [4] ""
```

Now we use collapse

```
a <- lapply(filelist, FUN=readLines)
lapply(a, FUN=paste, collapse = “ ”)
```

Only text from 5th document is shown



```
[[5]]
[1] "This zip package contains the HTML pages and files associated with the course.   Some mate
rials - such as videos, java applets, and other special content - are not posted on the OCW ser
ver, and are therefore not part of this package. This prevents zip packages from becoming too l
arge for download. To download these resources to your computer, please read the FAQ at http://
ocw.mit.edu/help/faq-technology/ .  Use of the materials in this package are governed by the sa
me Creative Commons license as all other materials published on MIT OpenCourseWare. For more in
formation, see http://ocw.mit.edu/terms .  If you have any trouble using this package, please c
ontact us at ocw@mit.edu ."
```

cleaning text into corpus using gsub()

```
corpus <- lapply(a, FUN=paste, collapse = “ ”)
```

now corpus have as many elements as many text files we had and collapse will combine all lines of one document into one long text

So now in corpus we have as many long text elements as many files were combined

cleaning text into corpus using gsub()

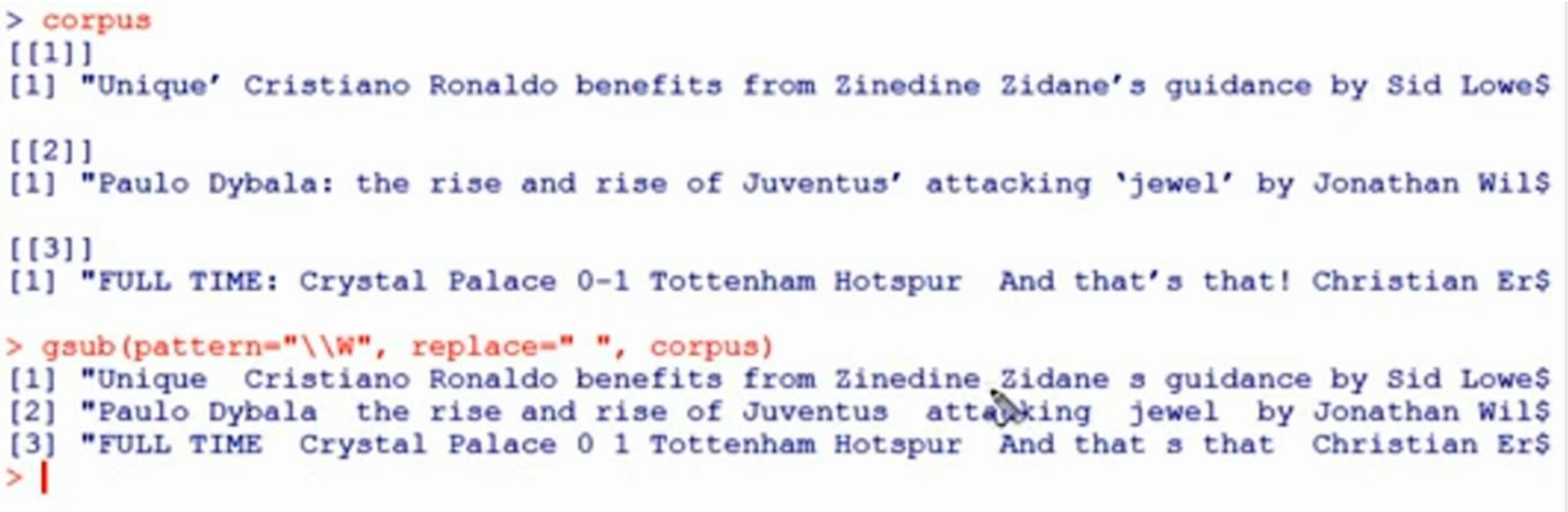
6th part

Needs tm package and wordCloud package

Remove punctuations

corpus2<-gsub(pattern = "\\W",replace = " ", corpus)

Punctuation has been removed

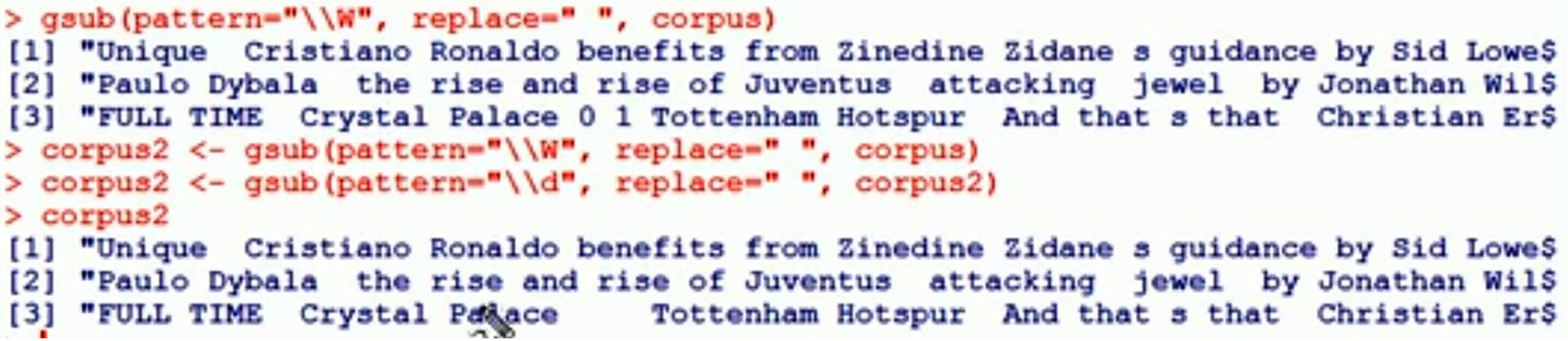


```
> corpus
[[1]]
[1] "Unique’ Cristiano Ronaldo benefits from Zinedine Zidane’s guidance by Sid Lowe$

[[2]]
[1] "Paulo Dybala: the rise and rise of Juventus’ attacking ‘jewel’ by Jonathan Wil$

[[3]]
[1] "FULL TIME: Crystal Palace 0-1 Tottenham Hotspur  And that’s that! Christian Er$

> gsub(pattern="\\W", replace=" ", corpus)
[1] "Unique  Cristiano Ronaldo benefits from Zinedine Zidane s guidance by Sid Lowe$
[2] "Paulo Dybala  the rise and rise of Juventus  attacking  jewel  by Jonathan Wil$
[3] "FULL TIME  Crystal Palace 0 1 Tottenham Hotspur  And that s that  Christian Er$
> |
```

Remove digits

corpus2<-gsub(pattern = "\\d",replace = " ", corpus2)

digits has been removed



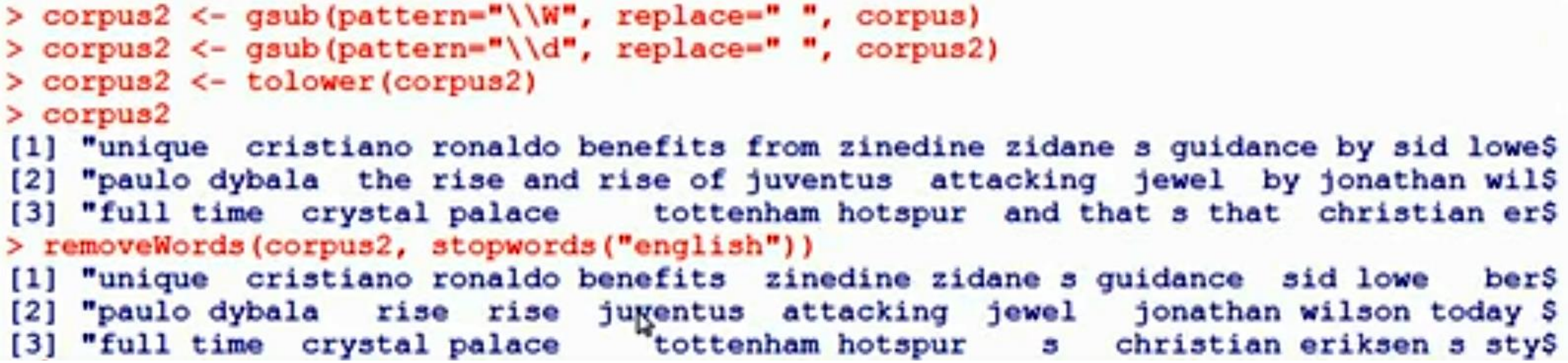
```
> gsub(pattern="\\W", replace=" ", corpus)
[1] "Unique  Cristiano Ronaldo benefits from Zinedine Zidane s guidance by Sid Lowe$
[2] "Paulo Dybala  the rise and rise of Juventus  attacking  jewel  by Jonathan Wil$
[3] "FULL TIME  Crystal Palace 0 1 Tottenham Hotspur  And that s that  Christian Er$
> corpus2 <- gsub(pattern="\\W", replace=" ", corpus)
> corpus2 <- gsub(pattern="\\d", replace=" ", corpus2)
> corpus2
[1] "Unique  Cristiano Ronaldo benefits from Zinedine Zidane s guidance by Sid Lowe$
[2] "Paulo Dybala  the rise and rise of Juventus  attacking  jewel  by Jonathan Wil$
[3] "FULL TIME  Crystal Palace     Tottenham Hotspur  And that s that  Christian Er$
```

Lowercase and Remove stopwords

corpus2<-tolower(corpus2)

removeWords(corpus2,stopwords("english"))

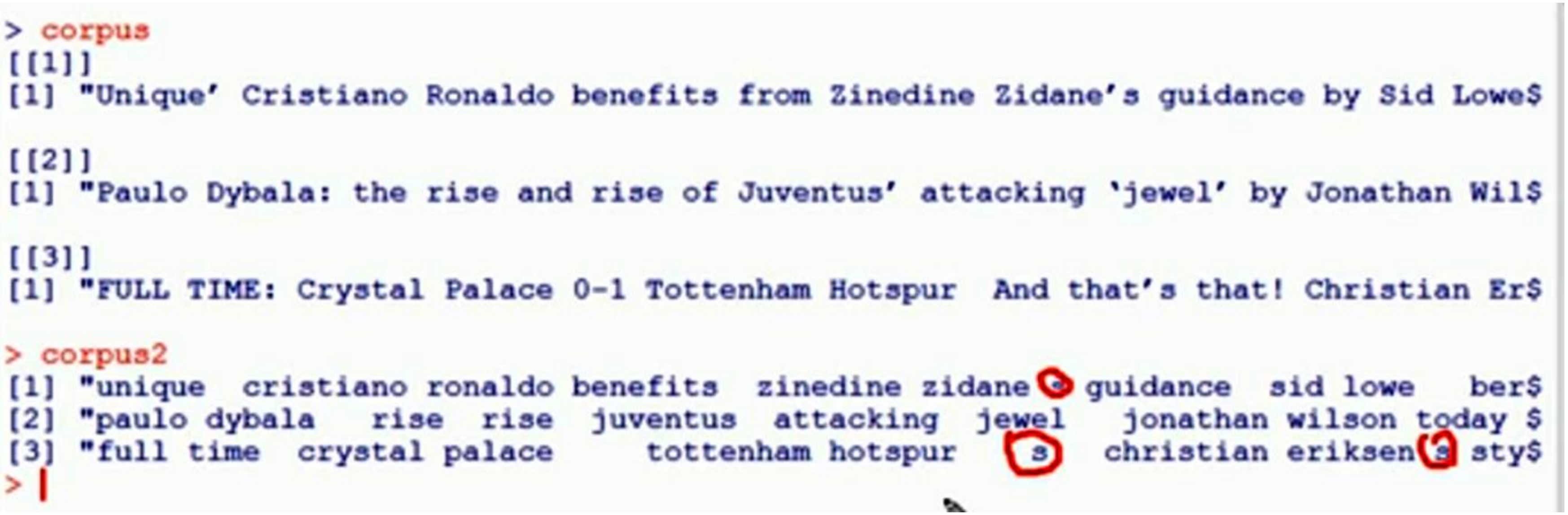
Check the lowercase and stopwords



```
> corpus2 <- gsub(pattern="\\W", replace=" ", corpus)
> corpus2 <- gsub(pattern="\\d", replace=" ", corpus2)
> corpus2 <- tolower(corpus2)
> corpus2
[1] "unique  cristiano ronaldo benefits from zinedine zidane s guidance by sid lowe$
[2] "paulo dybala  the rise and rise of juventus  attacking  jewel  by jonathan wil$
[3] "full time  crystal palace     tottenham hotspur  and that s that  christian er$
> removeWords(corpus2, stopwords("english"))
[1] "unique  cristiano ronaldo benefits  zinedine zidane s guidance  sid lowe   ber$
[2] "paulo dybala   rise  rise  juventus  attacking  jewel   jonathan wilson today $
[3] "full time  crystal palace     tottenham hotspur    s  christian eriksen s sty$
```

Now we remove single letter words

Check the single letter words in the corpus2

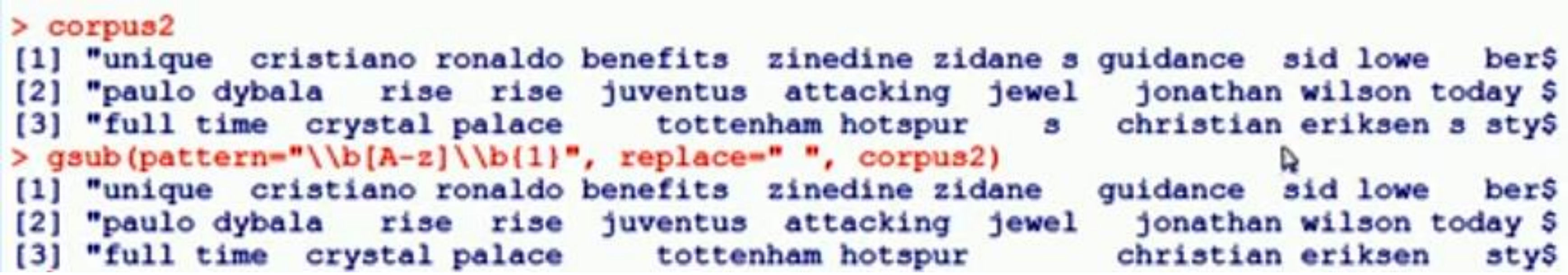


```
> corpus
[[1]]
[1] "Unique’ Cristiano Ronaldo benefits from Zinedine Zidane’s guidance by Sid Lowe$

[[2]]
[1] "Paulo Dybala: the rise and rise of Juventus’ attacking ‘jewel’ by Jonathan Wil$

[[3]]
[1] "FULL TIME: Crystal Palace 0-1 Tottenham Hotspur  And that’s that! Christian Er$

> corpus2
[1] "unique  cristiano ronaldo benefits  zinedine zidane s guidance  sid lowe   ber$
[2] "paulo dybala   rise  rise  juventus  attacking  jewel   jonathan wilson today $
[3] "full time  crystal palace     tottenham hotspur    s   christian eriksen s sty$
> |
```

Now we remove single letter words

corpus2<-gsub(pattern = "\\b[A-z]\\b{1}",replace=" ",corpus2)

Single letter words has been removed

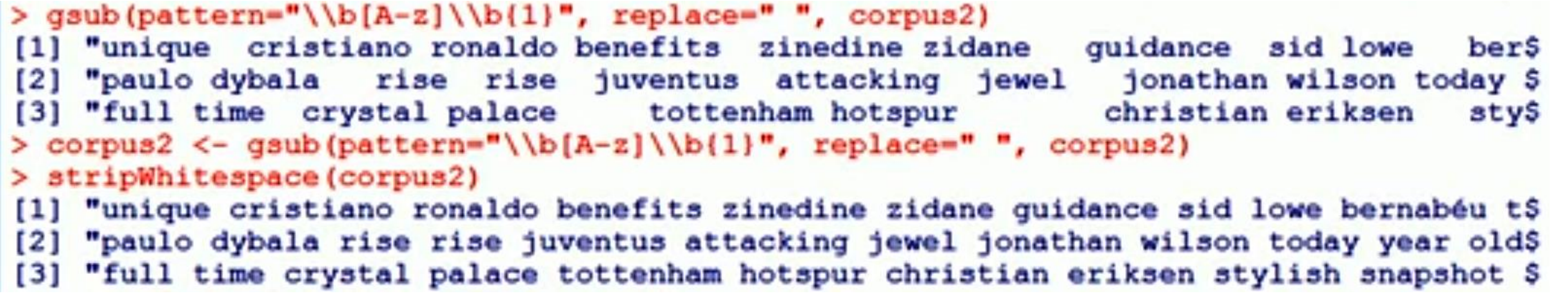


```
> corpus2
[1] "unique  cristiano ronaldo benefits  zinedine zidane s guidance  sid lowe   ber$
[2] "paulo dybala   rise  rise  juventus  attacking  jewel   jonathan wilson today $
[3] "full time  crystal palace     tottenham hotspur    s   christian eriksen s sty$
> gsub(pattern="\\b[A-z]\\b{1}", replace=" ", corpus2)
[1] "unique  cristiano ronaldo benefits  zinedine zidane   guidance  sid lowe   ber$
[2] "paulo dybala   rise  rise  juventus  attacking  jewel   jonathan wilson today $
[3] "full time  crystal palace     tottenham hotspur        christian eriksen   sty$
```

Removing whitespaces

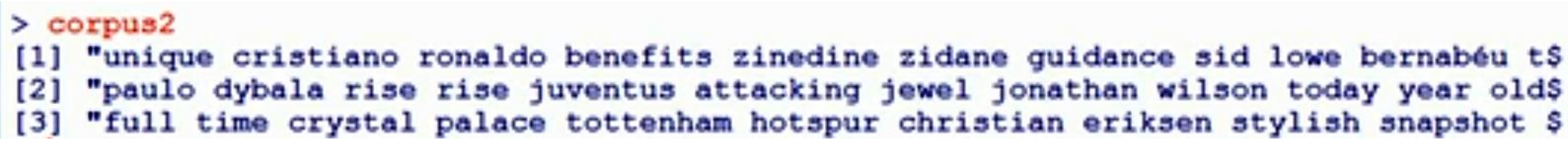
corpus2<-stripWhitespace(corpus2)

Whitespaces has been removed



```
> gsub(pattern="\\b[A-z]\\b{1}", replace=" ", corpus2)
[1] "unique  cristiano ronaldo benefits  zinedine zidane   guidance  sid lowe   ber$
[2] "paulo dybala   rise  rise  juventus  attacking  jewel   jonathan wilson today $
[3] "full time  crystal palace     tottenham hotspur        christian eriksen   sty$
> corpus2 <- gsub(pattern="\\b[A-z]\\b{1}", replace=" ", corpus2)
> stripWhitespace(corpus2)
[1] "unique cristiano ronaldo benefits zinedine zidane guidance sid lowe bernabéu t$
[2] "paulo dybala rise rise juventus attacking jewel jonathan wilson today year old$
[3] "full time crystal palace tottenham hotspur christian eriksen stylish snapshot $
```

Cleaned corpus

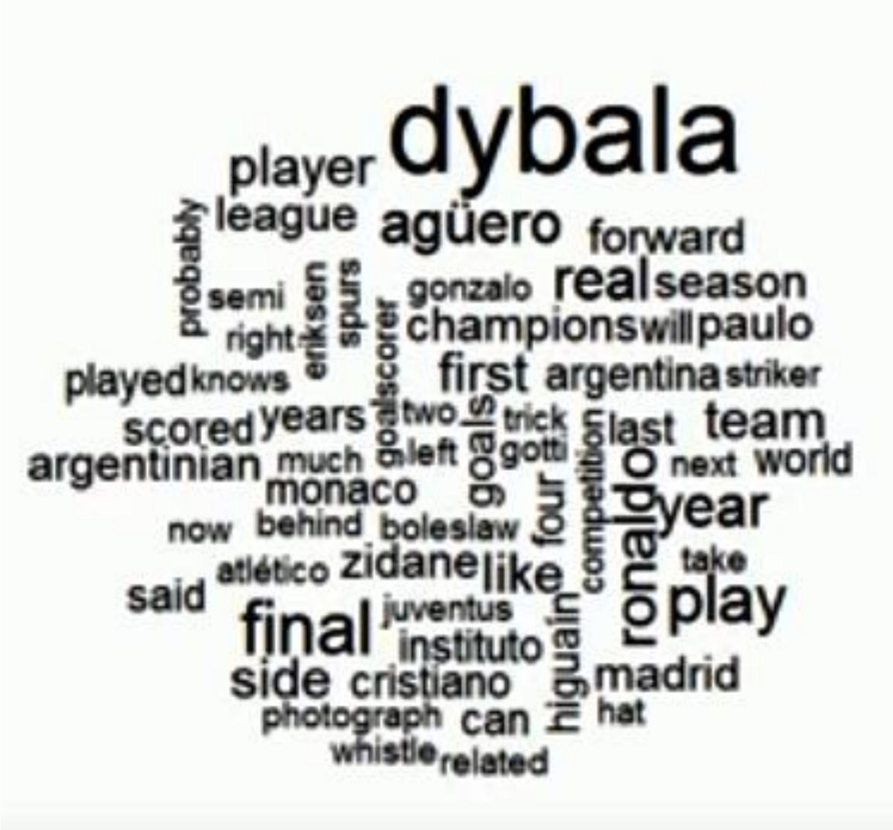


```
> corpus2
[1] "unique cristiano ronaldo benefits zinedine zidane guidance sid lowe bernabéu t$
[2] "paulo dybala rise rise juventus attacking jewel jonathan wilson today year old$
[3] "full time crystal palace tottenham hotspur christian eriksen stylish snapshot $
```

Making wordcloud

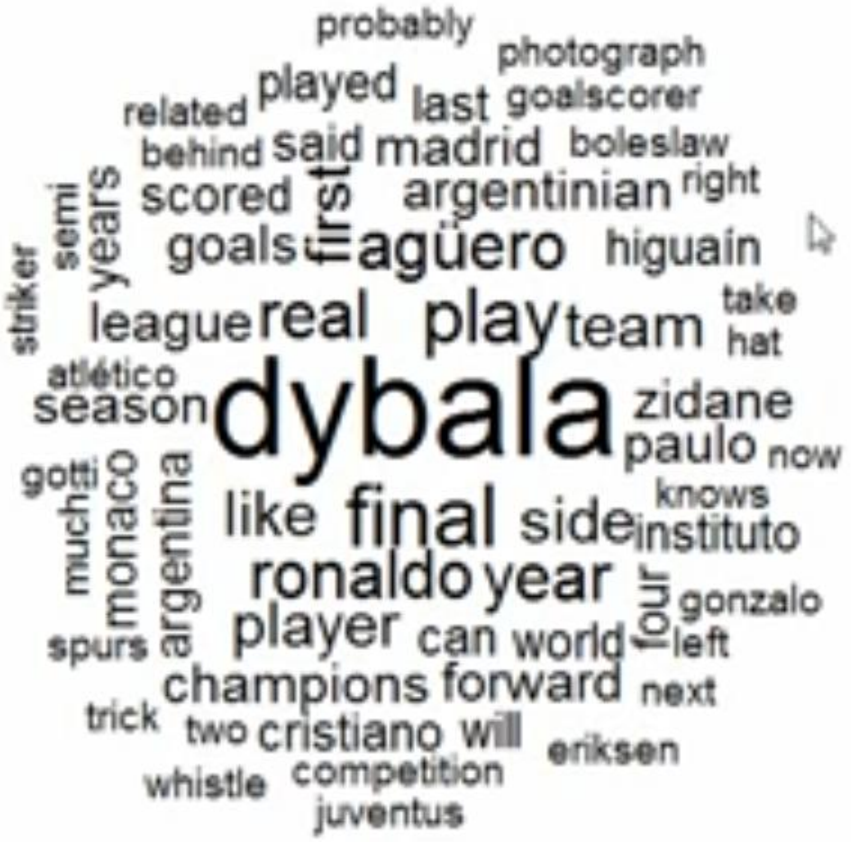
6th part started

wordcloud(corpus2)



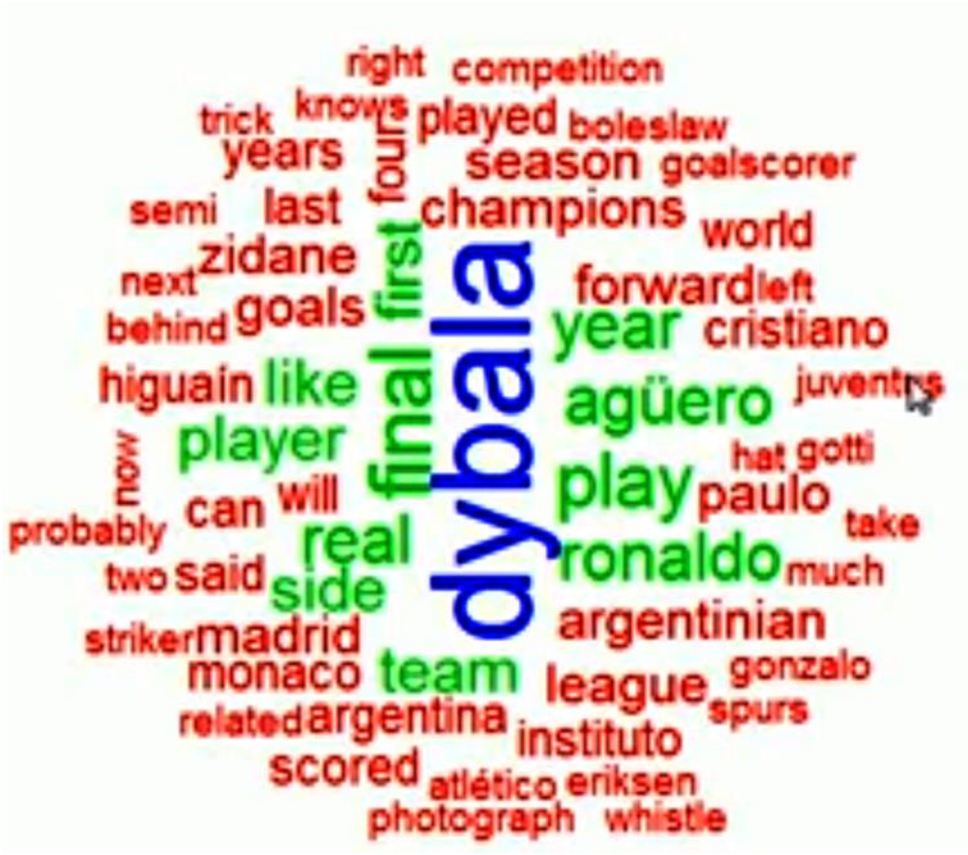
random.order = False

```
wordcloud(corpus2, random.order=FALSE)
```



rainbow(3)

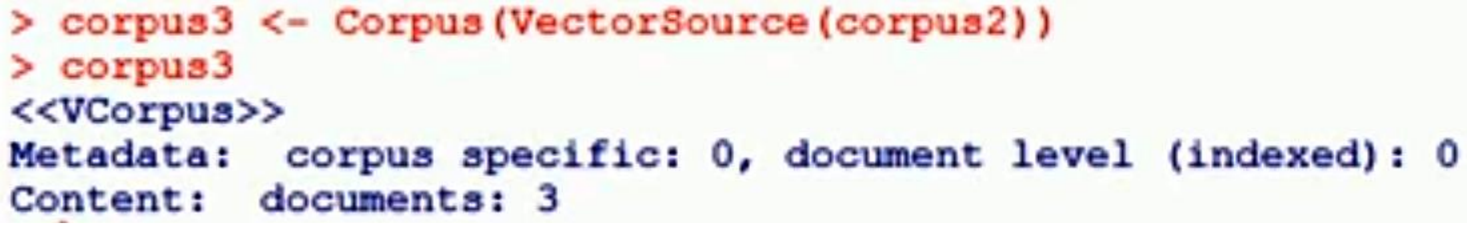
```
wordcloud(corpus2, random.order=FALSE,color=rainbow(3))
```



Comparing wordclouds

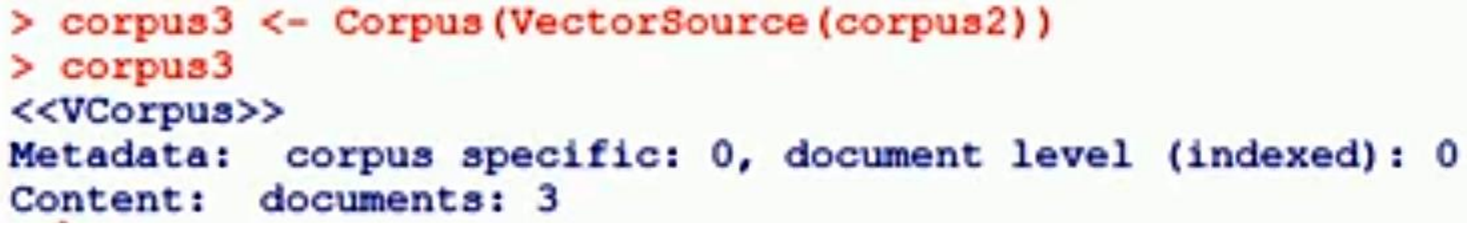
corpus2 is not an official corpus of tm package now we create official corpus

corpus3 <- Corpus(VectorSource(corpus2))



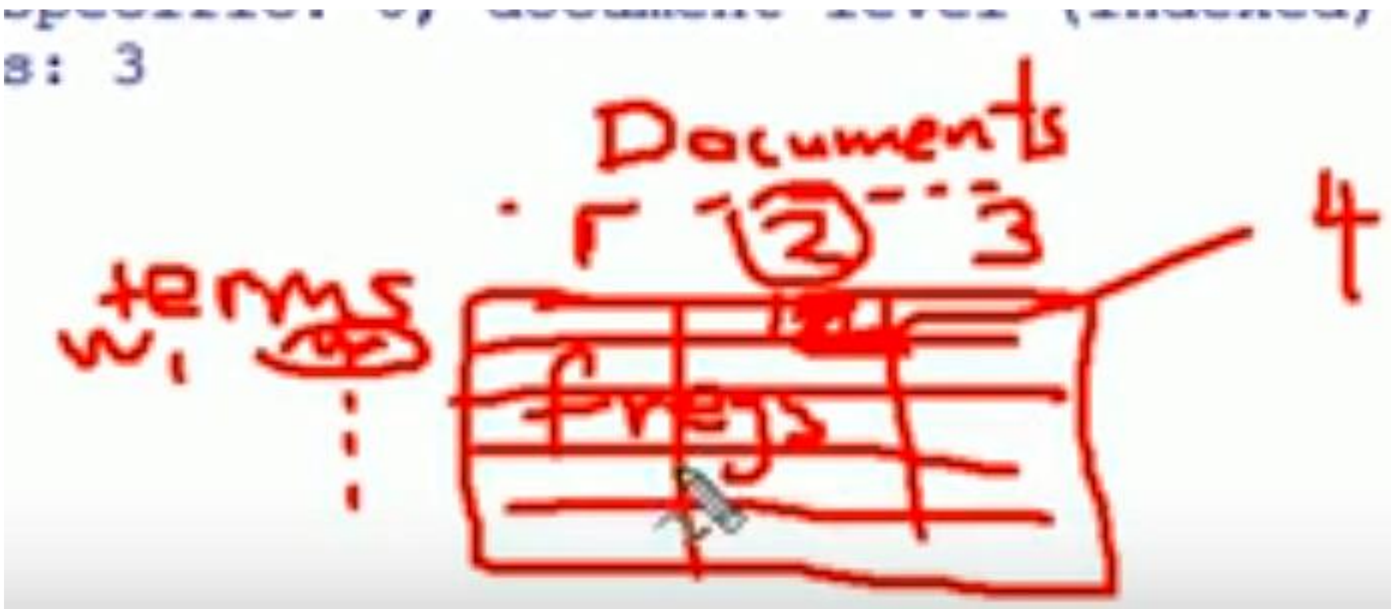
```
> corpus3 <- Corpus(VectorSource(corpus2))
> corpus3
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 3
```

Structure of corpus in memory



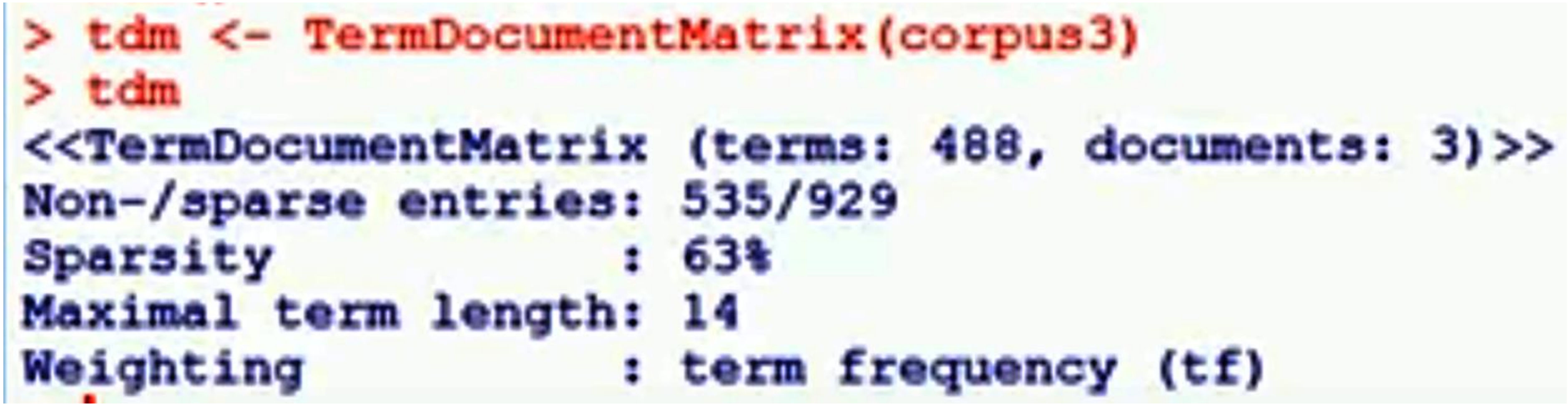
```
> corpus3 <- Corpus(VectorSource(corpus2))
> corpus3
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 3
```

Each unique word is listed left side and in columns we have document number in each cell we have frequency of each word in each document



Term documents matrix

tdm <- TermDocumentMatrix(corpus3)



```
> tdm <- TermDocumentMatrix(corpus3)
> tdm
<<TermDocumentMatrix (terms: 488, documents: 3)>>
Non-/sparse entries: 535/929
Sparsity           : 63%
Maximal term length: 14
Weighting          : term frequency (tf)
```

488 unique words i.e rows

3 Documents i.e. 3 columns

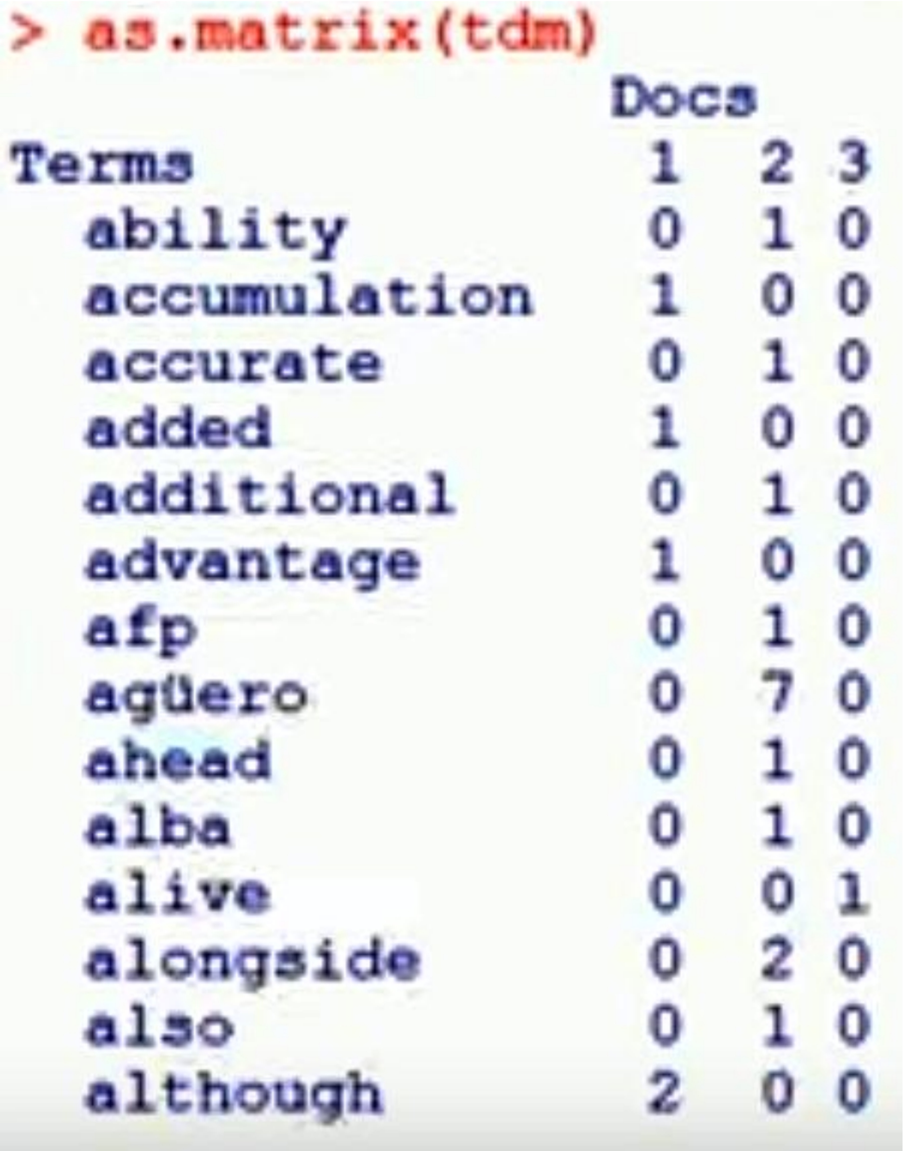
929 empty cells

535 non empty cells

Total cells   488*3   =   1464 cells =   929 +535

Converting tdm into matrix

m <- as.matrix(tdm)



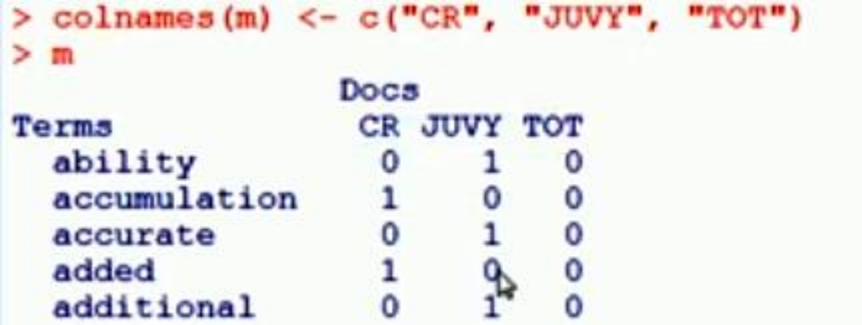
```
> as.matrix(tdm)
                  Docs
Terms              1  2 3
  ability          0  1 0
  accumulation     1  0 0
  accurate         0  1 0
  added            1  0 0
  additional       0  1 0
  advantage        1  0 0
  afp              0  1 0
  agüero           0  7 0
  ahead            0  1 0
  alba             0  1 0
  alive            0  0 1
  alongside        0  2 0
  also             0  1 0
  although         2  0 0
```

Changing column names

```
 m <- as.matrix(tdm)
colnames(m)
```

“1” “2” “3”

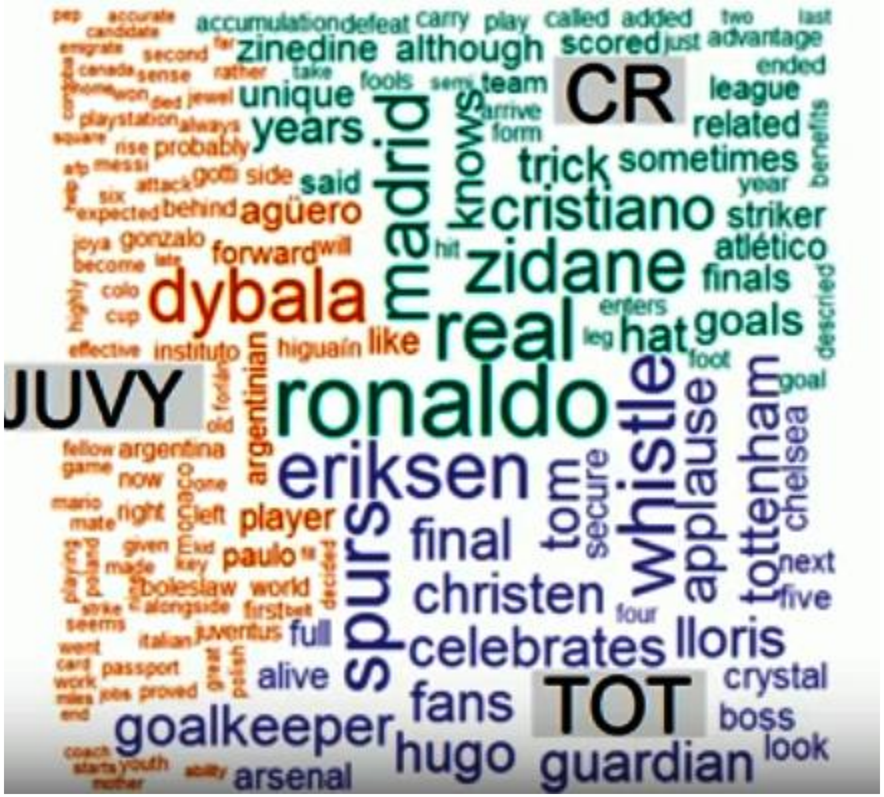
```
colnames(m) <- c(“CR”,”JUVY”,”TOT”)
```



```
> colnames(m) <- c("CR", "JUVY", "TOT")
> m
                Docs
Terms             CR JUVY TOT
  ability          0    1   0
  accumulation     1    0   0
  accurate         0    1   0
  added            1    0   0
  additional       0    1   0
```

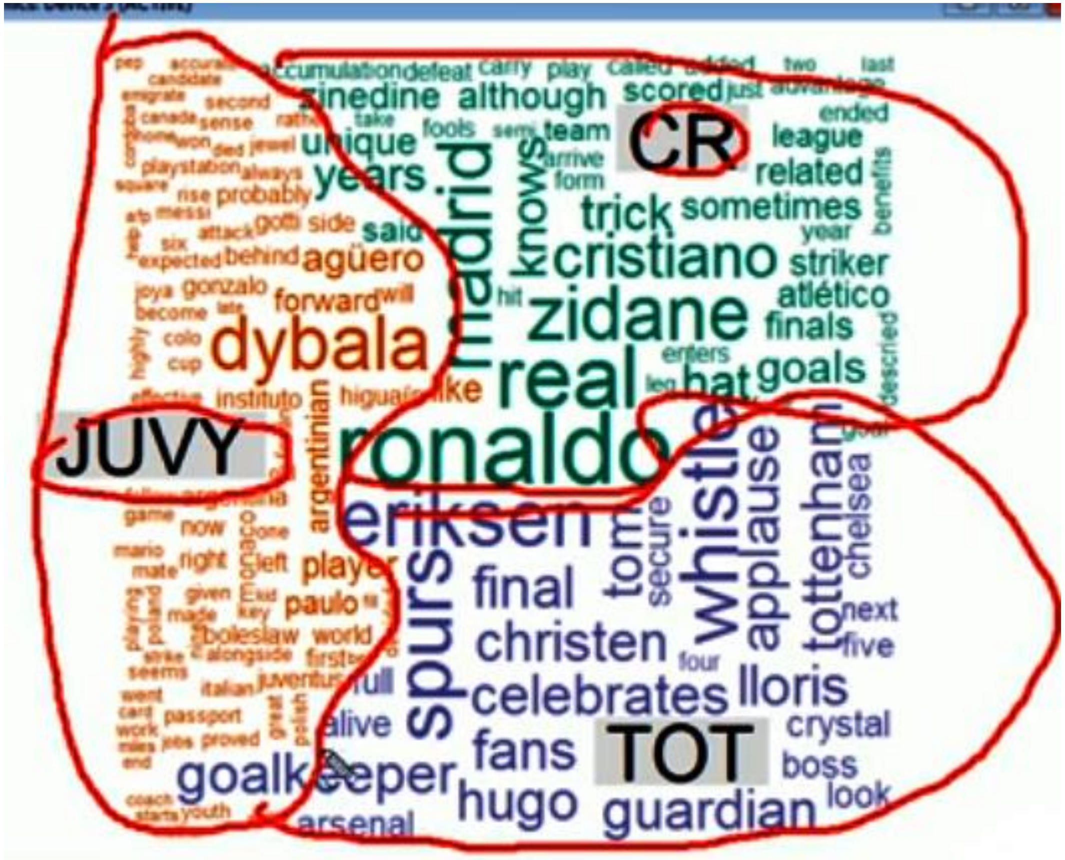
Comparison of wordclouds

comparison.cloud(m)



Comparison of wordclouds

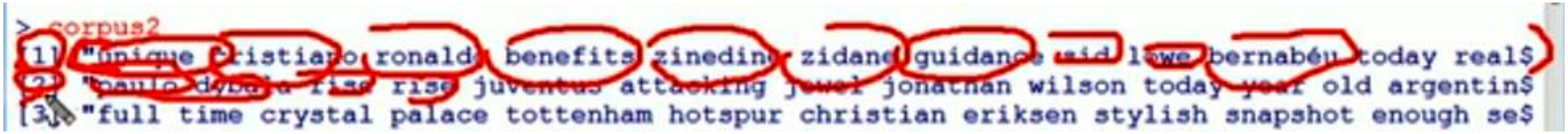
```
comparison.cloud(m)
```



Calculation of sentiment score

8th part

Entire corpus contains three documents now we have to convert all three documents into wordBags



```
> corpus2
[1] "unique cristiano ronaldo benefits zinedine zidane guidance said lowe bernabéu today real$
[2] "paulo dybala rise rise juventus attacking jewel jonathan wilson today year old argentin$
[3] "full time crystal palace tottenham hotspur christian eriksen stylish snapshot enough se$
```

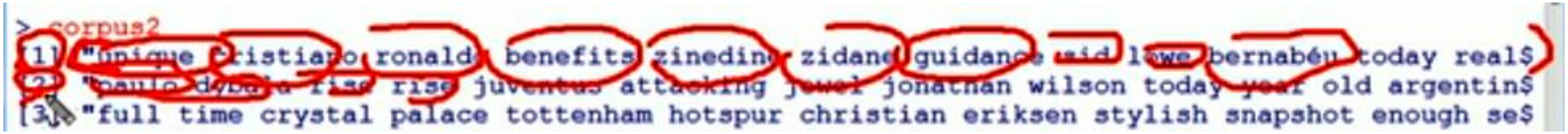
Needs stringr package for the following code

Str_split(corpus2, pattern = “\\s+”)

Calculation of sentiment score

8th part

Entire corpus contains three documents now we have to convert all three documents into wordBags



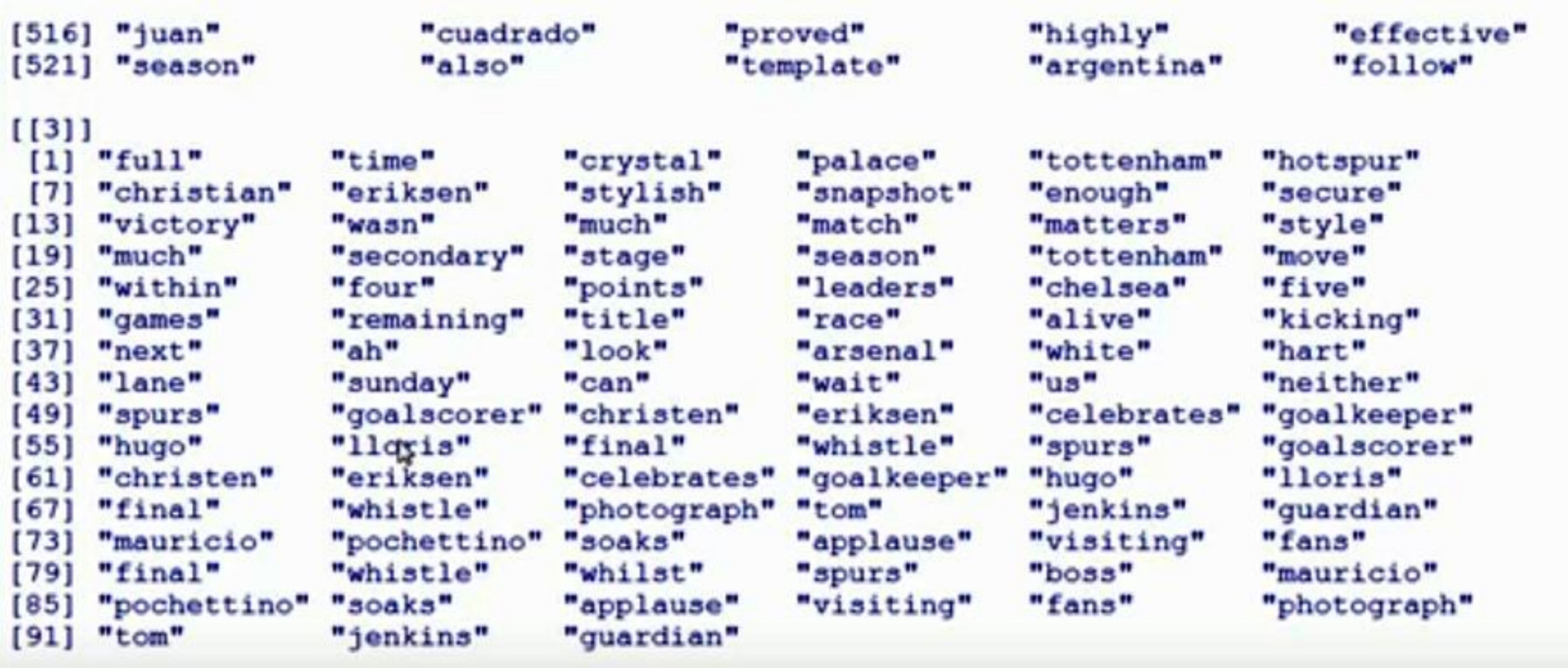
```
> corpus2
[1] "unique cristiano ronaldo benefits zinedine zidane guidance said lowe bernabéu today real$
[2] "paulo dybala rise rise juventus attacking jewel jonathan wilson today year old argentin$
[3] "full time crystal palace tottenham hotspur christian eriksen stylish snapshot enough se$
```

Needs stringr package for the following code

Str_split(corpus2, pattern = “\\s+”)

Three wordBags are created

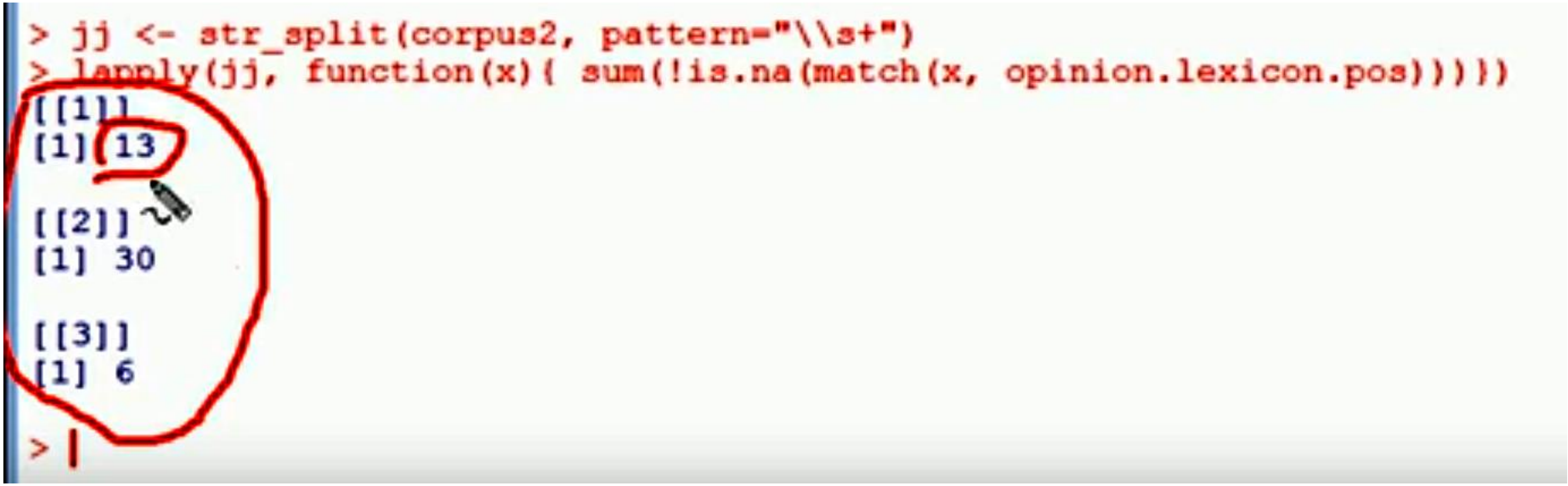
jj <- Str_split(corpus2, pattern = “\\s+”)



```
[516] "juan"        "cuadrado"    "proved"      "highly"      "effective"
[521] "season"      "also"        "template"    "argentina"   "follow"

[[3]]
 [1] "full"        "time"        "crystal"     "palace"      "tottenham"   "hotspur"
 [7] "christian"   "eriksen"     "stylish"     "snapshot"    "enough"      "secure"
[13] "victory"     "wasn"        "much"        "match"       "matters"     "style"
[19] "much"        "secondary"   "stage"       "season"      "tottenham"   "move"
[25] "within"      "four"        "points"      "leaders"     "chelsea"     "five"
[31] "games"       "remaining"   "title"       "race"        "alive"       "kicking"
[37] "next"        "ah"          "look"        "arsenal"     "white"       "hart"
[43] "lane"        "sunday"      "can"         "wait"        "us"          "neither"
[49] "spurs"       "goalscorer"  "christen"    "eriksen"     "celebrates"  "goalkeeper"
[55] "hugo"        "lloris"      "final"       "whistle"     "spurs"       "goalscorer"
[61] "christen"    "eriksen"     "celebrates"  "goalkeeper"  "hugo"        "lloris"
[67] "final"       "whistle"     "photograph"  "tom"         "jenkins"     "guardian"
[73] "mauricio"    "pochettino"  "soaks"       "applause"    "visiting"    "fans"
[79] "final"       "whistle"     "whilst"      "spurs"       "boss"        "mauricio"
[85] "pochettino"  "soaks"       "applause"    "visiting"    "fans"        "photograph"
[91] "tom"         "jenkins"     "guardian"
```

Now matching with +ve words

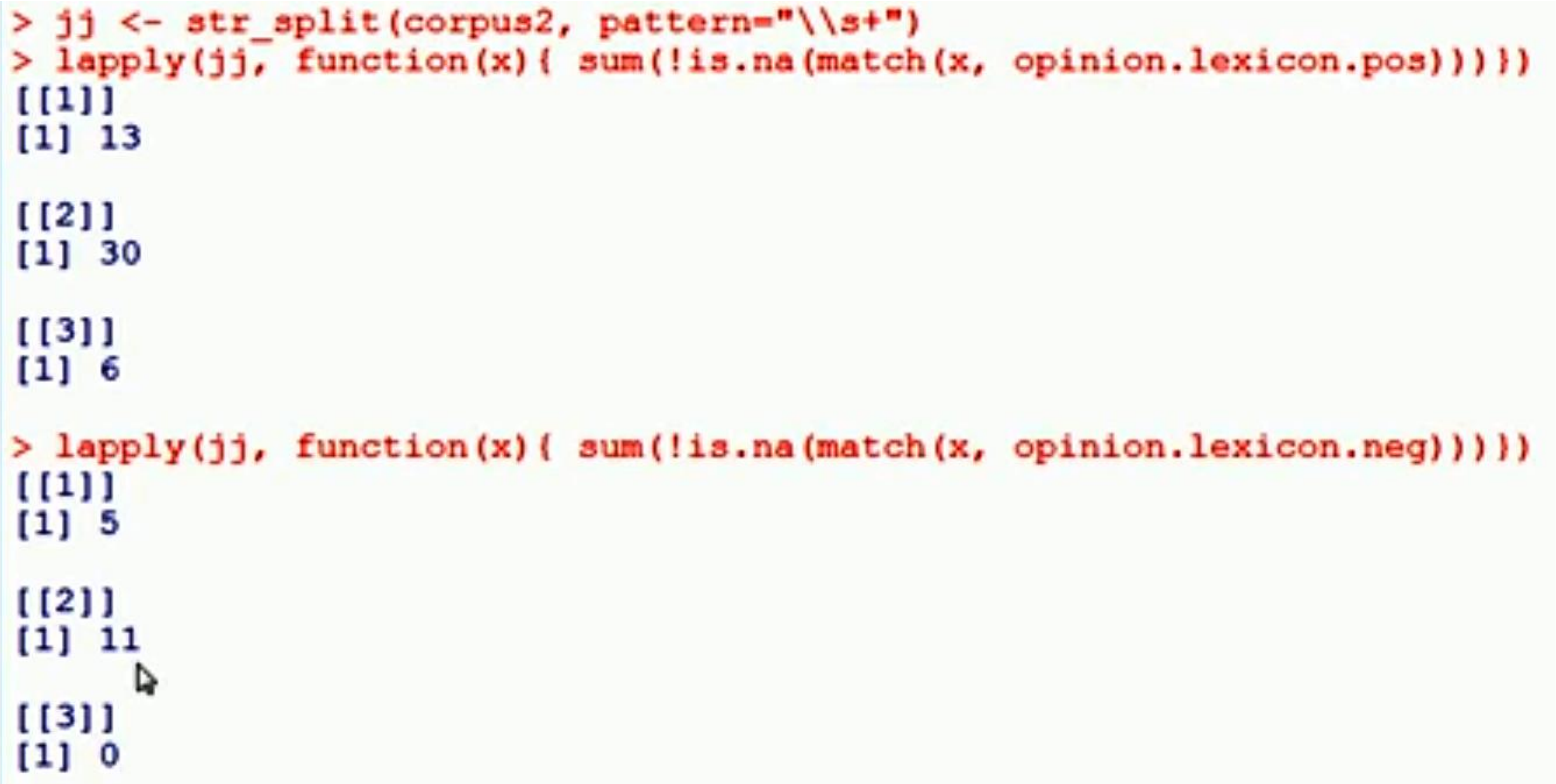
```
lapply(jj, function(x){
                sum(!is.na(match(x,opinion.lexicon.pos)))
                        })
```



```
> jj <- str_split(corpus2, pattern="\\s+")
> lapply(jj, function(x){ sum(!is.na(match(x, opinion.lexicon.pos)))})
[[1]]
[1] 13

[[2]]
[1] 30

[[3]]
[1] 6

> 
```

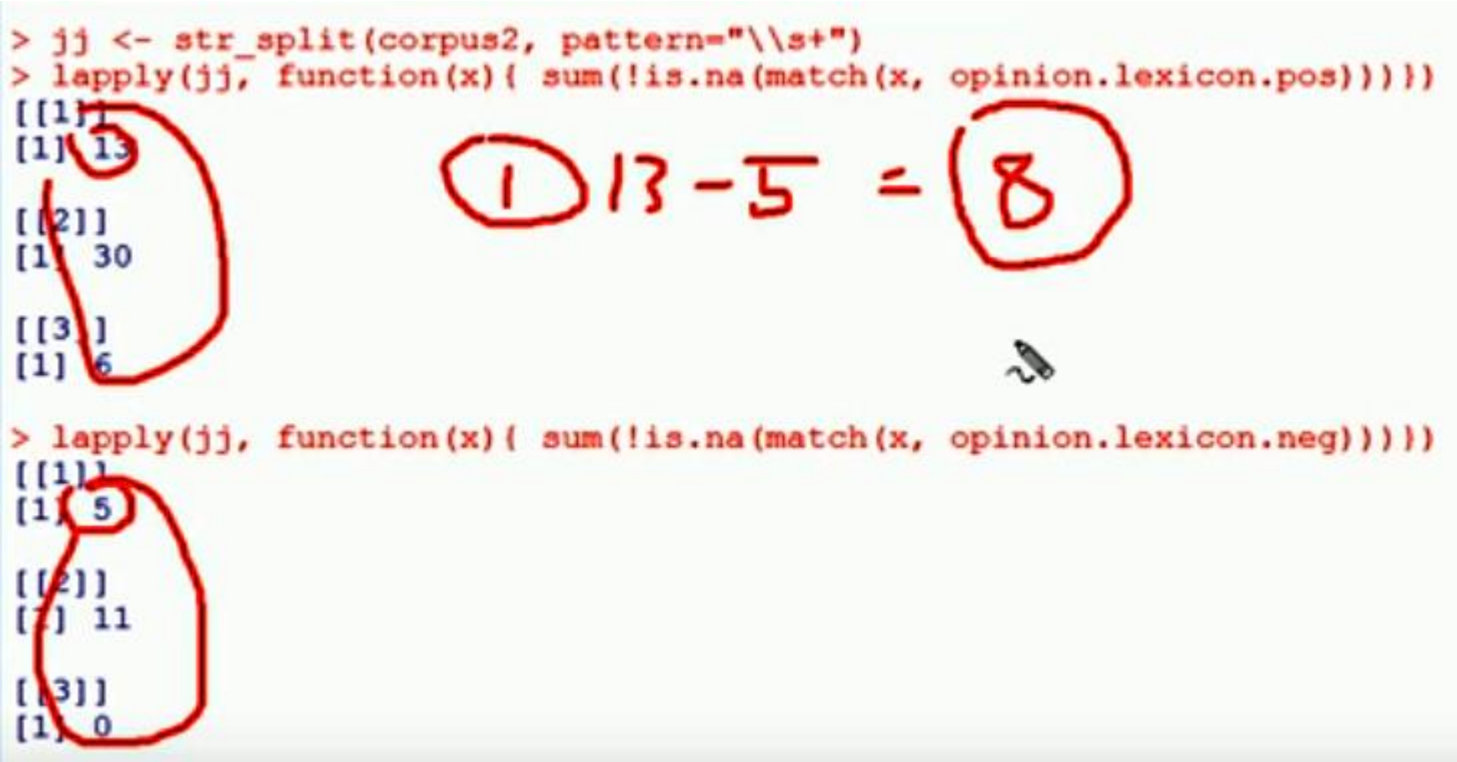
Now matching with –ve words

```
lapply(jj, function(x){
                    sum(!is.na(match(x,opinion.lexicon.neg)))
                              })
```



```
> jj <- str_split(corpus2, pattern="\\s+")
> lapply(jj, function(x){ sum(!is.na(match(x, opinion.lexicon.pos)))})
[[1]]
[1] 13

[[2]]
[1] 30

[[3]]
[1] 6

> lapply(jj, function(x){ sum(!is.na(match(x, opinion.lexicon.neg)))})
[[1]]
[1] 5

[[2]]
[1] 11

[[3]]
[1] 0
```

Sentiment score for doc 1



```
> jj <- str_split(corpus2, pattern="\\s+")
> lapply(jj, function(x){ sum(!is.na(match(x, opinion.lexicon.pos)))})
[[1]]
[1] 13

[[2]]
[1] 30

[[3]]
[1] 6

> lapply(jj, function(x){ sum(!is.na(match(x, opinion.lexicon.neg)))})
[[1]]
[1] 5

[[2]]
[1] 11

[[3]]
[1] 0
```

① 13 − 5 = ⑧

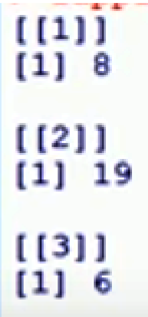
Now matching with –ve words

```
lapply(jj, function(x){
                    sum(!is.na(match(x,opinion.lexicon.pos)))
                              -
                  sum(!is.na(match(x,opinion.lexicon.neg)))
                                   })
```

Sentiment Score for doc 1

Sentiment Score for doc2

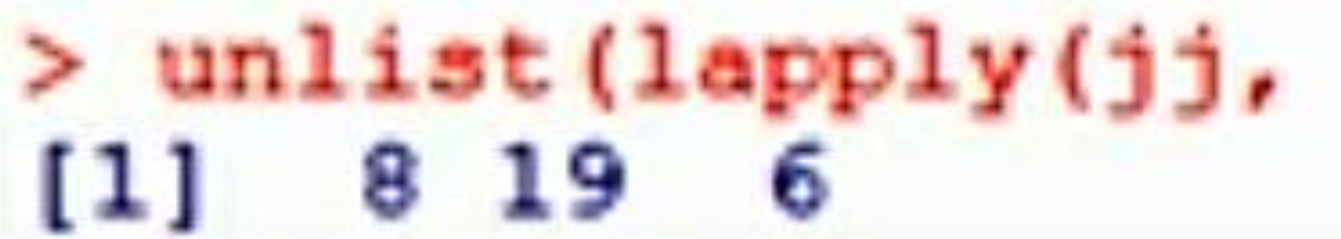
Sentiment Score for doc 3



```
[[1]]
[1]  8

[[2]]
[1]  19

[[3]]
[1]  6
```

Unlist sentiment score

```
Unlist(lapply(jj, function(x){
                    sum(!is.na(match(x,opinion.lexicon.pos)))
                              -
                  sum(!is.na(match(x,opinion.lexicon.neg)))
                                    }))
```



```
> unlist(lapply(jj,
[1]   8 19   6
```

Unlist sentiment score

```
Score<-Unlist(lapply(jj, function(x){
                    sum(!is.na(match(x,opinion.lexicon.pos)))
                              -
                  sum(!is.na(match(x,opinion.lexicon.neg)))
                                    }))


mean(score)
sd(score)
hist(score)
```