Group-4-Project-3 report

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

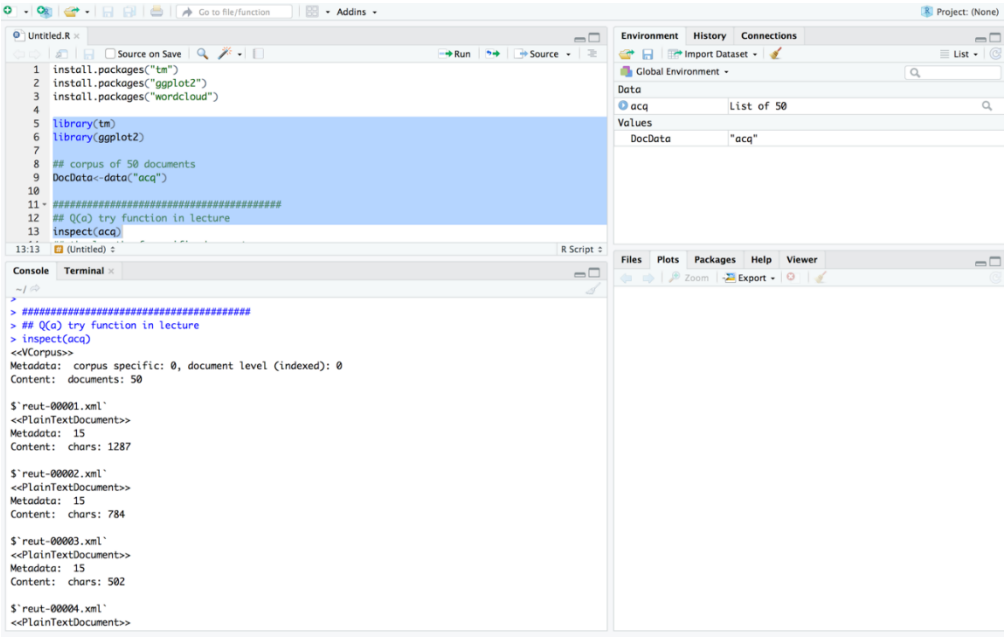
1. R functions:

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):**

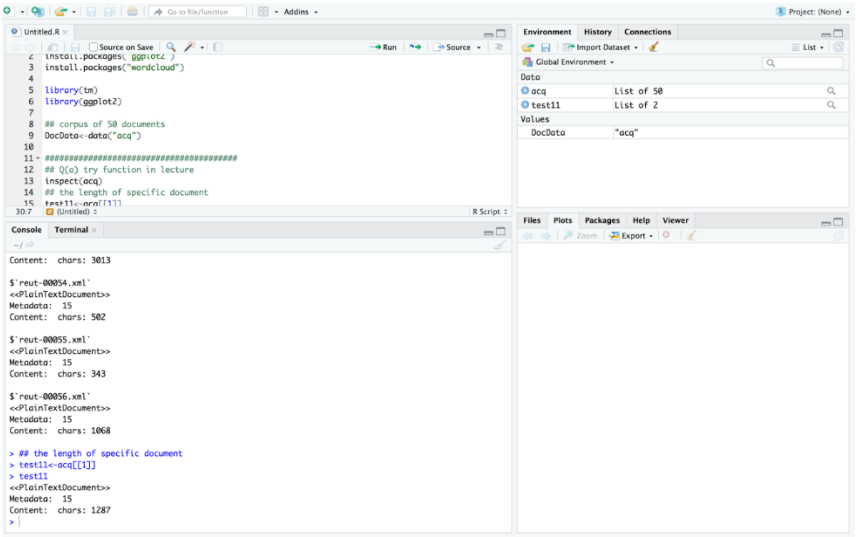
a. **For the complete set of documents, try the functions in lecture 9. What happens? Does it yield anything understandable about the documents.**

1. Inspect()

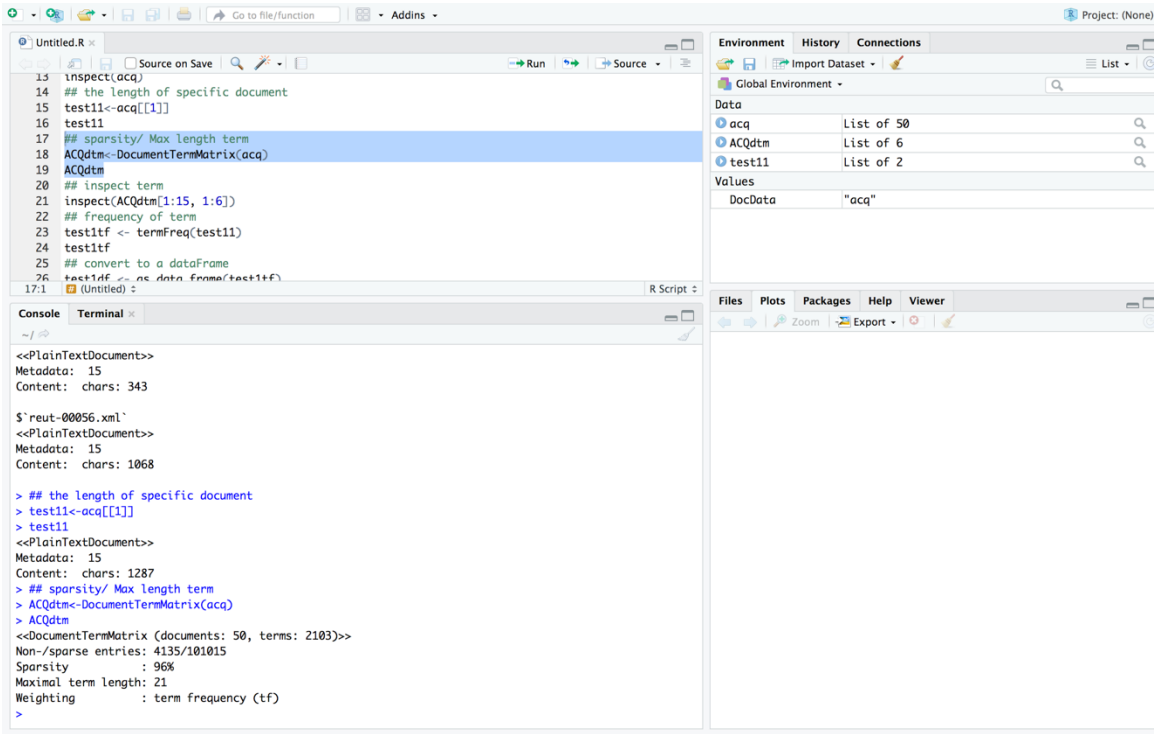


Yes, from this function, we can find the induction of the document.

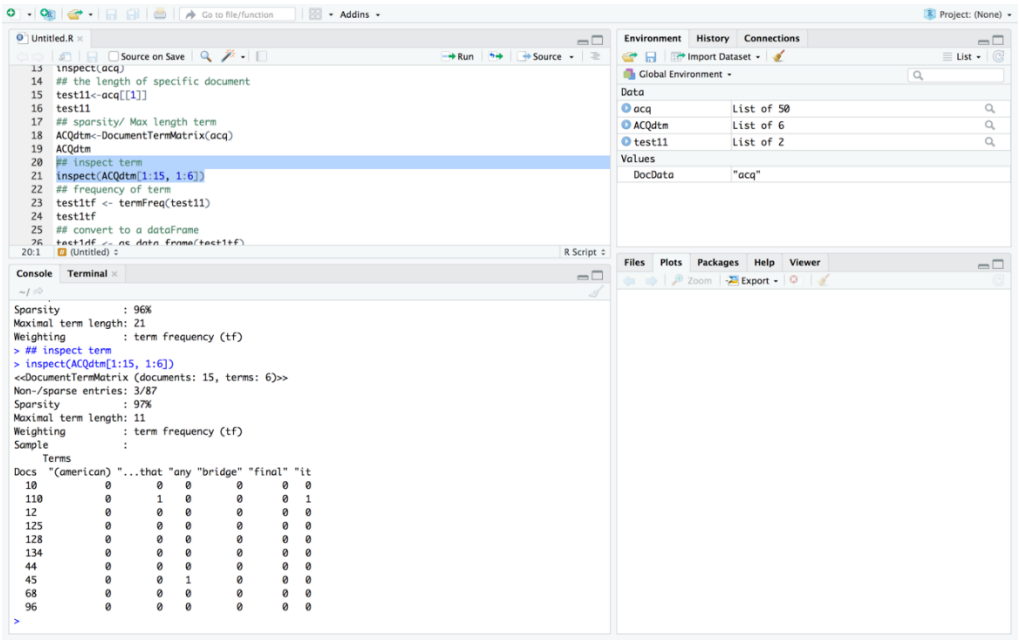
2. We can find the length of specific document



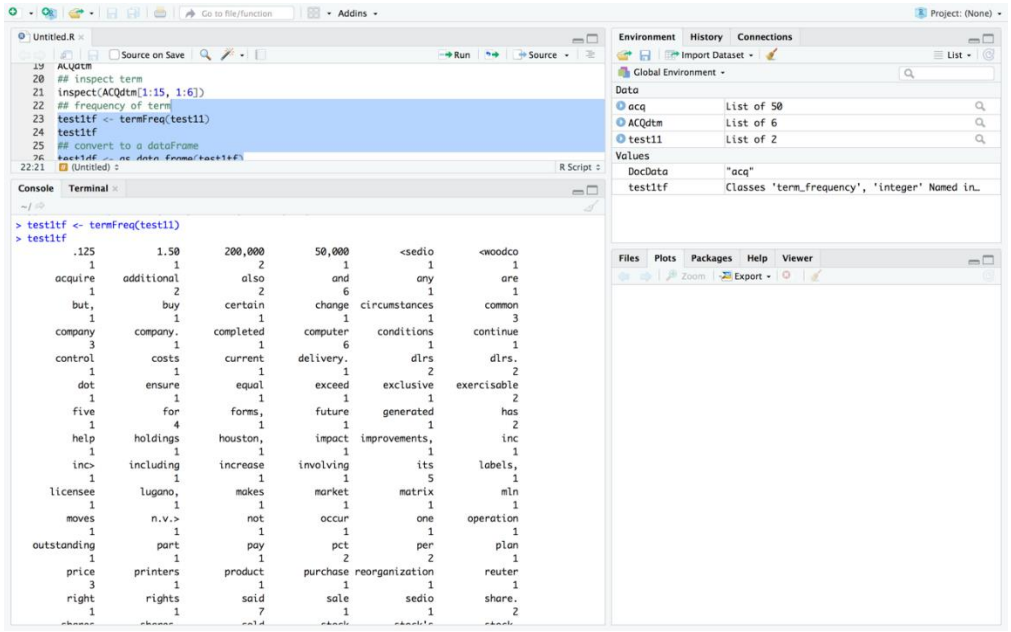
3. We can find sparsity or max length term



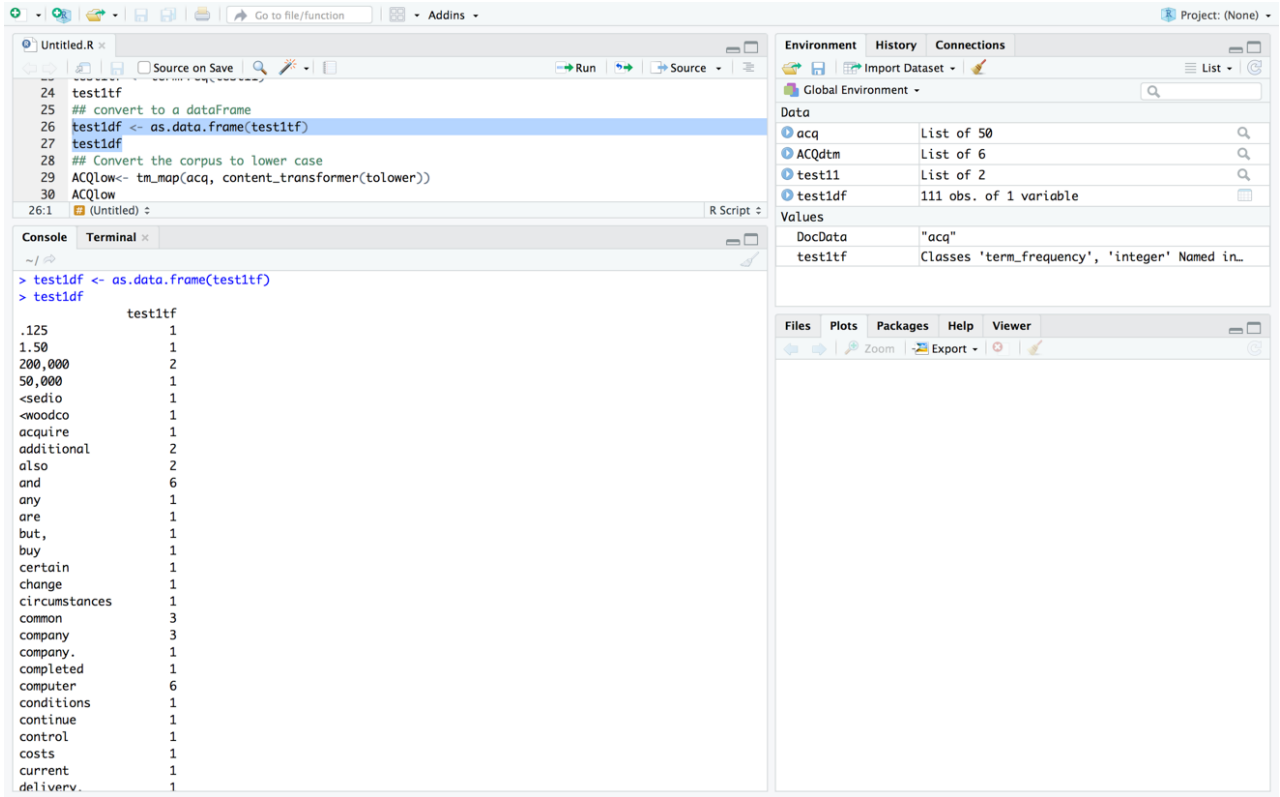
4. We can inspect term



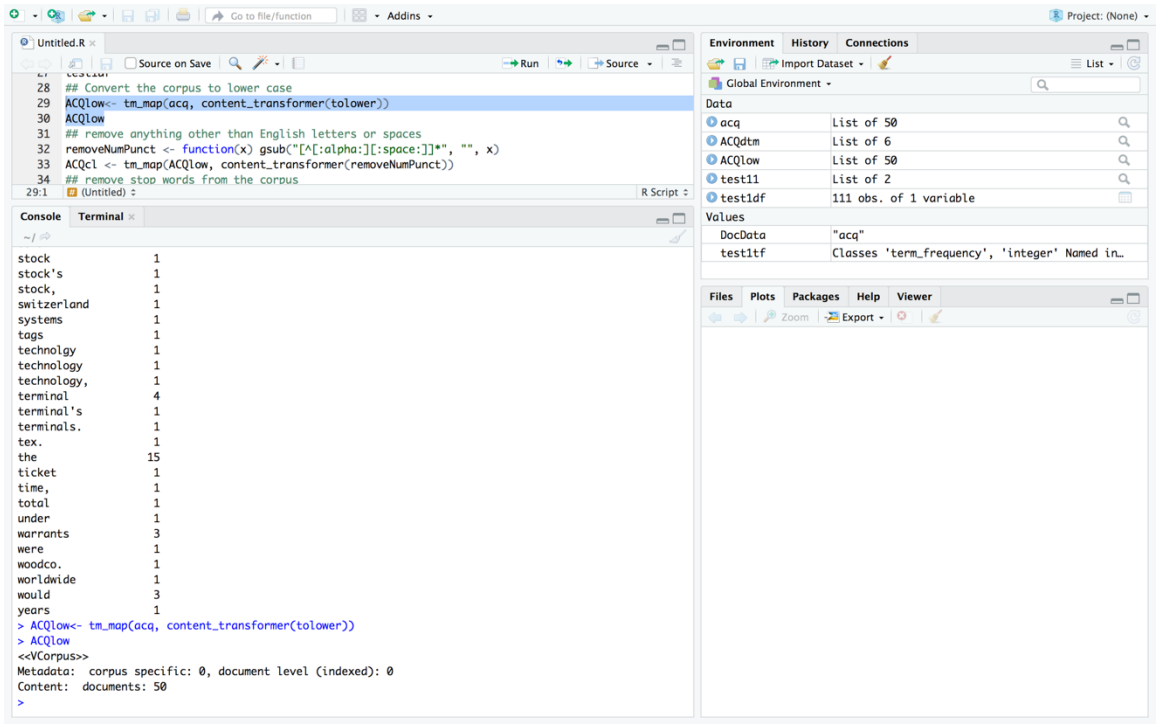
5. We can find frequency of term



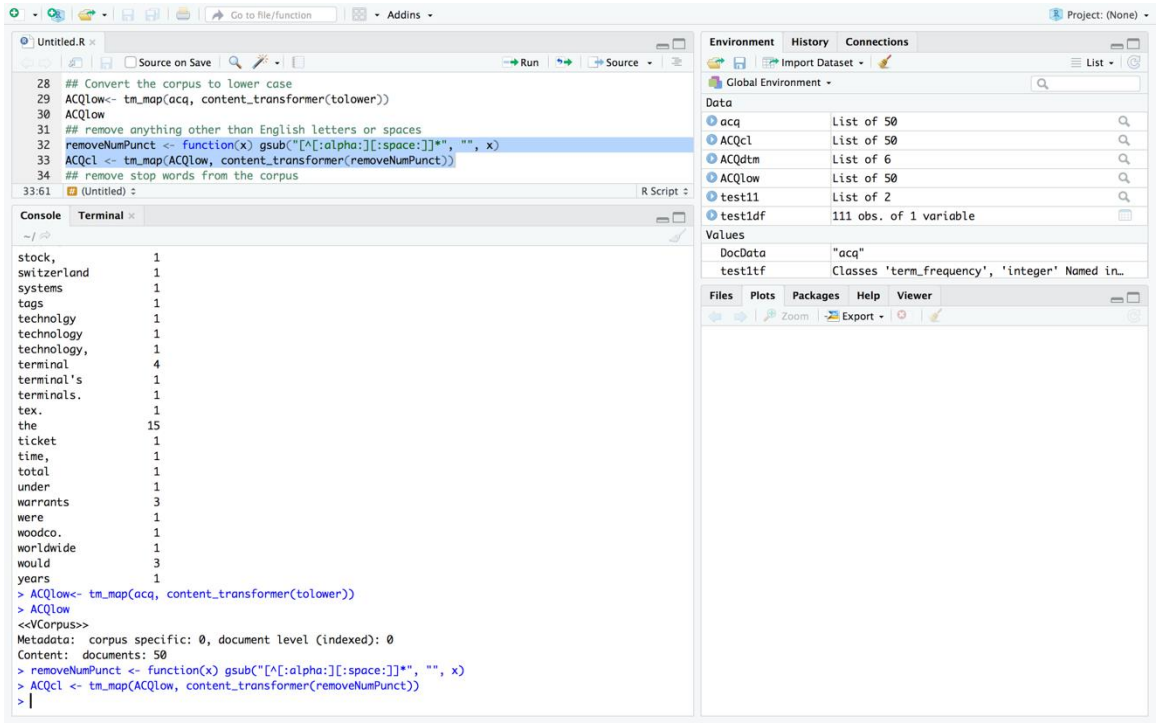
6. Convert to data frame



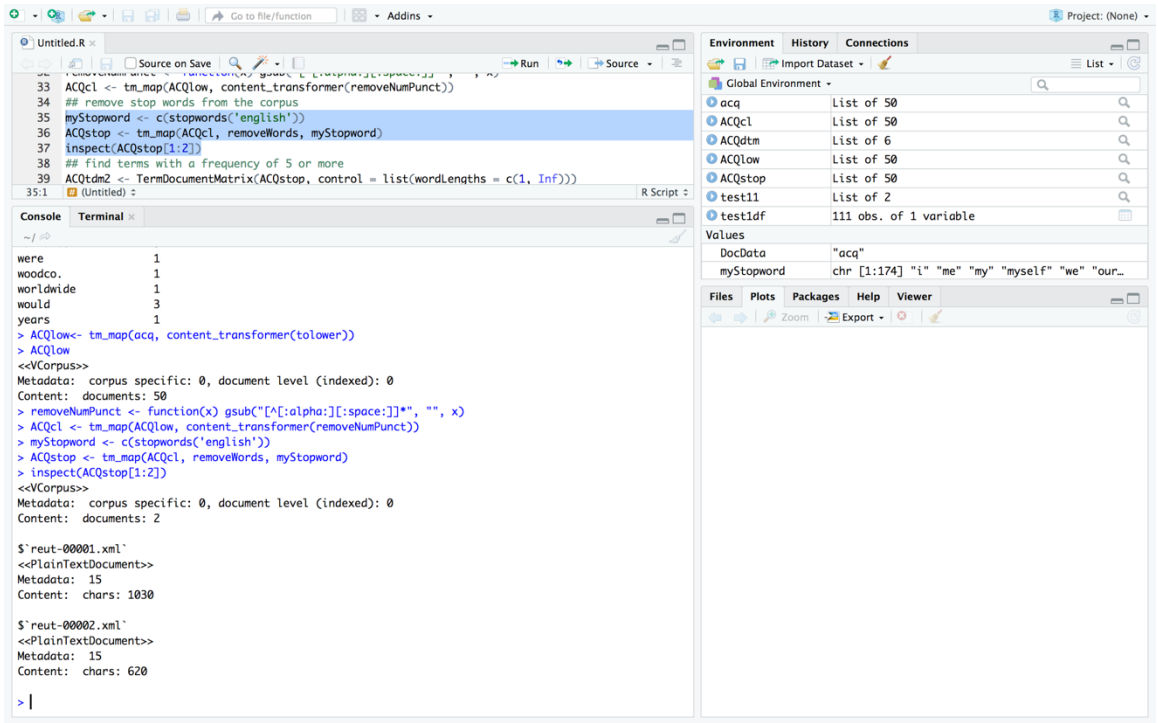
7. Convert the corpus to lower case



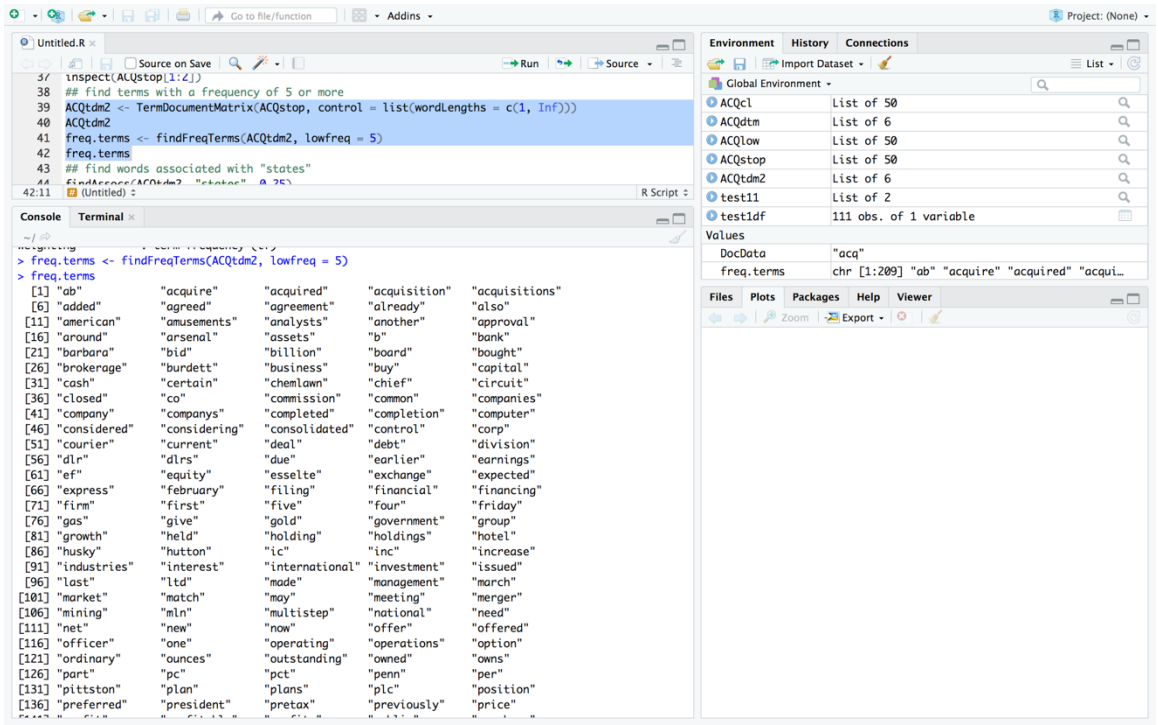
8. remove anything other than English letters or spaces



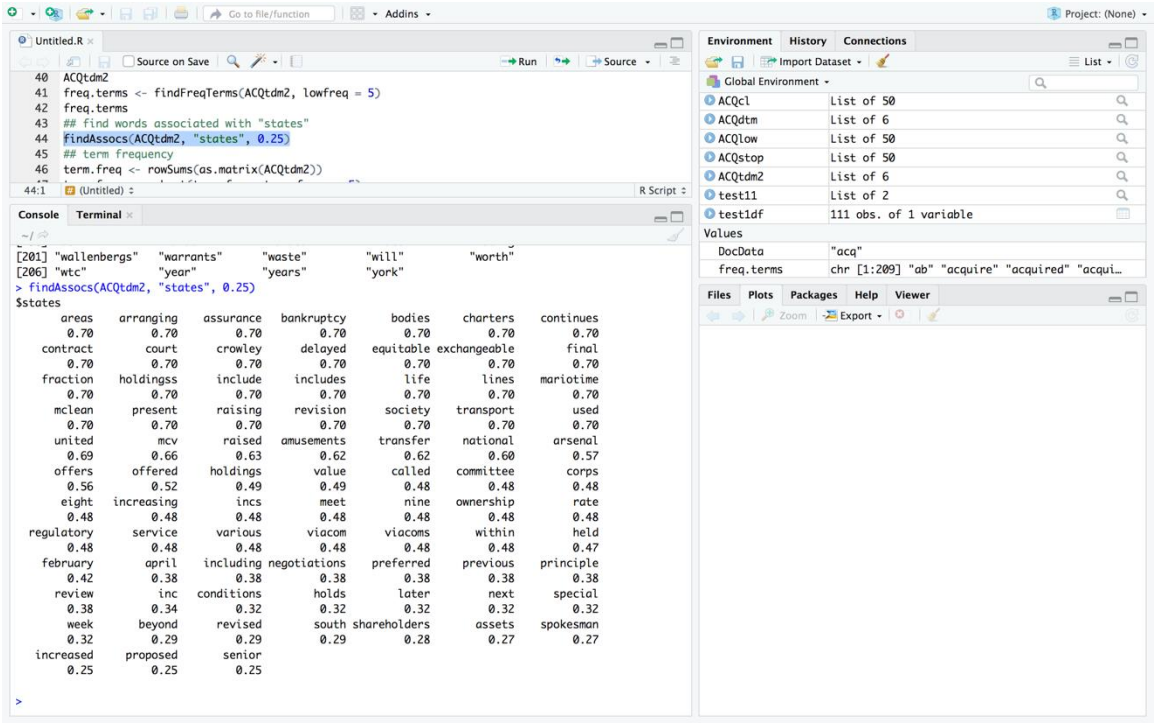
9. remove stop words from the corpus



10. find terms with a frequency of 5 or more



11. find words associated with "states"

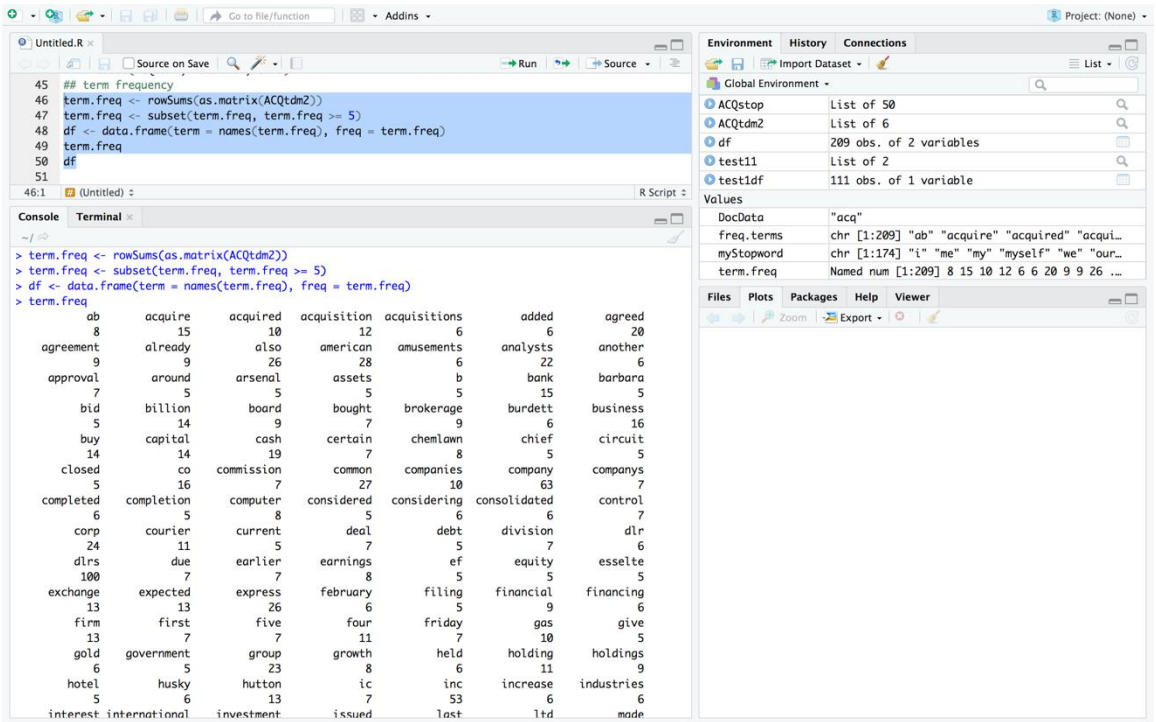


```
ACQtdm2
freq.terms <- findFreqTerms(ACQtdm2, lowfreq = 5)
freq.terms
## find words associated with "states"
findAssocs(ACQtdm2, "states", 0.25)
## term frequency
term.freq <- rowSums(as.matrix(ACQtdm2))
```

```
[201] "wallenbergs"   "warrants"      "waste"         "will"          "worth"
[206] "wtc"           "year"          "years"         "york"
> findAssocs(ACQtdm2, "states", 0.25)
$states
      areas    arranging    assurance   bankruptcy       bodies     charters    continues
       0.70         0.70         0.70         0.70         0.70         0.70         0.70
   contract        court      crowley      delayed    equitable exchangeable        final
       0.70         0.70         0.70         0.70         0.70         0.70         0.70
   fraction    holdingss      include     includes         life        lines    mariotime
       0.70         0.70         0.70         0.70         0.70         0.70         0.70
     mclean      present      raising     revision      society    transport         used
       0.70         0.70         0.70         0.70         0.70         0.70         0.70
     united          mcv       raised   amusements     transfer     national     arsenal
       0.69         0.66         0.63         0.62         0.62         0.60         0.57
     offers      offered     holdings        value       called    committee        corps
       0.56         0.52         0.49         0.49         0.48         0.48         0.48
      eight   increasing         incs         meet         nine    ownership         rate
       0.48         0.48         0.48         0.48         0.48         0.48         0.48
  regulatory      service      various       viacom      viacoms       within         held
       0.48         0.48         0.48         0.48         0.48         0.48         0.47
   february        april   including negotiations    preferred     previous    principle
       0.42         0.38         0.38         0.38         0.38         0.38         0.38
     review          inc   conditions        holds        later         next      special
       0.38         0.34         0.32         0.32         0.32         0.32         0.32
       week       beyond      revised        south shareholders       assets    spokesman
       0.32         0.29         0.29         0.29         0.28         0.27         0.27
  increased     proposed       senior
       0.25         0.25         0.25

>
```

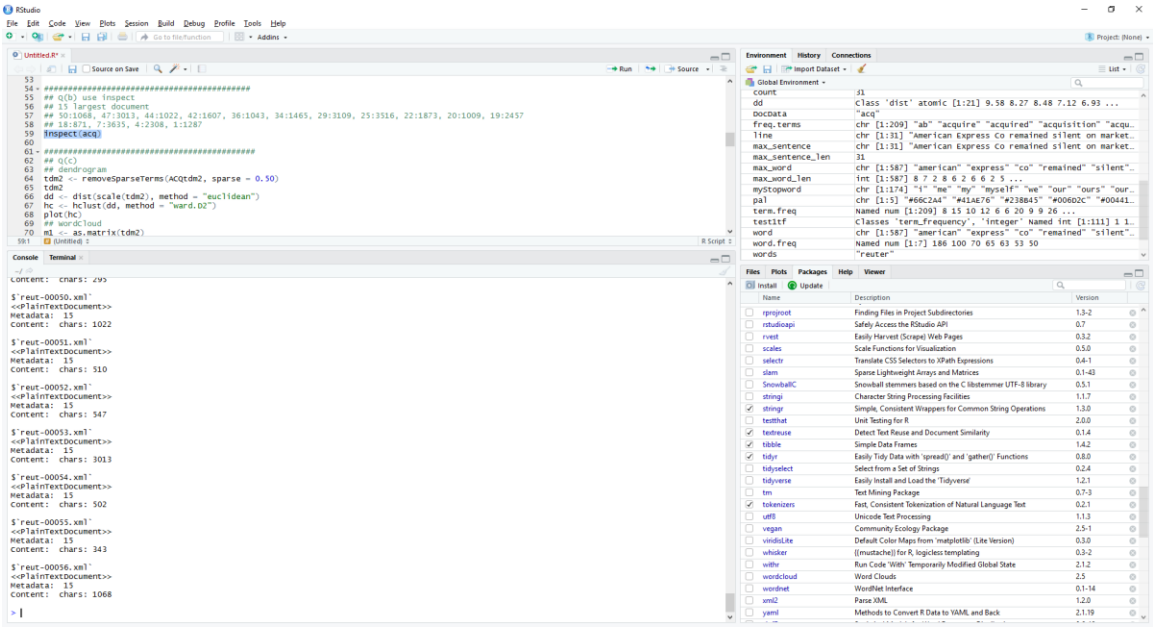
12. term frequency



```
## term frequency
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
```

```
> 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
          ab      acquire     acquired   acquisition  acquisitions         added        agreed
           8           15           10           12            6            6           20
   agreement      already         also     american    amusements     analysts      another
           9            9           26           28            6           22            6
    approval       around      arsenal       assets            b         bank      barbara
           7            5            5            5            5           15            5
         bid      billion        board       bought     brokerage      burdett     business
           5           14            9            7            9            6           16
         buy      capital         cash      certain     chemlawn        chief      circuit
          14           14           19            7            8            5            5
      closed           co   commission       common    companies      company     companys
           5           16            7           27           10           63            7
   completed   completion     computer   considered  considering consolidated      control
           6            5            8            5            6            6            7
        corp      courier      current         deal         debt     division          dlr
          24           11            5            7            5            7            6
        dlrs          due      earlier     earnings           ef       equity     esselte
         100            7            7            8            5            5            5
    exchange     expected      express     february       filing    financial    financing
          13           13           26            6            5            9            6
        firm        first         five         four       friday          gas         give
          13            7            7           11            7           10            5
        gold   government        group       growth         held      holding     holdings
           6            5           23            8            6           11            9
       hotel        husky       hutton           ic          inc     increase   industries
           5            6           13            7           53            6            6
  interest international   investment       issued         last          ltd         made
```

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



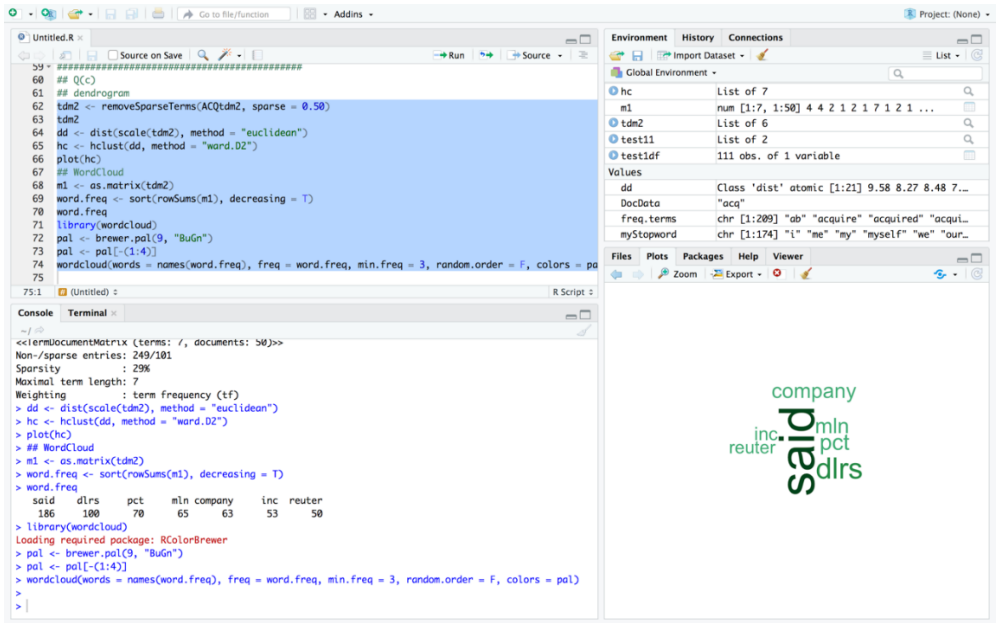
15 largest document

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

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

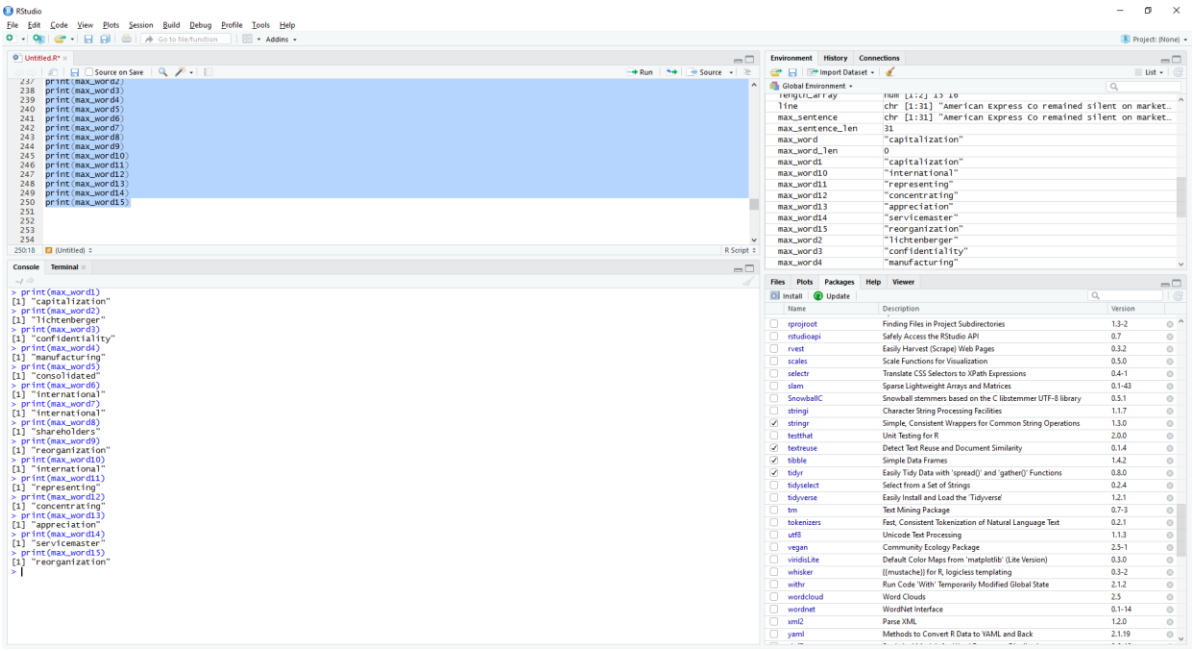
For the following you will need to write R functions to help you compute the results.

Use the packages textreuse, wordnet, zipfR

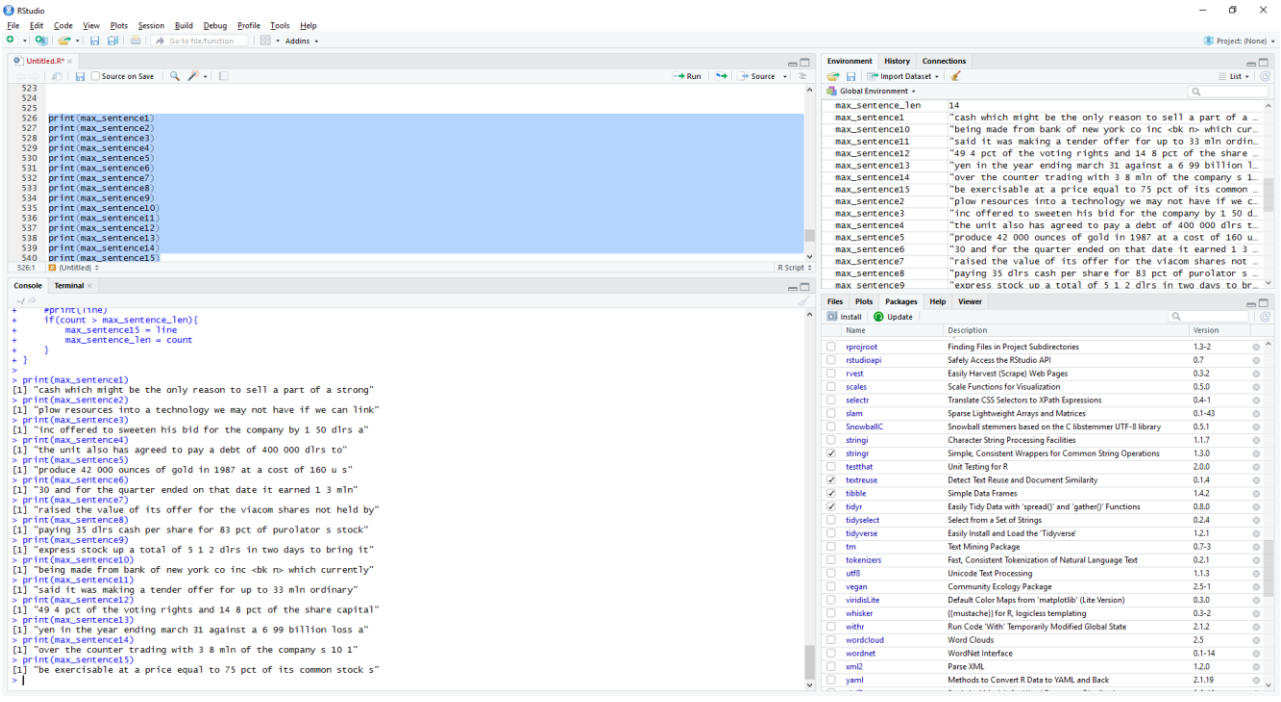


c. Prior to removing the punctuation, find the longest word and longest sentence in each document from the 15 largest documents.

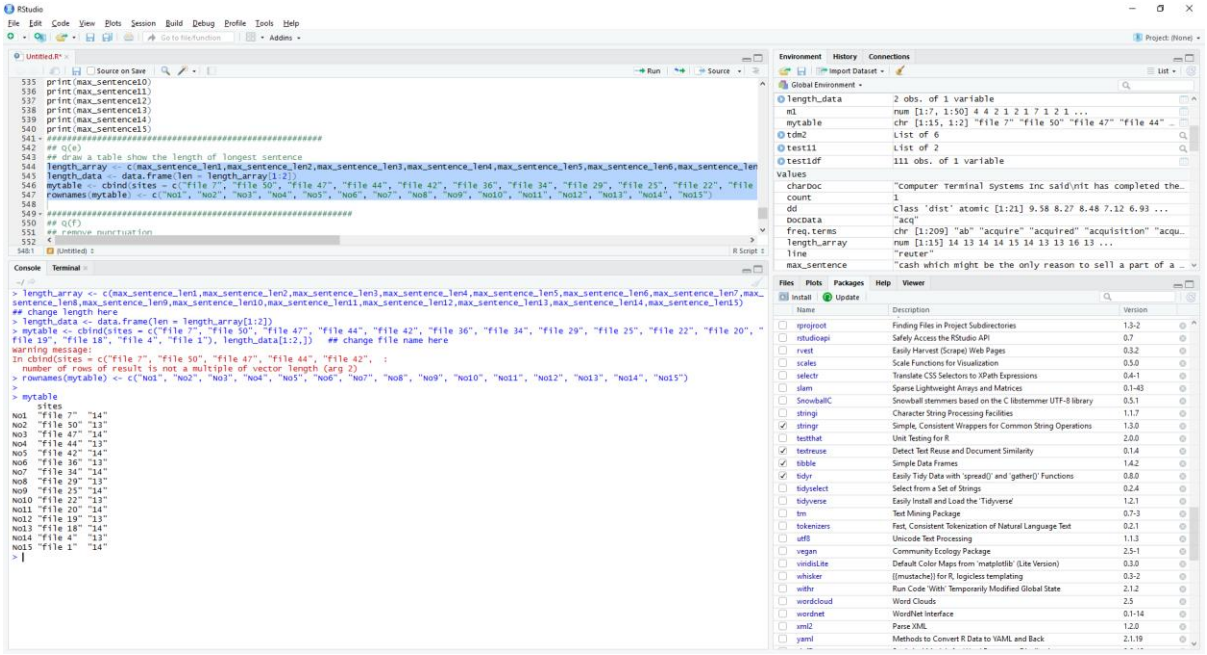
Longest word



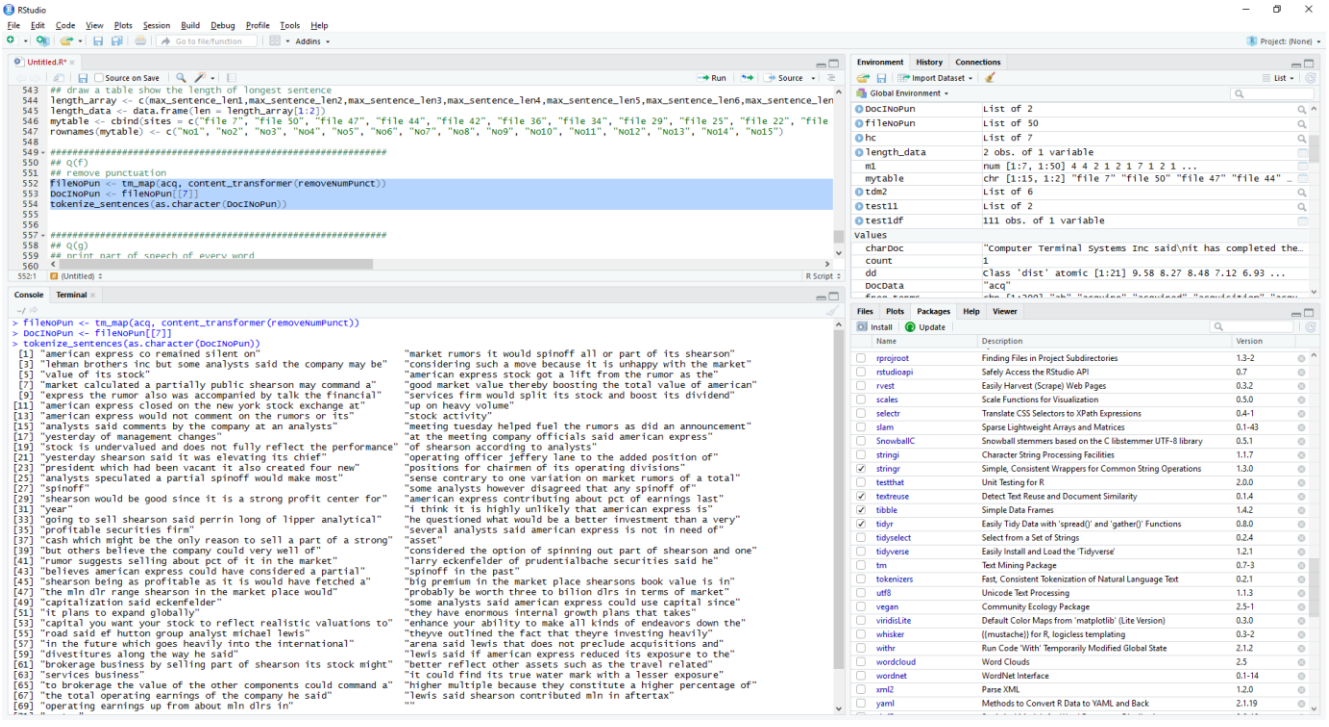
Longest sentence



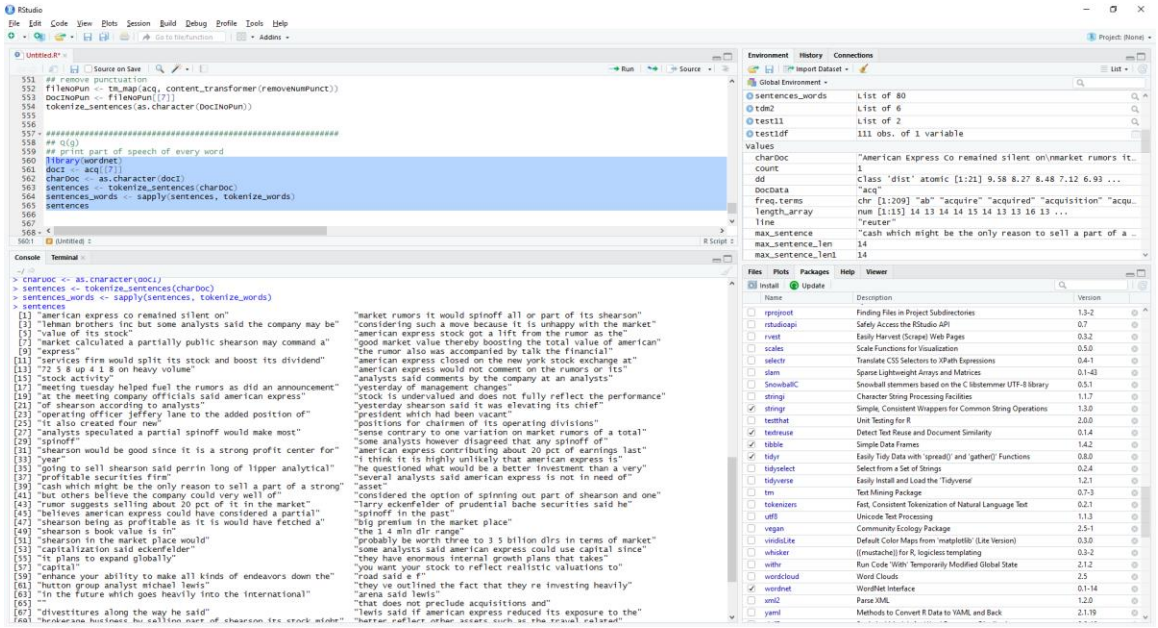
d. Print a table of the length of each sentence in each of the 15 documents.



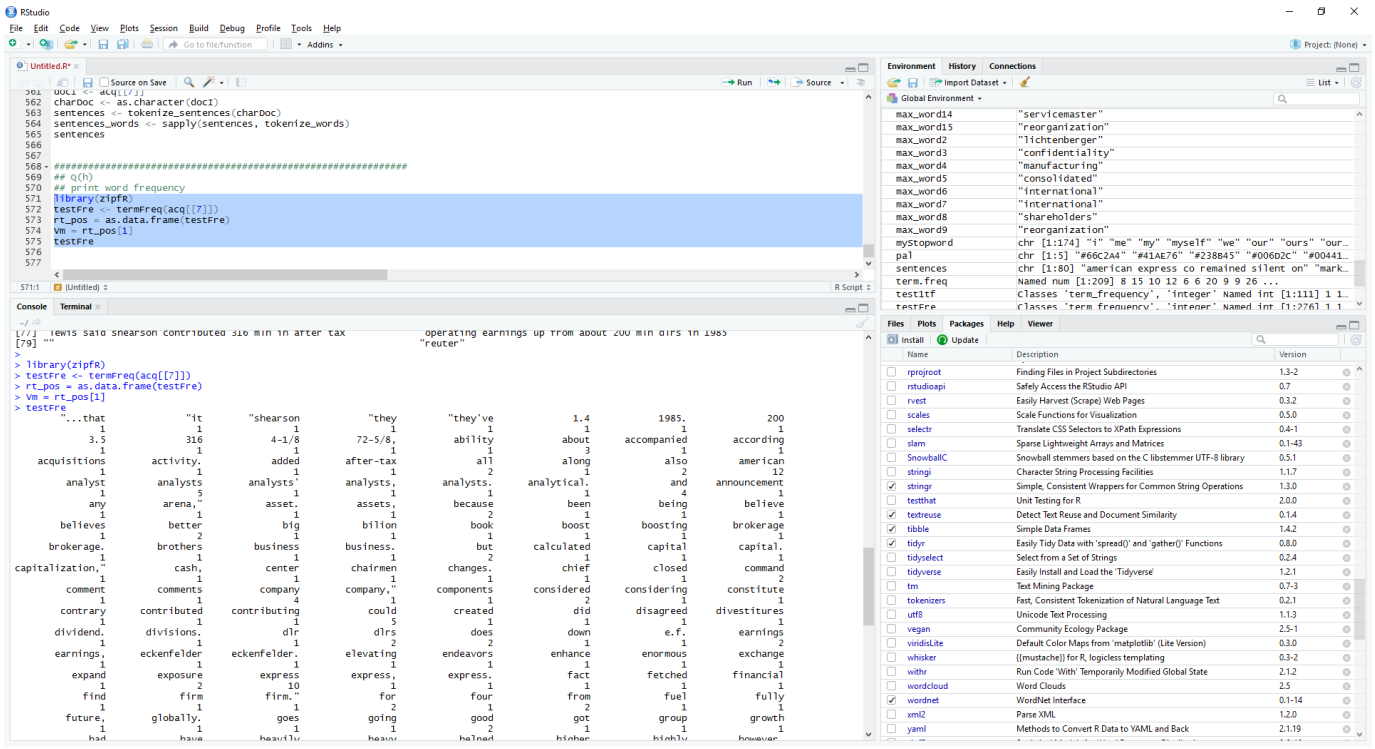
f. For each sentence of the longest document, remove the punctuation. Display the sentences in the largest document (just show results/screenshot for the longest document).



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).

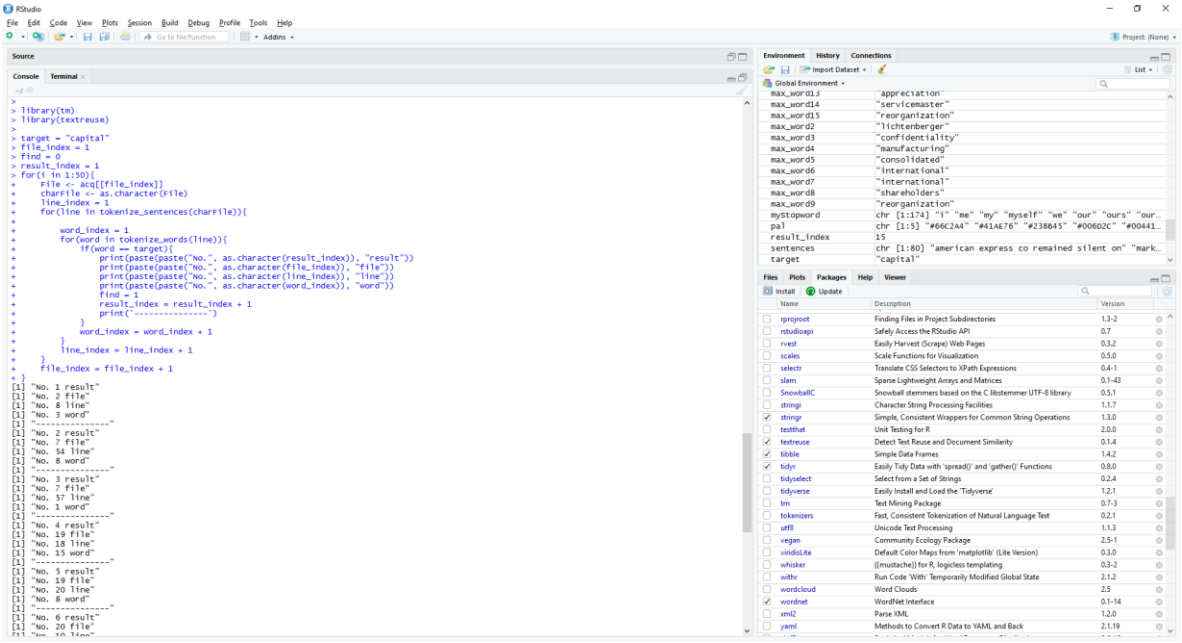


h. Analyze word frequency using functions from package zipfR in the 15 largest documents (just show results/screenshot for the longest document).



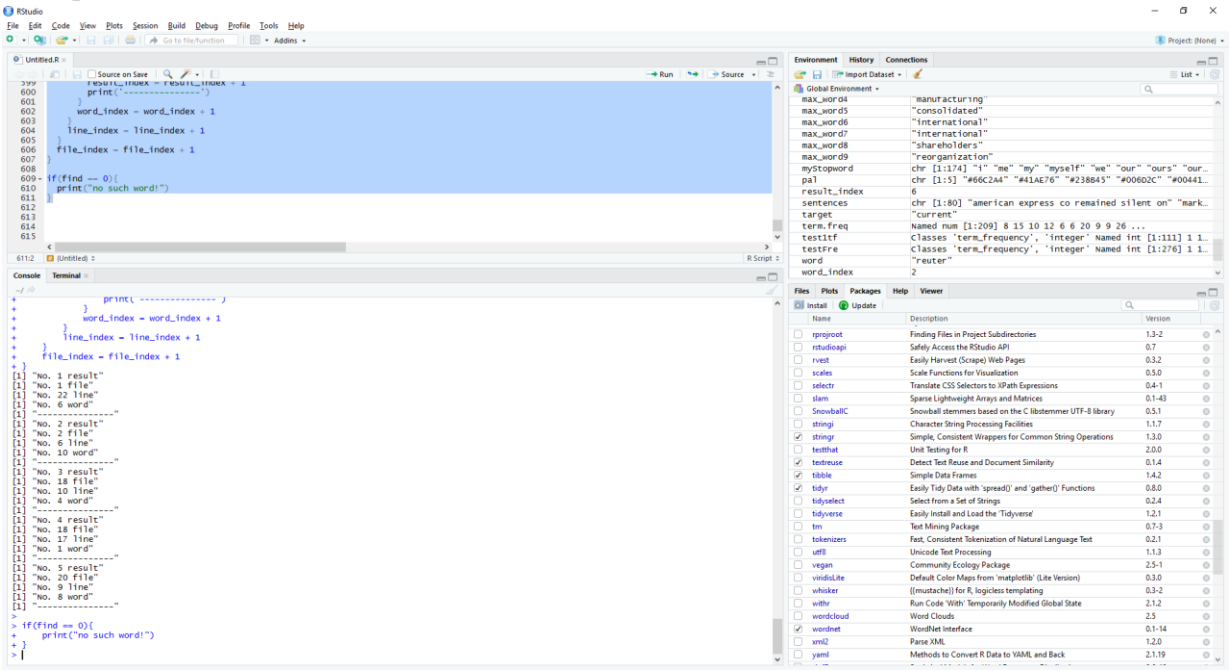
b. Write an R function to search through the documents to find a specific word or phrase. Print the document number, line number, and word index in the sentence. Demonstrate with three examples. Use words of 6 characters or more as your test cases. 3 points.

Example1:



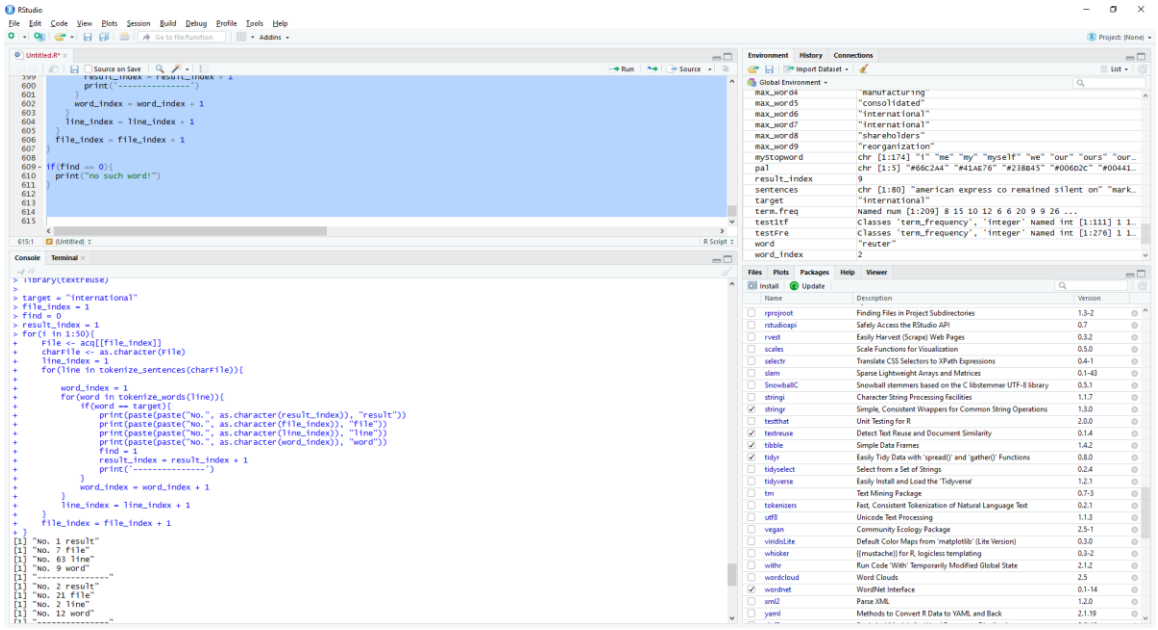
```
> library(tm)
> library(textreuse)
>
> target = "capital"
> file_index = 1
> find = 0
> result_index = 1
> for(i in 1:50){
+     file <- acq[[file_index]]
+     charFile <- as.character(file)
+     line_index = 1
+     for(line in tokenize_sentences(charFile)){
+
+         word_index = 1
+         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
+ }
[1] "No. 1 result"
[1] "No. 2 file"
[1] "No. 8 line"
[1] "No. 3 word"
[1] "---------------"
[1] "No. 2 result"
[1] "No. 7 file"
[1] "No. 54 line"
[1] "No. 8 word"
[1] "---------------"
[1] "No. 3 result"
[1] "No. 7 file"
[1] "No. 57 line"
[1] "No. 1 word"
[1] "---------------"
[1] "No. 4 result"
[1] "No. 19 file"
[1] "No. 18 line"
[1] "No. 15 word"
[1] "---------------"
[1] "No. 5 result"
[1] "No. 19 file"
[1] "No. 20 line"
[1] "No. 8 word"
[1] "---------------"
[1] "No. 6 result"
[1] "No. 20 file"
```

Example 2:



```
+             print( '---------------' )
+         }
+         word_index = word_index + 1
+     }
+     line_index = line_index + 1
+ }
+ file_index = file_index + 1
+ }
[1] "No. 1 result"
[1] "No. 1 file"
[1] "No. 22 line"
[1] "No. 6 word"
[1] "---------------"
[1] "No. 2 result"
[1] "No. 2 file"
[1] "No. 6 line"
[1] "No. 10 word"
[1] "---------------"
[1] "No. 3 result"
[1] "No. 18 file"
[1] "No. 10 line"
[1] "No. 4 word"
[1] "---------------"
[1] "No. 4 result"
[1] "No. 18 file"
[1] "No. 17 line"
[1] "No. 1 word"
[1] "---------------"
[1] "No. 5 result"
[1] "No. 20 file"
[1] "No. 9 line"
[1] "No. 8 word"
[1] "---------------"
>
> if(find == 0){
+     print("no such word!")
+ }
>
```

Example 3:



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.