Class Project #3:
Text Analytics in R

Functions used in the project:

Functions
setwd()
getwd()
Vcorpus()
inspect()
str()
paste()
library()
DocumentTermMatrix()
TermDocumentMatrix()
termFreq()
as.data.frame()
tm_map()
content_transformer()
findFreqTerms()
rowSums()
subset()
removeSparseTerms()
hclust()

plot()
as.matrix()
sort()
pal()
brewer.pal()
wordcloud()
colSums()
length()
order()
freq[()]
text$content[1]
tokens()
docfreq()
dfm_weight()
dfm_tfidf()
Text[[]]
Text1[[1]]$content
head()
set.seed()
noquote()
data.frame()
dist()

na.omit()
options()
a[which.max(a$char),]
summary()
N()
V()
Vm()
read.tfl()
with()
tfl2spc()
scan()
Corpus()
tm_term_score()
as.character()
strsplit()
vapply()
names()
textcnt()
readLines()
paste()
strsplit()

gsub()
length()
sapply()
unique()
print()
data.frame()
create_tcorpus()
anti_join()
tibble()
unnest_tokens()
stri_replace_all()
stri_dfm()
stri_trans_tolower()

Part a: All Functions from Lecture 8

1. Create Vcorpus: Function to create a VCorpus object.

```
setwd("C://Users//aneri//OneDrive//Documents//RProject3//Projec3//Text Document")
getwd()
Text <- VCorpus(DirSource(".", ignore.case=TRUE,mode="text"))
```

```
> setwd("C://Users//aneri//OneDrive//Documents//RProject3//Projec3//Text Document")
> getwd()
[1] "C:/Users/aneri/OneDrive/Documents/RProject3/Projec3/Text Document"
> Text <- VCorpus(DirSource(".", ignore.case=TRUE,mode="text"))
>
```

2. Text and inspect(Text)

```
> Text <- VCorpus(DirSource(".", ignore.case=TRUE,mode="text"))
> Text
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 11
```

```
> inspect(Text)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 11

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 12773

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 37182

[[3]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 15768

[[4]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 8410

[[5]]
<<PlainTextDocument>>
```

```
  ..$ content: chr [1:272] "DR. LANYONâ\200\231S NARRATIVE" "" "On the ninth of January, now
 four days ago, I received by the evening" "delivery a registered envelope, addressed in the
 hand of my colleague" ...
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2019-04-22 00:23:07"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "Chapter 9.txt"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ DrJekyllAndMrHyde.txt:List of 2
  ..$ content: chr [1:2559] "ï»¿THE STRANGE CASE OF DR. JEKYLL AND MR. HYDE ***" "" "" "Prod
uced by David Widger" ...
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2019-04-22 00:23:07"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "DrJekyllAndMrHyde.txt"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 - attr(*, "class")= chr [1:2] "VCorpus" "Corpus"
```

3. Extracting specific document from the list of documents

```
Chapter1 <- Text[[1]]
Chapter2 <- Text[[2]]
Chapter3 <- Text[[3]]
Chapter4 <- Text[[4]]
Chapter5 <- Text[[5]]
Chapter6 <- Text[[6]]
Chapter7 <- Text[[7]]
Chapter8 <- Text[[8]]
Chapter9 <- Text[[9]]
Chapter10 <- Text[[10]]
```

```
> Chapter1 <- Text[[1]]
> Chapter1
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 12773
```

4. DocumentTermMartix(): Function to create document term matrix

Texttdm <- TermDocumentMatrix(Text)

```
> Textdtm <- DocumentTermMatrix(Text)
> Textdtm
<<DocumentTermMatrix (documents: 11, terms: 6198)>>
Non-/sparse entries: 15179/52999
Sparsity           : 78%
Maximal term length: 31
Weighting          : term frequency (tf)
```

```
> inspect(Texttdm[1:50,1:10])
<<TermDocumentMatrix (terms: 50, documents: 10)>>
Non-/sparse entries: 45/455
Sparsity           : 91%
Maximal term length: 57
Weighting          : term frequency (tf)
Sample             :
```

5. Document Term Frequency: Function to count the number of occurrences of words in the documents.

```
Chapter1tf <- termFreq(Chapter1)
Chapter2tf <- termFreq(Chapter2)
Chapter3tf <- termFreq(Chapter3)
Chapter4tf <- termFreq(Chapter4)
Chapter5tf <- termFreq(Chapter5)
Chapter6tf <- termFreq(Chapter6)
Chapter7tf <- termFreq(Chapter7)
Chapter8tf <- termFreq(Chapter8)
Chapter9tf <- termFreq(Chapter9)
Chapter10tf <- termFreq(Chapter10)
```

```
> Chapter1tf <- termFreq(Chapter1)
> Chapter1tf
               (the                (what          "frightened                 â€˜i
                 1                    1                    1                    2
             â€˜if              â€˜name               â€˜set                 â€œa
                 1                    1                    1                    2
            â€œand               â€œbut               â€œdid â€œenfield,â€\u009d
                 3                    4                    1                    1
             â€œhe              â€œhere       â€œhm,â€\u009d                 â€œi
                 1                    1                    1                   10
 â€œindeed?â€\u009d                â€œit                â€œmy               â€œno,
                 1                    2                    1                    1
        â€œthatâ€™s         â€œthereâ€™s â€œtut-tut!â€\u009d            â€œwell,
                 1                    1                    1                    1
   â€œwell,â€\u009d              â€œwhat              â€œwith             â€œyes,
                 1                    1                    2                    3
            â€œyou                 able                about             abreast
                 1                    1                    9                    1
            accent               accept         accident,â€™           according
                 1                    1                    1                    1
       acquaintance                 acts                added               added,
```

```
          whatâ€™s                when                where               which
               1                   7                    4                   5
           while              while,               white                 who
               1                   1                    1                   6
           whole                whom                why.                wild
               1                   2                    1                   1
            will             window,             windows                wine
               1                   1                    2                   1
          winter              wishes                with               with,
               1                   1                   20                   1
           women          wondering,               word;               words
               1                   1                    1                   1
          world,              worse)              worse,               would
               1                   1                    1                   6
           wrong              years.                 yet                 you
               1                   1                    4                  18
           young                your      yours.â€\u009d              youth.
               1                   4                    1                   1
attr(,"class")
[1] "term_frequency" "integer"
```

6. Data frame: Function to convert the term frequency result to data frame.

```
Chapter1df <- as.data.frame(Chapter1tf)
Chapter2df <- as.data.frame(Chapter2tf)
Chapter3df <- as.data.frame(Chapter3tf)
Chapter4df <- as.data.frame(Chapter4tf)
Chapter5df <- as.data.frame(Chapter5tf)
Chapter6df <- as.data.frame(Chapter6tf)
Chapter7df <- as.data.frame(Chapter7tf)
Chapter8df <- as.data.frame(Chapter8tf)
Chapter9df <- as.data.frame(Chapter9tf)
Chapter10df <- as.data.frame(Chapter10tf)
```

```
below;                  1
best                    1
best,                   1
best.                   1
better                  2
bird                    1
bit                     1
black                   3
blackmail,              1
bland                   1
blind                   1
blistered               1
block                   1
blood                   1
body                    2
bond                    1
bore                    1
brasses,                1
breakfasted,            1
broken                  1
brother                 1
```

7. Convert to lower class, remove numbers and punctuations: Function to convert the documents text into lower class, remove numbers and punctuations.

```
> Textlow <- tm_map(Text, content_transformer(tolower))
> RemoveNumPunc <- function(x)
+ gsub("[^[:alpha:][:space:]]*", "", x)
> Textcl <- tm_map(Textlow,content_transformer(RemoveNumPunc))
> Textcl
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 11
```

```
> Textcl
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 11
```

8. Stop Words: Function to check the stop words.

```
> StopWords
  [1] "i"          "me"         "my"         "myself"     "we"         "our"
  [7] "ours"       "ourselves"  "you"        "your"       "yours"      "yourself"
 [13] "yourselves" "he"         "him"        "his"        "himself"    "she"
 [19] "her"        "hers"       "herself"    "it"         "its"        "itself"
 [25] "they"       "them"       "their"      "theirs"     "themselves" "what"
 [31] "which"      "who"        "whom"       "this"       "that"       "these"
 [37] "those"      "am"         "is"         "are"        "was"        "were"
 [43] "be"         "been"       "being"      "have"       "has"        "had"
 [49] "having"     "do"         "does"       "did"        "doing"      "would"
 [55] "should"     "could"      "ought"      "i'm"        "you're"     "he's"
 [61] "she's"      "it's"       "we're"      "they're"    "i've"       "you've"
 [67] "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"
 [79] "we'll"      "they'll"    "isn't"      "aren't"     "wasn't"     "weren't"
 [85] "hasn't"     "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"
 [97] "couldn't"   "mustn't"    "let's"      "that's"     "who's"      "what's"
[103] "here's"     "there's"    "when's"     "where's"    "why's"      "how's"
[109] "a"          "an"         "the"        "and"        "but"        "if"
[115] "or"         "because"    "as"         "until"      "while"      "of"
```

```
 [49] "having"     "do"         "does"       "did"        "doing"      "would"
 [55] "should"     "could"      "ought"      "i'm"        "you're"     "he's"
 [61] "she's"      "it's"       "we're"      "they're"    "i've"       "you've"
 [67] "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"
 [79] "we'll"      "they'll"    "isn't"      "aren't"     "wasn't"     "weren't"
 [85] "hasn't"     "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"
 [97] "couldn't"   "mustn't"    "let's"      "that's"     "who's"      "what's"
[103] "here's"     "there's"    "when's"     "where's"    "why's"      "how's"
[109] "a"          "an"         "the"        "and"        "but"        "if"
[115] "or"         "because"    "as"         "until"      "while"      "of"
[121] "at"         "by"         "for"        "with"       "about"      "against"
[127] "between"    "into"       "through"    "during"     "before"     "after"
[133] "above"      "below"      "to"         "from"       "up"         "down"
[139] "in"         "out"        "on"         "off"        "over"       "under"
[145] "again"      "further"    "then"       "once"       "here"       "there"
[151] "when"       "where"      "why"        "how"        "all"        "any"
[157] "both"       "each"       "few"        "more"       "most"       "other"
[163] "some"       "such"       "no"         "nor"        "not"        "only"
[169] "own"        "same"       "so"         "than"       "too"        "very"
```

9. Inspect Stop words: Function to inspect documents after the removal of stop words.

```
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 10

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 8733

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 25862

[[3]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 11047

[[4]]
<<PlainTextDocument>>
Metadata:  7
```

```
[[5]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 6205

[[6]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 6009

[[7]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 5400
```

```
[[7]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 5400

[[8]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 1898

[[9]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 15951

[[10]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 10161
```

10. Term Document Matrix: Function to computer document term matrix without stop words

```
> Textdm2 <- TermDocumentMatrix(TextStop,control= list(wordlengths=c(1,Inf)))
> Textdm2
<<TermDocumentMatrix (terms: 4475, documents: 11)>>
Non-/sparse entries: 11408/37817
Sparsity           : 77%
Maximal term length: 21
Weighting          : term frequency (tf)
```

11. Finding frequency terms: Function to find frequent words from the documents.

FreqTerms <- findFreqTerms(Textdm2,lowfreq = 4)

```
[1549] "vain"          "vast"          "venture"       "victim"
[1553] "view"          "views"         "virtue"        "visibly"
[1557] "visit"         "visitor"       "voice"         "voluntary"
[1561] "wait"          "walk"          "walked"        "wall"
[1565] "want"          "wanting"       "warned"        "watch"
[1569] "watched"       "way"           "ways"          "weakness"
[1573] "wear"          "week"          "weeping"       "weight"
[1577] "welcome"       "welcomed"      "well"          "went"
[1581] "wept"          "whatever"      "whether"       "whipped"
[1585] "whispered"     "white"         "whole"         "wholesale"
[1589] "wholly"        "whose"         "wicked"        "wild"
[1593] "will"          "wind"          "window"        "windows"
[1597] "wine"          "winter"        "wise"          "wish"
[1601] "withdrawn"     "within"        "without"       "woke"
[1605] "woman"         "women"         "wonder"        "wonderful"
[1609] "wood"          "word"          "worded"        "words"
[1613] "wore"          "work"          "world"         "worse"
[1617] "worst"         "write"         "writing"       "written"
[1621] "wrong"         "wrote"         "yard"          "year"
[1625] "years"         "yes"           "yet"           "youãââ"
[1629] "young"         "younger"       "youth"
```

12. Term Frequency: Function to find how important a word is in a document.

```
> TermFrequency <- rowSums(as.matrix(Textdm2))
> TermFrequencySub <- subset(TermFrequency,TermFrequency >= 6)
> TermFrequencydf <- as.data.frame(names(TermFrequency),freq=TermFrequency)
>
```

```
        wouldnât            wrack          wrappers           wreaths
               1                2                 2                 2
           wreck          wrecked         wrestling             write
               2                2                 2                 9
       writeãââ           writer        writeãââ       writeãââs
               1                1                 1                 2
       writeâs          writing          written             wrong
               2                6                10                10
       wrongãââ            wrote            wrung              yard
               2                4                 2                 9
      yardãââ             year            years            yellow
               1               10                20                 2
           yes        yesterday              yet           yetãââ
            24                2                61                 1
          youâ          youãââ        youãââll           youâll
               1               10                 1                 2
         young          younger       youngerãââ           yoursâ
            12                5                 1                 1
      yoursãââ       yourselfãââ             youth
               2                3                 4
```

13. Sparseset: Function to compute the sparseset.
Sparsetdm2 <- removeSparseTerms(Textdtm2,sparse=0.75)

```
> sparsetdm2
<<TermDocumentMatrix (terms: 1365, documents: 11)>>
Non-/sparse entries: 5909/9106
Sparsity           : 61%
Maximal term length: 14
Weighting          : term frequency (tf)
```

```
  jekyll               0            28            12            12             2
  lawyer               5             0            16             6             2
  man                 19            13             9            12             4
  now                  0            21            10             2             1
  one                 14            17             5            12             4
  said                15             3            16            18             8
  upon                 2            16             3             6             7
  utterson            11             0            21            26             7
  will                 1            12            14            12             3
          Docs
Terms      Chapter 5.txt Chapter 6.txt Chapter 8.txt Chapter 9.txt DrJekyllAndMrHyde.txt
  hyde                 3             4             5             1                    87
  jekyll               6             7            10             5                    70
  lawyer               9             3            27             0                    67
  man                  3             3            13             7                    74
  now                  3             5            16             6                    55
  one                  7             6             9             9                    75
  said                14             5            47             7                   129
  upon                 3             4            11            10                    60
  utterson            15            15            36             1                   116
  will                 3             4             4            17                    61
```

14. Frequency Analysis: Function for frequency analysis by creating adocument term matrix and find column sums and means for the number arrays.

```
> Texdtm <- DocumentTermMatrix(TextStop)
> freq <- colSums(as.matrix(Texdtm))
> Texdtm
<<DocumentTermMatrix (documents: 11, terms: 4475)>>
Non-/sparse entries: 11408/37817
Sparsity           : 77%
Maximal term length: 21
Weighting          : term frequency (tf)
> length(freq)
[1] 4475
```

15. Frequency Order: Function to order frequency in decreasing order.

```
> ord <- order(freq,decreasing = TRUE)
> freq[head(ord)]
    said utterson      hyde       one       man    jekyll
     268      256       184       160       157       156
```

16. Enforcing Word lengths: Function to enforce the word lengths,

```
> Textdtmr <- DocumentTermMatrix(TextStop,control=list(wordLengths=c(4,20)))
> Textdtmr
<<DocumentTermMatrix (documents: 11, terms: 4343)>>
Non-/sparse entries: 10866/36907
Sparsity           : 77%
Maximal term length: 18
Weighting          : term frequency (tf)
>
```

17. Frequency order after word lengths: Function to order frequency in decreasing order after enforcing word lengths.

```
> freqr <- colSums(as.matrix(Textdtmr))
> order <- order(freqr,decreasing = TRUE)
> freq[head(order)]
  repulsion    trembles        hold inseparable     killing visitorãâãâs
          1           2           5           2           2           1
>
```

18. Quanteda package extracting documents: Function from quanteda package to extract the documents.

Text1 <- Text$content[1]
text <- Text1[[1]]$content

```
> text <- Text1[[1]]$content
> text
  [1] "STORY OF THE DOOR"
  [2] ""
  [3] "Mr. Utterson the lawyer was a man of a rugged countenance that was"
  [4] "never lighted by a smile; cold, scanty and embarrassed in discourse;"
  [5] "backward in sentiment; lean, long, dusty, dreary and yet somehow"
  [6] "lovable. At friendly meetings, and when the wine was to his taste,"
  [7] "something eminently human beaconed from his eye; something indeed which"
  [8] "never found its way into his talk, but which spoke not only in these"
  [9] "silent symbols of the after-dinner face, but more often and loudly in"
 [10] "the acts of his life. He was austere with himself; drank gin when he"
 [11] "was alone, to mortify a taste for vintages; and though he enjoyed the"
 [12] "theatre, had not crossed the doors of one for twenty years. But he had"
 [13] "an approved tolerance for others; sometimes wondering, almost with"
 [14] "envy, at the high pressure of spirits involved in their misdeeds; and"
 [15] "in any extremity inclined to help rather than to reprove. â€œI incline to"
 [16] "Cainâ€™s heresy,â€\u009d he used to say quaintly: â€œI let my brother go to the"
 [17] "devil in his own way.â€\u009d In this character, it was frequently his fortune"
```

19. Extract the tokens: Function to extract the tokens from the chapters.
texttokens <- tokens(text)

```
text40 :
 [1] "It"      "chanced" "on"      "one"     "of"      "these"   "rambles" "that"    "their"   "way"
[11] "led"     "them"    "down"    "a"

text41 :
 [1] "by-street" "in"        "a"         "busy"      "quarter"   "of"        "London"    "."
 [9] "The"       "street"    "was"       "small"     "and"       "what"      "is"

text42 :
 [1] "called"   "quiet"    ","        "but"      "it"       "drove"    "a"        "thriving" "trade"
[10] "on"       "the"      "weekdays" "."        "The"

text43 :
 [1] "inhabitants" "were"        "all"         "doing"       "well"        ","           "it"
 [8] "seemed"      "and"         "all"         "emulously"   "hoping"      "to"
```

20. Apply dfm: Function to create dfm object for the chapters.

```
> dfmtext1 <- dfm(text)
> dfmtext1
Document-feature matrix of: 225 documents, 854 features (98.6% sparse).
> |
```

21. Sentiment Analysis: Function to compute all the list words and their emotions.

```
> textstn <- syuzhet::get_nrc_sentiment(text)
> textstn
    anger anticipation disgust fear joy sadness surprise trust negative positive
1       0            0       0    0   0       0        0     0        0        0
2       0            0       0    0   0       0        0     0        0        0
3       1            0       1    1   0       0        0     0        2        0
4       0            0       0    0   1       0        1     1        2        1
5       0            1       0    0   0       1        0     0        2        0
6       0            1       0    0   2       0        0     2        0        2
7       0            0       0    0   0       0        0     0        0        1
8       0            0       0    0   1       0        0     1        1        2
9       0            0       0    0   0       0        0     0        0        1
10      0            0       0    1   0       1        0     0        1        0
11      0            0       0    0   0       0        0     0        0        0
12      0            0       0    0   0       0        0     0        0        0
13      0            0       0    0   0       0        0     0        0        0
14      0            1       0    0   1       0        1     0        1        1
```

22. rowSums and colSums in sentiment analysis: Function to compute the rowSums and colSums.

```
textrdt <- rowSums(textstn)
textrdt
 [1]  0  0  5  6  4  7  1  5  1  3  0  0  0  5  2  3 11  9  1  2  0  8  3  7  7  4  0  6  2  1  1  0  6  6  2  0  7  0  0  0  1  6  0  0  0  4  0  0  1  4  3  0  0  7  6  2  2  7
[59]  0  0  0  0  0  0  0  4  2  0  0  3  1  0  0  1  0  0  1  3  0  0  4  4  1  3  5  4  2  4  7  6  1  2  7  5  1  6  3  6  0  4  3  7  0  2  7 10  4  1  4  2  7  1  2  8  5  2
117]  9  4  3  0  1  6  0  2  1  0  1  8  7  2  0  0  1  6  9  1  1  2  2  0  0  0  5  2  0  6 15 10  0  0  4  0  1  2  0  1  0  0  0  0  3  0  5  5  2  2  5  0  0 11  0  0  2  2
175]  5  3  0  0  3  0  0  0  7  0  0  0  4  3  0  2  0  0  0  0  1  0  6 17  1  2  0  0  0  0  9  0  0  4  0  1  0  1  0  0  3  0  2  0  0  6  3  6  0  0  4
textcdt <- colSums(textstn)
textcdt
     anger anticipation      disgust         fear          joy      sadness     surprise        trust     negative     positive
        37           59           28           56           46           38           28           77           84          112
```

23. Document frequency of words: computing the frequency of words in the chapters.

```
> textfreq <- docfreq(dfmtext1)
> textfreq
          story              of             the            door              mr
              3              71             111               9              17
              .        utterson          lawyer             was               a
             98              11               5              41              67
            man          rugged     countenance            that           never
             19               1               1              29               8
        lighted              by           smile               ;            cold
              2               5               1              48               1
              ,          scanty             and     embarrassed              in
            115               1              77               1              38
      discourse        backward       sentiment            lean            long
              1               1               1               1               4
          dusty          dreary             yet         somehow         lovable
              1               1               4               1               1
             at        friendly        meetings            when            wine
             16               2               1               7               1
             to             his           taste       something       eminently
             43              21               2               4               1
          human        beaconed            from             eye          indeed
              1               1               7               2               1
          which           found             its             way            into
              5               1               5               7               6
           talk             but           spoke             not            only
              1              22               1              11               4
          these          silent         symbols    after-dinner            face
              6               1               1               1               1
           more           often          loudly            acts            life
              6               2               1               1               2
             he         austere            with         himself           drank
             32               1              21               2               1
            gin           alone        mortify             for        vintages
              1               1               1              23               1
         though         enjoyed         theatre             had         crossed
              3               1               1              15               1
          doors             one          twenty           years              an
              2              14               1               1               5
       approved       tolerance          others       sometimes       wondering
              1               1               2               1               1
         almost             envy            high        pressure         spirits
              1               1               2               1               1
       involved           their        misdeeds             any       extremity
              1               6               1               3               1
```

24. Text weighting: Function to show what kind of structure the chapters have.

```
> textweights <- dfm_weight(dfmtext1,scheme="prop")
> textweights
Document-feature matrix of: 225 documents, 854 features (98.6% sparse).
> str(textweights)
Formal class 'dfm' [package "quanteda"] with 15 slots
  ..@ settings    : list()
  ..@ weightTf    :List of 3
  .. ..$ scheme: chr "prop"
  .. ..$ base  : NULL
  .. ..$ K     : NULL
  ..@ weightDf    :List of 5
  .. ..$ scheme   : chr "unary"
  .. ..$ base     : NULL
  .. ..$ c        : NULL
  .. ..$ smoothing: NULL
  .. ..$ threshold: NULL
  ..@ smooth      : num 0
  ..@ ngrams      : int 1
  ..@ skip        : int 0
  ..@ concatenator: chr "_"
  ..@ version     : int [1:3] 1 4 3
  ..@ docvars     :'data.frame':        225 obs. of  0 variables
  ..@ i           : int [1:2724] 0 125 142 0 2 8 9 11 13 18 ...
  ..@ p           : int [1:855] 0 3 74 185 194 211 309 320 325 366 ...
  ..@ Dim         : int [1:2] 225 854
  ..@ Dimnames    :List of 2
  .. ..$ docs    : chr [1:225] "text1" "text2" "text3" "text4" ...
  .. ..$ features: chr [1:854] "story" "of" "the" "door" ...
  ..@ x           : Named num [1:2724] 0.25 0.05 0.0333 0.25 0.0714 ...
  .. ..- attr(*, "names")= chr [1:2724] "text1" "text126" "text143" "text1" ...
  ..@ factors     : list()
>
```

25. Term frequency inverse document frequency: Function to compute Term Frequency Inverse Document Frequency

```
> texttfidf <- dfm_tfidf(dfmtext1, scheme_tf = "count",scheme_df = "inverse")
> texttfidf
Document-feature matrix of: 225 documents, 854 features (98.6% sparse).
> texttfidf@i
  [1]   0 125 142   0   2   8   9  11  13  18  19  23  24  25  27  33  35  36  39  40  43  45  46  49  50  53  54  56  57  65  66  73  77  78  81  82  84  87 102 104 106 108 109
 [44] 112 113 115 119 123 125 128 135 137 144 145 147 150 153 161 163 164 166 173 182 187 190 192 197 199 200 204 207 210 214 221   0   2   5   8   9  10  11  13  15  17  18  21
 [87]  22  23  24  25  26  27  28  29  33  34  36  40  41  43  44  47  50  52  53  54  55  56  57  59  60  61  65  66  70  77  79  82  84  86  87  88  92  94  95  96  97 100 101
[130] 103 105 106 107 109 110 111 113 118 119 120 121 122 123 125 126 128 130 133 134 135 136 138 144 145 147 148 150 153 158 160 161 163 164 165 166 167 169 171 173 174 175 177
[173] 178 181 186 187 190 198 199 209 210 214 215 219 224   0  55  58  69 122 130 148 158 172   2  21  28  65  73  76 140 142 152 155 158 171 182 189 192 203 219   2   5   9  11
[216]  14  18  19  21  23  25  27  28  29  31  33  37  40  41  45  50  54  57  59  63  65  67  73  76  79  83  86  89  90  92  94  96  98  99 100 101 103 105 107 109 110 112 115
[259] 116 117 120 126 127 131 132 136 137 138 140 142 146 147 148 150 152 155 158 160 162 166 169 171 172 174 175 177 182 184 187 189 192 194 195 196 198 199 200 203 204 205 207
[302] 209 211 215 216 219 220 221 224   2  21  73 140 152 158 182 192 203 209 219   2  65 169 186 224   2   5   9  10  16  21  25  29  31  40  52  55  58  74  76  78  83  85  89
[345]  90  92  96  98 100 101 103 105 106 113 119 120 125 126 127 131 137 138 143 152 160 189   2   3  10  19  22  23  29  33  39  40  41  48  53  55  56  61  71  73  77  78  80
[388]  81  82  83  84  85  90  92  99 102 103 108 112 113 117 118 122 123 124 125 129 131 136 137 142 143 150 160 162 163 166 172 173 176 181 182 189 192 196 197 203 204 214 216
[431] 217 219 221   2  23  29  82  83  87  90 107 113 129 143 144 146 187 190 192 196 198 219   2   2   2  24  28  32  33  36  39  44  53  69  74  81  94 104 110 120 121 124 127
[474] 128 129 143 144 149 173 178 186 187 224   3   7  19 112 158 196 219 221   3  80   3  31  34  53 152   3   3   4   6   9  10  12  13  21  24  26  44  48  53  55  56  60  61
[517]  65  69  87  90  95  97 104 106 113 118 119 121 126 135 144 146 156 160 161 163 175 176 177 181 195 197 199 200 209 216 219   3   3   4   5   7   8  10  11  12  13  15  16
[560]  18  19  22  26  27  28  29  30  32  33  34  35  36  41  42  43  45  47  48  49  52  55  57  58  61  66  70  73  76  77  78  80  83  84  86  89  90  91  93  95  96  97 100
[603] 102 103 104 106 107 109 110 111 113 114 115 116 117 120 122 123 124 125 129 132 133 134 135 136 142 143 145 146 148 149 155 160 162 163 166 167 169 171 172 173 178 181 182
[646] 184 186 189 192 195 196 197 198 199 207 209 210 211 214 215 216 220 224   3   3   4   5   8  10  13  17  18  22  24  32  35  40  42  43  46  48  49  50  53  56  57  59  61
[689]  65  66  69  78  79  81  82  84  87  88  91  93  95  97 101 102 103 105 106 107 110 113 120 121 122 123 124 126 129 130 131 133 134 135 137 144 145 149 162 163 164 165 172
[732] 175 177 178 181 196 198 200 203 216 219   3   3   4   7   8  13  14  16  17  19  22  27  30  31  40  43  47  48  57  70  96 105 106 111 113 122 123 129 130 135 136 148 156
[775] 165 172 173 181 203 212   3   4   4   4   4  18  82 221   4   4   4 177 196 198   4   5   5  13  22  53  77  81  83  84  86  99 104 120 125 130 132 204   5  24   5   5   9
[818]  46  66  69  82 135   5   5  10  14  15  17  18  21  22  23  28  29  42  47  62  79  82  89  91  97  98  99 105 109 115 117 118 119 120 121 124 128 136 137 143 155 165 178
[861] 186 187 192 194 220 221   5   6   7   9  16  19  22  24  25  26  28  61  66  69  96 105 108 147 156 165 194   5  10   6 119 194 195   6   6   6   6  24  52  76 108 149 152
[904]   6  50   6   6   7  58 100 176   7   7  46  47  48  54   7  25  39  76  78 199 203   7  59  81  86 129 150   7   7   8  11  35  41  55  65  79  89  93  98 100 114 116 119
[947] 121 125 126 131 173 175 219   7   7  11  35  96 129 137 177 194 200 210 217   7  35 100 127   7  18  30  34  39  62   8   8   8   8   8  46  97 127 166 216   8 126   8   9
[990]   9 129   9  10  11  15  19  21  26  69  70  92 101 103 104 106 109 110 116 117 118 120 121 131 132 149 152 156 192 196 197 204 216 220   9   9  12  33  44  48  58  73 102
[1033] 105 113 121 122 124 130 132 143 148 149 158 194 214   9 207   9   9  10  10  10  11  12  21  29  33  61  80  82  87  95 112 118 123 127 131 142 147 171 177 181 186 200  10
[1076]  10 125 148  10  11  11  26  60  61  69  94  96  99 100 109 135 137 160 212 216  11  11  52  11  39  52  61  83  86  93  98 108 125 145 173 178 186  11  11  12  44  98 146
[1119] 198  12  12  12 163  12  12  12  13  13  55  13  13  13  13  31  39  43  62 166  13  14 109 212  14  14  14  14 152  14 127  14  14  15  69  70  71  73  74  76 115 116 117
[1162] 122 132 140 142 149 152 153 155 158 160 169 171 181 182 184 186 189 192 194 204 207 209 214 215 220 224  14  15  16  25  69  70  71  73  74  76  77  80  81  88  89  95 100
[1205] 101 114 115 116 117 118 122 123 124 125 130 132 133 134 138 140 142 149 152 153 155 156 158 160 162 167 169 171 175 177 178 179 181 182 184 186 187 189 190 192 194 198 200
[1248] 201 204 207 209 212 214 215 216 217 220 222 224  14  15  76 142 160 189 209 214 220 224  14  15  15  25  77  88  89  95 100 101 116 117 118 123 124 125 130 132 133 134 142
[1291] 153 155 162 175 177 178 182 186 189 198 200 216  15  25  88  95 100 101 118 123 125 130 134 142 162 177 178 182 186 198 200 216  15  15 204  15 178 220  15  15  76  83 152
[1334] 166  15 221  15  70  78  91  99 104 125 128 135 143 173 221 224  15  15  16  16  25  95 165  16  16  76 108 116 152 201 222  16  16  23  29  31  39  41  42  46  54  76  89
[1377]  90  94  98 111 115 125 127 137 138 145 155 161 166 184 189 200 209 210 211 215 216  16  16  17  22  79  98 196  17  81 120 164 205  17  17  17  84 127 146 169 182  17  18
[1420] 153 156  18  18  34  18 108 112  18  80  81  85 103 106 108 110 111 112 142 215  18  44  93 100 133 176 177 196  18  27  30  32  36  66 112 146  18  66  87 122  19  29  77
[1463]  92 103 119 158 160 178  19 135  19  19  19  73 166  19  21  27  55  61  93 102 116 166 172 199  21  27  21  21 131 194  21  22 107 111  22  36  45 149  22  22  42  22  22
[1506]  23  23  23  40  70 144 148 149 165 172 176 184 192 194 210 220  23  23  23  24 112  24  24 224  24  25  25 109  25  26  42  65  95 110 111 112  25  31  25  25  30  62  85
[1549] 109 156 172  26  96  26 126  26  26  26  45  48  90  94 103 115 162 167  26  27  27 104 110  27  27  27  27  28  28  28  91 105 111 118 200 201 216  28 211  28  65  76 142
[1592] 155 171 184 189 207  28  28  29  29  78  29  29  29  29  30  40  73 101 105 106 145 146 148 189  30  33  52  55  83  86  30 107 112 114 143  30  89 114 142 189 192 201 211
[1635]  30  35  30  65  84 109 172 210 214  30  30  31  31  31  83  85  94 146 187  31  31  34  36  39 110  31  45  32  32  73 116 137 140 142 158 169 182 189 192 209 219 220 224
[1678]  32  55  79  89 199 220  32 104 129  32  32  32  98 107 118 164 189  32  33  33  33  96 195  33 134  33  42  79  80  81  83 110 133 149 186 224  34  96  34  34  34  34  34
```

Part b: 10 longest sentences (in number of words)

To get the 10 longest sentences based on the number of words, we start by reading the text from the given text file, DrJekyllAndMrHyde.txt. Next, we use collapse paste to store the text as one string and then split them into different sentences at period(.) and single space using strsplit. We remove extra spaces from all the sentences and then get the number of words in each sentence. Next, we sort the sentences by word count in decreasing order and take the first ten which gives us the ten highest word counts. Now, we run a loop to check the ten highest word counts to the sentences that we have stored previously and display the ten sentences whose length matches these word counts.

```
#Read the text from the text file
text = readLines("DrJekyllAndMrHyde.txt")

#Use collapse paste to store the text as one string and then split at period(.) and then
a single space.
str = paste(text,collapse=" ")
splitstr = strsplit(str, ". ", fixed = TRUE)
str_new = list()

#Remove extra spaces
i = 1
for(s in splitstr){
  s = gsub("^\\s+|\\s+$", "", s)
  str_new[[i]] = s
  i = i + 1
}
str2 = str_new[[1]]

str2 = gsub("  ", " ", str2)

lengthSents = list()
```

```
#get length of words for each sentence
for (s in 1:length(str2)) {
  lengthSents[s] = sapply(strsplit(str2[s], " "), length)
}


#Function to sort sentences by number of words in decreasing order
sortnumlist = function(x) {
  n = length(x)
  for (k in n:2) {
    i = 1
    while (i < k) {
      if (x[[i]] < x[[i+1]]) {
        tmp = x[[i+1]]
        x[[i+1]] = x[[i]]
        x[[i]] = tmp
      }
      i = i+1
    }
  }
  x
}

#Store list of sorted sentences by words and get the first 10.
wordlist = sortnumlist(lengthSents)
wordlist = unique(wordlist)
wordlist = wordlist[1:10]

#Match word count to sentences stored in str2.
Sentences = list()
wordlist_final = list()
k = 1
for (t in 1:length(wordlist)) {
  for (s in 1:length(str2)) {
```

```
    if (sapply(strsplit(str2[s], " "), length) == wordlist[t]) {
      Sentences[k] = str2[s]
      wordlist_final[k] = wordlist[t]
      k = k + 1
    }
  }
}

Sentences = Sentences[1:10]
wordlist_final = wordlist_final[1:10]

#Display the result
print(paste0("The 10 longest sentences: "))
Sentences

print(paste0("Word count of the 10 longest sentences: "))
wordlist_final
```

<u>The Output showing the 10 longest sentences and their word counts</u>

```
> #Display the result
> print(paste0("The 10 longest sentences: "))
[1] "The 10 longest sentences: "
> Sentences
[[1]]
[1] "Your master, Poole, is plainly seized with one of those maladies that both torture and deform the sufferer; hence, for aught I know, the alteration of his voice; hence the mask and the avoidance of his friends; hence his eagerness to find this drug, by means of which the poor soul retains some hope of ultimate recovery—God grant that he be not deceived! There is my explanation; it is sad enough, Poole, ay, and appalling to consider; but it is plain and natural, hangs well together, and delivers us from all exorbitant alarms.” “Sir,” said the butler, turning to a sort of mottled pallor, “that thing was not my master, and there’s the truth"
```

[[2]]
[1] "It was on the moral side, and in my own person, that I learned to recognise the thorough and primitive duality of man; I saw that, of the two natures that contended in the field of my consciousness, even if I could rightly be said to be either, it was only because I was radically both; and from an early date, even before the course of my scientific discoveries had begun to suggest the most naked possibility of such a miracle, I had learned to dwell with pleasure, as a beloved daydream, on the thought of the separation of these elements"

[[3]]
[1] "Hyde was thenceforth impossible; whether I would or not, I was now confined to the better part of my existence; and O, how I rejoiced to think of it! with what willing humility I embraced anew the restrictions of natural life! with what sincere renunciation I locked the door by which I had so often gone and come, and ground the key under my heel! The next day, came the news that the murder had not been overlooked, that the guilt of Hyde was patent to the world, and that the victim was a man high in public estimation"

[[4]]
[1] "How could the presence of these articles in my house affect either the honour, the sanity, or the life of my flighty colleague? If his messenger could go to one place, why could he not go to another? And even granting some impediment, why was this gentleman to be received by me in secret? The more I reflected the more convinced I grew that I was dealing with a case of cerebral disease; and though I dismissed my servants to bed, I loaded an old revolver, that I might be found in some posture of self-defence"

[[5]]
[1] "Fell? or is it the mere radiance of a foul soul that thus transpires through, and transfigures, its clay continent? The last, I think; for, O my poor old Harry Jekyll, if ever I read Satan’s signature upon a face, it is on that of your new friend.” Round the corner from the by-street, there was a square of ancient, handsome houses, now for the most part decayed from their high estate and let in flats and chambers to all

sorts and conditions of men; map-engravers, architects, shady lawyers and the agents of obscure enterprises"

[[6]]
[1] "I could not think that this earth contained a place for sufferings and terrors so unmanning; and you can do but one thing, Utterson, to lighten this destiny, and that is to respect my silence." Utterson was amazed; the dark influence of Hyde had been withdrawn, the doctor had returned to his old tasks and amities; a week ago, the prospect had smiled with every promise of a cheerful and an honoured age; and now in a moment, friendship, and peace of mind, and the whole tenor of his life were wrecked"

[[7]]
[1] "And in the meantime, if you can sit and talk with me of other things, for God's sake, stay and do so; but if you cannot keep clear of this accursed topic, then in God's name, go, for I cannot bear it." As soon as he got home, Utterson sat down and wrote to Jekyll, complaining of his exclusion from the house, and asking the cause of this unhappy break with Lanyon; and the next day brought him a long answer, often very pathetically worded, and sometimes darkly mysterious in drift"

[[8]]
[1] "That part of me which I had the power of projecting, had lately been much exercised and nourished; it had seemed to me of late as though the body of Edward Hyde had grown in stature, as though (when I wore that form) I were conscious of a more generous tide of blood; and I began to spy a danger that, if this were much prolonged, the balance of my nature might be permanently overthrown, the power of voluntary change be forfeited, and the character of Edward Hyde become irrevocably mine"

[[9]]
[1] "This person (who had thus, from the first moment of his entrance, struck in me what I can only describe as a disgustful curiosity) was dressed in a fashion that would have made an ordinary person laughable; his clothes, that is to say, although they were of rich and sober fabric, were enormously too large for him in every

measurement—the trousers hanging on his legs and rolled up to keep them from the ground, the waist of the coat below his haunches, and the collar sprawling wide upon his shoulders"

[[10]]
[1] "As the cab drew up before the address indicated, the fog lifted a little and showed him a dingy street, a gin palace, a low French eating house, a shop for the retail of penny numbers and twopenny salads, many ragged children huddled in the doorways, and many women of many different nationalities passing out, key in hand, to have a morning glass; and the next moment the fog settled down again upon that part, as brown as umber, and cut him off from his blackguardly surroundings"

```
>
> print(paste0("Word count of the 10 longest sentences: "))
[1] "Word count of the 10 longest sentences: "
> wordlist_final
[[1]]
[1] 114

[[2]]
[1] 101

[[3]]
[1] 99

[[4]]
[1] 96

[[5]]
[1] 95

[[6]]
[1] 92
```

```
[[7]]
[1] 91

[[8]]
[1] 91

[[9]]
[1] 90

[[10]]
[1] 87
```

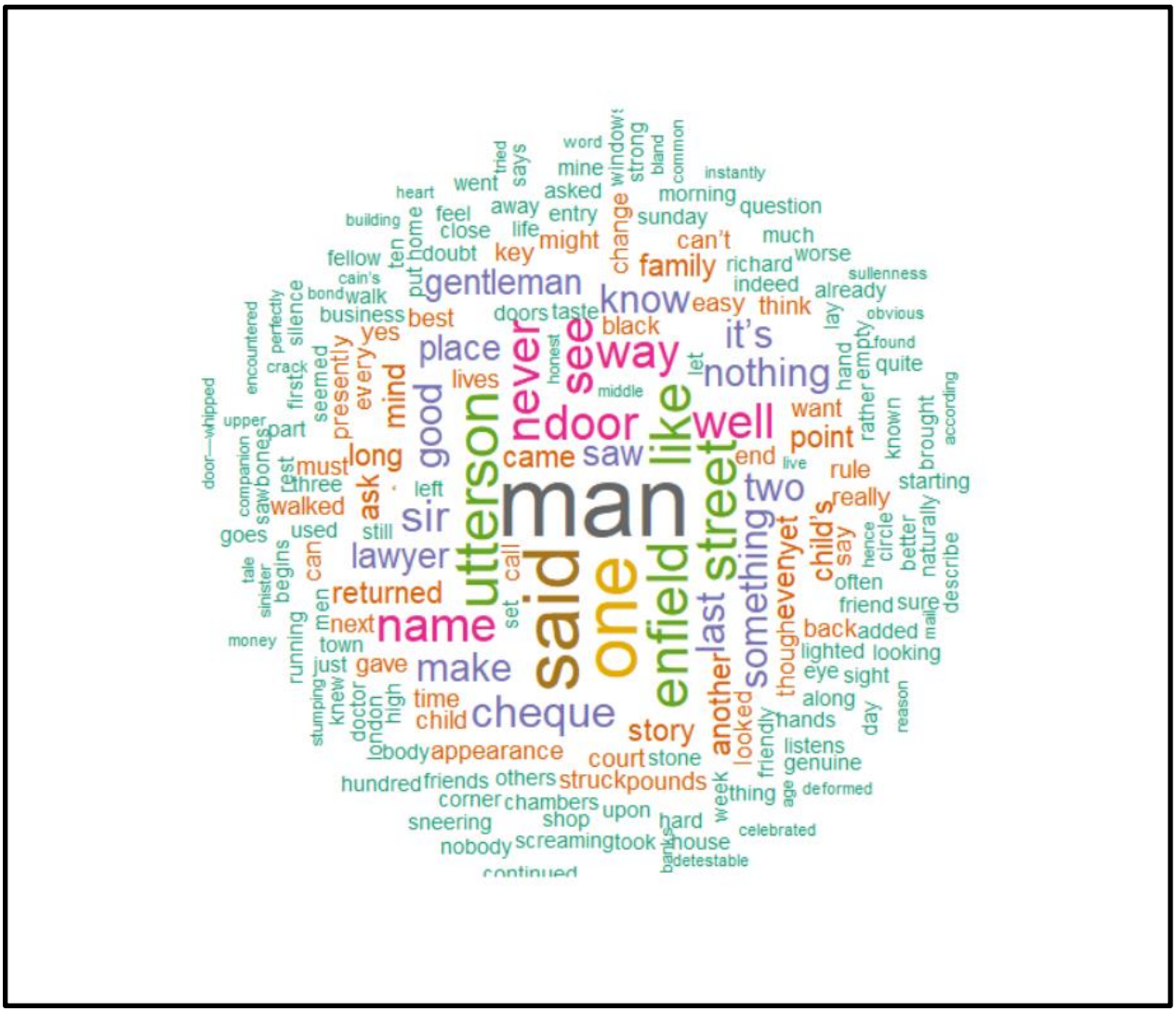
Part c: Word Cloud and Dendrogram

For displaying the wordcloud, we start with content transformation. Then, there is the mapping function which is used to remove special characters. The text is converted to lower class, the numbers are removed, the white spaces are eliminated, and punctuations are removed.

Chapter 1

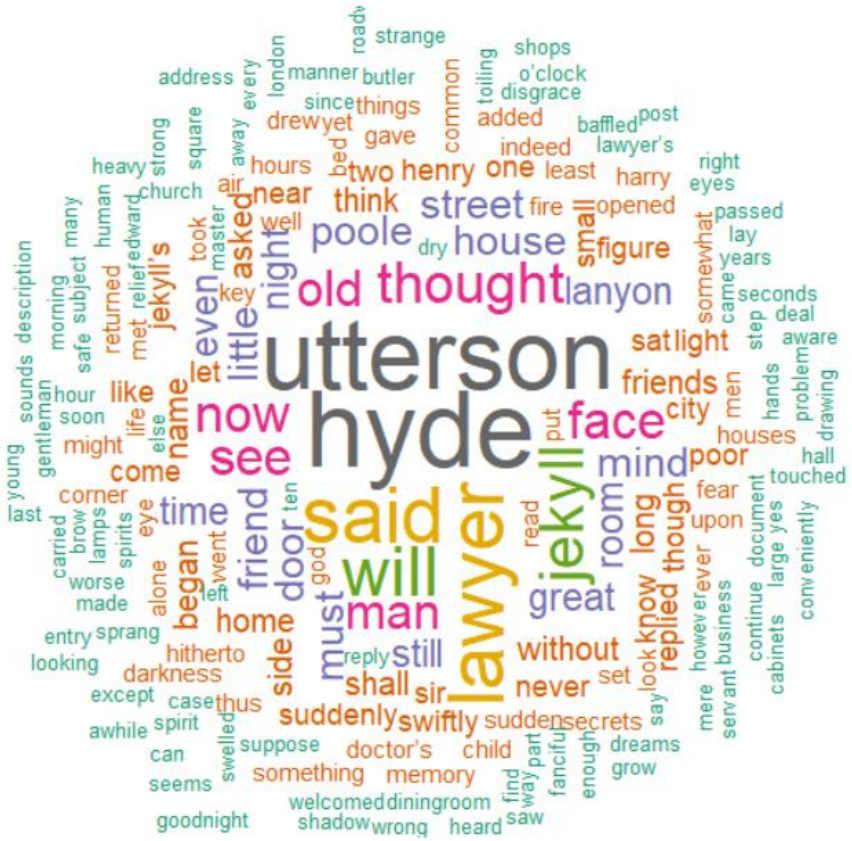
```
text1 <- readLines(file.choose())
docs1 <- Corpus(VectorSource(text1))
inspect(docs1)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs1 <- tm_map(docs1, toSpace, "/")
docs1 <- tm_map(docs1, toSpace, "@")
docs1 <- tm_map(docs1, toSpace, " â ")
docs1 <- tm_map(docs1, toSpace, "\\|")
# Convert the text to lower case
docs1 <- tm_map(docs1, content_transformer(tolower))
# Remove numbers
docs1 <- tm_map(docs1, removeNumbers)
# Remove english common stopwords
docs1 <- tm_map(docs1, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs1 <- tm_map(docs1, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs1 <- tm_map(docs1, removePunctuation)
# Eliminate extra white spaces
docs1 <- tm_map(docs1, stripWhitespace)
dtm1 <- TermDocumentMatrix(docs1)
m1 <- as.matrix(dtm1)
v1 <- sort(rowSums(m1),decreasing=TRUE)
d1 <- data.frame(word = names(v1),freq=v1)
```

```
noquote(d1)
head(d1, 10)
set.seed(1234)
wordcloud(words = d1$word, freq = d1$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))
```



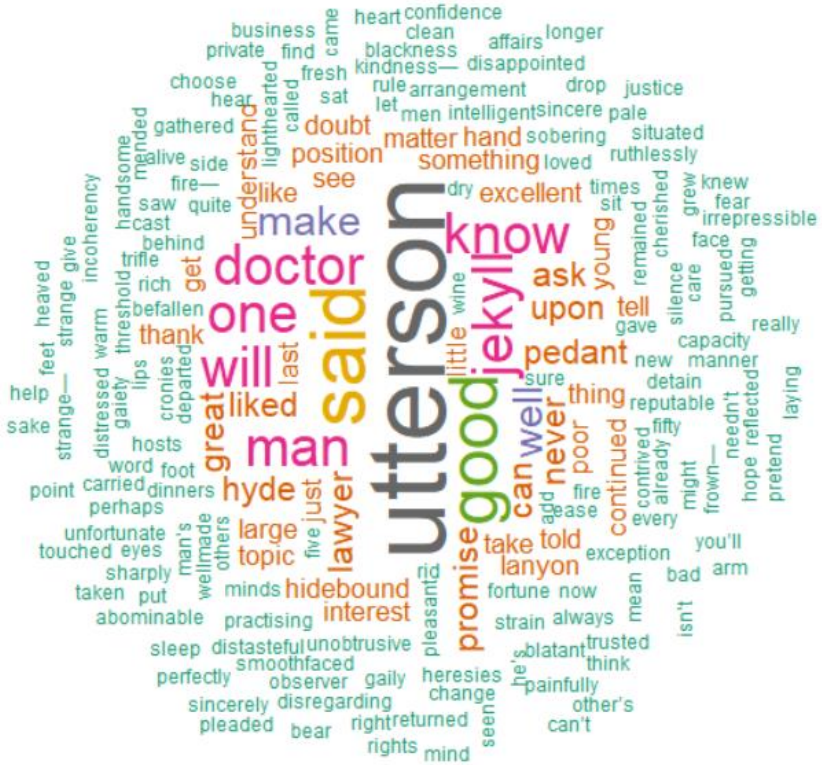
Chapter 2

```
text2 <- readLines(file.choose())
docs2 <- Corpus(VectorSource(text2))
inspect(docs2)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs2 <- tm_map(docs2, toSpace, "/")
docs2 <- tm_map(docs2, toSpace, "@")
docs2 <- tm_map(docs2, toSpace, " â ")
docs2 <- tm_map(docs2, toSpace, "\\|")
# Convert the text to lower case
docs2 <- tm_map(docs2, content_transformer(tolower))
# Remove numbers
docs2 <- tm_map(docs2, removeNumbers)
# Remove english common stopwords
docs2 <- tm_map(docs2, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs2 <- tm_map(docs2, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs2 <- tm_map(docs2, removePunctuation)
# Eliminate extra white spaces
docs2 <- tm_map(docs2, stripWhitespace)
dtm2 <- TermDocumentMatrix(docs2)
m2 <- as.matrix(dtm2)
v2 <- sort(rowSums(m2),decreasing=TRUE)
d2 <- data.frame(word = names(v2),freq=v2)
noquote(d2)
head(d2, 10)
set.seed(1234)
wordcloud(words = d2$word, freq = d2$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))
```


utterson
hyde
lawyer
said
will
jekyll
thought
old
now
see
face
man
poole
street
house
lanyon
mind
great
time
friend
door
room
long
must
still
shall
without
never
little
night
even
asked
name
jekyll's
think
two
henry
one
near
well
hours
gave
things
drew
yet
since
manner
butler
strange
road
london
every
away
common
toiling
shops
o'clock
disgrace
added
indeed
least
baffled
post
lawyer's
harry
right
eyes
opened
fire
small
figure
dry
passed
lay
years
somewhat
came
seconds
deal
step
aware
sat
light
friends
city
men
hands
problem
drawing
houses
put
poor
hall
touched
fear
upon
document
large
yes
conveniently
read
know
replied
though
ever
however
business
continue
cabinets
look
set
say
mere
servant
sudden
secrets
dreams
grow
enough
fanciful
part
way
find
saw
heard
diningroom
welcomed
wrong
shadow
memory
child
doctor's
something
suppose
swiftly
suddenly
sir
reply
god
ten
side
home
began
went
left
alone
eye
spirits
lamps
brow
carried
last
young
corner
come
might
like
let
key
took
master
air
church
edward
human
many
heavy
strong
square
address
description
morning
subject
safe
returned
relief
met
sounds
gentleman
hour
soon
life
else
worse
made
entry
sprang
looking
hitherto
darkness
except
case
thus
awhile
spirit
swelled
can
seems
goodnight
bed

Chapter 3

```
text3 <-readLines(file.choose())
docs3 <- Corpus(VectorSource(text3))
inspect(docs3)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs3 <- tm_map(docs3, toSpace, "/")
docs3 <- tm_map(docs3, toSpace, "@")
docs3 <- tm_map(docs3, toSpace, " â ")
docs3 <- tm_map(docs3, toSpace, "\\|")
# Convert the text to lower case
docs3 <- tm_map(docs3, content_transformer(tolower))
# Remove numbers
docs3 <- tm_map(docs3, removeNumbers)
# Remove english common stopwords
docs3 <- tm_map(docs3, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs3 <- tm_map(docs3, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs3 <- tm_map(docs3, removePunctuation)
# Eliminate extra white spaces
docs3 <- tm_map(docs3, stripWhitespace)
dtm3 <- TermDocumentMatrix(docs3)
m3 <- as.matrix(dtm3)
v3 <- sort(rowSums(m3),decreasing=TRUE)
d3 <- data.frame(word = names(v3),freq=v3)
noquote(d3)
head(d3, 10)
set.seed(1234)
wordcloud(words = d3$word, freq = d3$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))
```


utterson
said
know
doctor
one
will
man
jekyll
good
make
hyde
liked
great
ask
upon
pedant
thing
never
well
can
lawyer
promise
lanyon
take
told
large
topic
hidebound
interest
position
doubt
matter
hand
something
excellent
see
like
understand
thank
get
last
little
continued
poor
young
tell
gave
new
manner
capacity
really
detain
reputable
fifty
contrived
already
might
every
exception
you'll
bad
arm
mean
isn't
fortune
now
strain
always
blatant
trusted
think
painfully
other's
can't
heresies
change
observer
gaily
smoothfaced
perfectly
distasteful
unobtrusive
sleep
practising
abominable
taken
put
minds
touched
eyes
unfortunate
sharply
perhaps
point
carried
dinners
foot
word
hosts
sake
help
strange—
distressed
galety
lips
cronies
departed
warm
threshold
rich
befallen
trifle
behind
cast
saw
quite
fire—
alive
side
gathered
mended
handsome
incoherency
give
heaved
feet
heart
confidence
clean
business
private
find
came
blackness
affairs
longer
kindness—
disappointed
choose
hear
fresh
sat
called
lighthearted
rule
arrangement
drop
justice
let
men
intelligent
sincere
pale
situated
sobering
ruthlessly
loved
dry
times
sit
remained
cherished
grew
knew
fear
irrepressible
face
silence
care
pursued
getting
wine
five
just
add
ease
fire
every
rid
pleasant
he's
seen
can't
returned
right
rights
mind
bear
pleaded
sincerely
disregarding
man's
wellmade
others
reflected
needn't
hope
pretend
laying
frown—

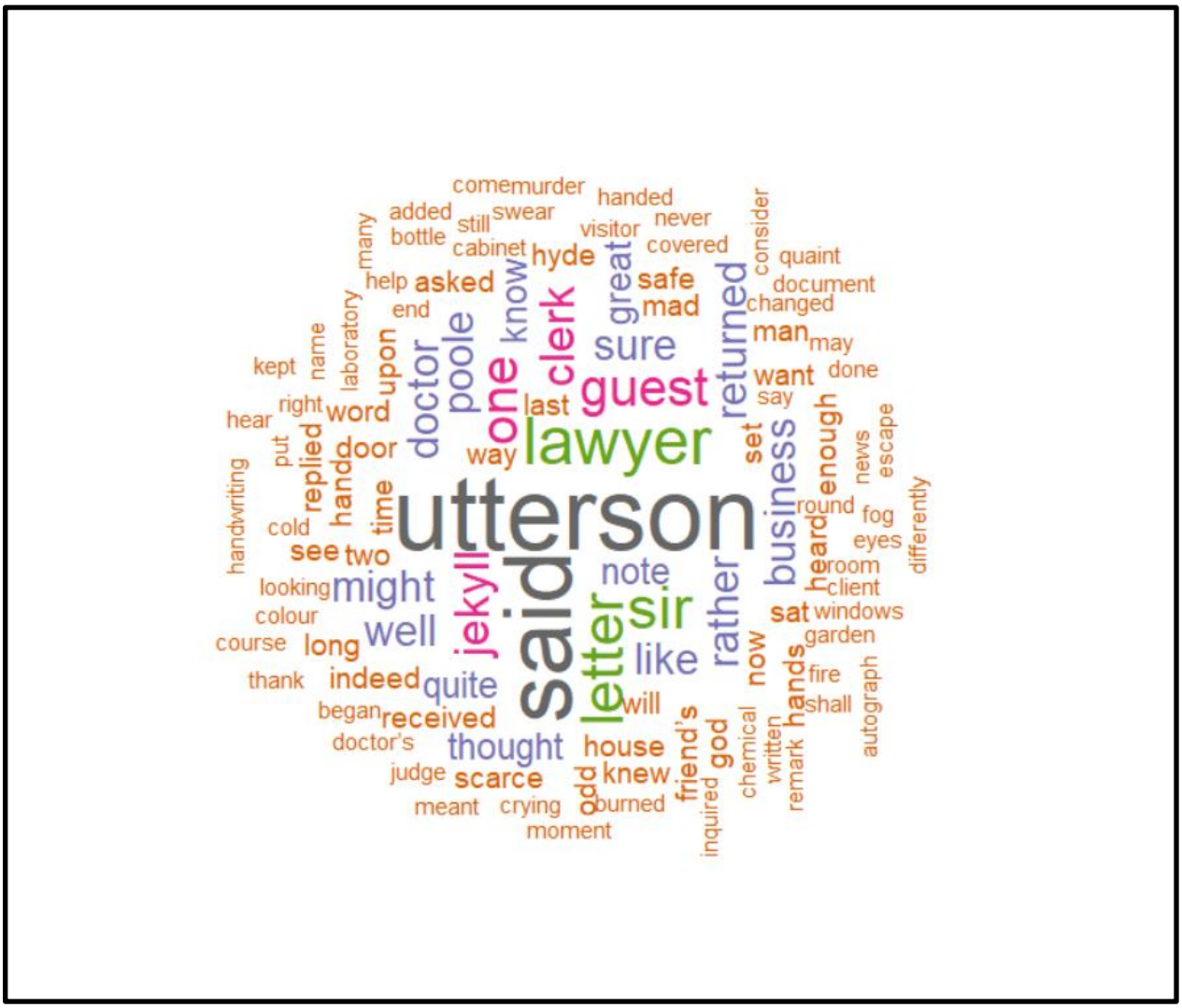
Chapter 4

```
text4 <-readLines(file.choose())
docs4 <- Corpus(VectorSource(text4))
inspect(docs4)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs4 <- tm_map(docs4, toSpace, "/")
docs4 <- tm_map(docs4, toSpace, "@")
docs4 <- tm_map(docs4, toSpace, " â ")
docs4 <- tm_map(docs4, toSpace, "\\|")
# Convert the text to lower case
docs4 <- tm_map(docs4, content_transformer(tolower))
# Remove numbers
docs4 <- tm_map(docs4, removeNumbers)
# Remove english common stopwords
docs4 <- tm_map(docs4, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs4 <- tm_map(docs4, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs4 <- tm_map(docs4, removePunctuation)
# Eliminate extra white spaces
docs4 <- tm_map(docs4, stripWhitespace)
dtm4 <- TermDocumentMatrix(docs4)
m4 <- as.matrix(dtm4)
v4 <- sort(rowSums(m4),decreasing=TRUE)
d4 <- data.frame(word = names(v4),freq=v4)
noquote(d4)
head(d4, 10)
set.seed(1234)
wordcloud(words = d4$word, freq = d4$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))
```


said
utterson
hyde
upon
never
moment
many
maid
stick
broken
seen
man
woman
fog
house
gone
body
inspector
appeared
officer
face
victim
street
one
good
great
small
gentleman
lane
morning
rooms
address
like
sir
lawyer
cheque
person
seemed
henry
carried
eyes
rolled
carrying
came
hand
found
cab
head
home
window
quarter
just
bore
may
bed
night
door
come
foot
bank
yes
watch
broke
high
two
old
next
strange
murderer
maid's
jekyll
half
wait
told
moon
nearly
heavy
part
nothing
say
lay
less
firstname
carew
city
brown
servant
recognise
particularly
surprised
used
will
however
done
conflagration
fury
tough
million
nowhere
pretty
haggard
book
word
eye
insensate
step
within
although
night
kindness
papers
seen
door
come
foot
can
master
sooner
exchange
unexpressed
splintered
probably
stature
quailed
life
glanced
daylight
thousand
aware
favourite
fainted
soho
sat
law
let
dislike
touch
beholders
twopenny
elegant
thoughts
agreed
alone
aged
clinched
gloomiest
conscious
hearth
wind
butt
calls
tears
london
hailing
plate
gin
dream
hair
much
stood
arrested

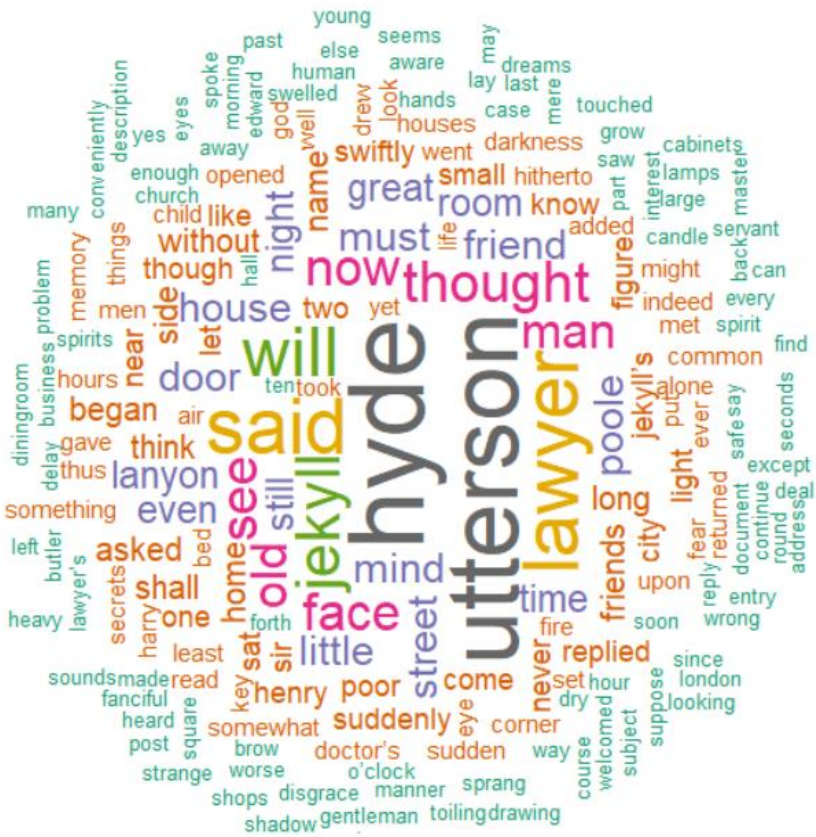
Chapter 5

```
text5 <- readLines(file.choose())
docs5 <- Corpus(VectorSource(text5))
inspect(docs5)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs5 <- tm_map(docs5, toSpace, "/")
docs5 <- tm_map(docs5, toSpace, "@")
docs5 <- tm_map(docs5, toSpace, " â ")
docs5 <- tm_map(docs5, toSpace, "\\|")
# Convert the text to lower case
docs5 <- tm_map(docs5, content_transformer(tolower))
# Remove numbers
docs5 <- tm_map(docs5, removeNumbers)
# Remove english common stopwords
docs5 <- tm_map(docs5, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs5 <- tm_map(docs5, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs5 <- tm_map(docs5, removePunctuation)
# Eliminate extra white spaces
docs5 <- tm_map(docs5, stripWhitespace)
dtm5 <- TermDocumentMatrix(docs5)
m5 <- as.matrix(dtm5)
v5 <- sort(rowSums(m5),decreasing=TRUE)
d5 <- data.frame(word = names(v5),freq=v5)
noquote(d5)
head(d5, 10)
set.seed(1235)
wordcloud(words = d5$word, freq = d5$freq, min.freq = 2,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))
```


comemurder
handed
added
swear
still
bottle
cabinet
visitor
never
covered
consider
quaint
many
hyde
great
safe
document
help
asked
know
mad
changed
end
clerk
man
may
sure
returned
kept
name
laboratory
upon
doctor
poole
one
guest
want
done
say
right
word
last
hear
put
replied
door
way
lawyer
set
business
enough
news
escape
handwriting
hand
time
utterson
round
fog
differently
cold
see
two
heard
eyes
room
looking
might
note
client
jekyll
said
sir
sat
windows
colour
rather
garden
course
long
well
letter
like
now
hands
fire
thank
indeed
quite
will
shall
autograph
began
received
friend's
god
chemical
written
doctor's
thought
house
remark
judge
scarce
knew
odd
meant
crying
burned
inquired
moment

Chapter 6

```
text6 <- readLines(file.choose())
docs6 <- Corpus(VectorSource(text6))
inspect(docs6)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs6 <- tm_map(docs6, toSpace, "/")
docs6 <- tm_map(docs6, toSpace, "@")
docs6 <- tm_map(docs6, toSpace, " â ")
docs6 <- tm_map(docs6, toSpace, "\\|")
# Convert the text to lower case
docs6 <- tm_map(docs6, content_transformer(tolower))
# Remove numbers
docs6 <- tm_map(docs6, removeNumbers)
# Remove english common stopwords
docs6 <- tm_map(docs6, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs6 <- tm_map(docs6, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs6 <- tm_map(docs6, removePunctuation)
# Eliminate extra white spaces
docs6 <- tm_map(docs6, stripWhitespace)
dtm6 <- TermDocumentMatrix(docs6)
m6 <- as.matrix(dtm6)
v6 <- sort(rowSums(m6),decreasing=TRUE)
d6 <- data.frame(word = names(v6),freq=v6)
noquote(d6)
head(d6, 10)
set.seed(1236)
wordcloud(words = d6$word, freq = d6$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.36,
        colors=brewer.pal(8, "Dark2"))
```


young
seems
past
else
aware
may
dreams
human
lay
last
spoke
morning
edward
swelled
god
well
drew
look
hands
case
mere
touched
description
conveniently
eyes
yes
houses
darkness
grow
away
swiftly
went
cabinets
saw
enough
opened
small
hitherto
lamps
interest
part
large
master
church
great
room
know
many
child
like
name
night
added
candle
servant
without
must
friend
figure
life
though
hall
now
thought
might
back
can
memory
things
indeed
every
problem
men
side
house
two
yet
met
spirit
spirits
near
let
will
hyde
man
find
common
hours
door
ten
took
jekyll's
alone
put
ever
safe
say
seconds
diningroom
business
began
air
poole
delay
gave
think
said
utterson
lawyer
thus
lanyon
see
still
jekyll
light
except
deal
something
even
long
returned
document
continue
round
address
left
butler
lawyer's
bed
asked
old
city
fear
secrets
shall
home
mind
friends
upon
reply
entry
heavy
one
harry
face
time
forth
fire
soon
wrong
least
sat
sir
little
street
replied
since
sounds
made
read
key
henry
poor
come
never
set
hour
dry
london
fanciful
heard
square
somewhat
suddenly
eye
corner
looking
suppose
post
brow
doctor's
sudden
way
subject
welcomed
course
strange
worse
o'clock
shops
disgrace
manner
sprang
shadow
gentleman
toiling
drawing

Chapter 7

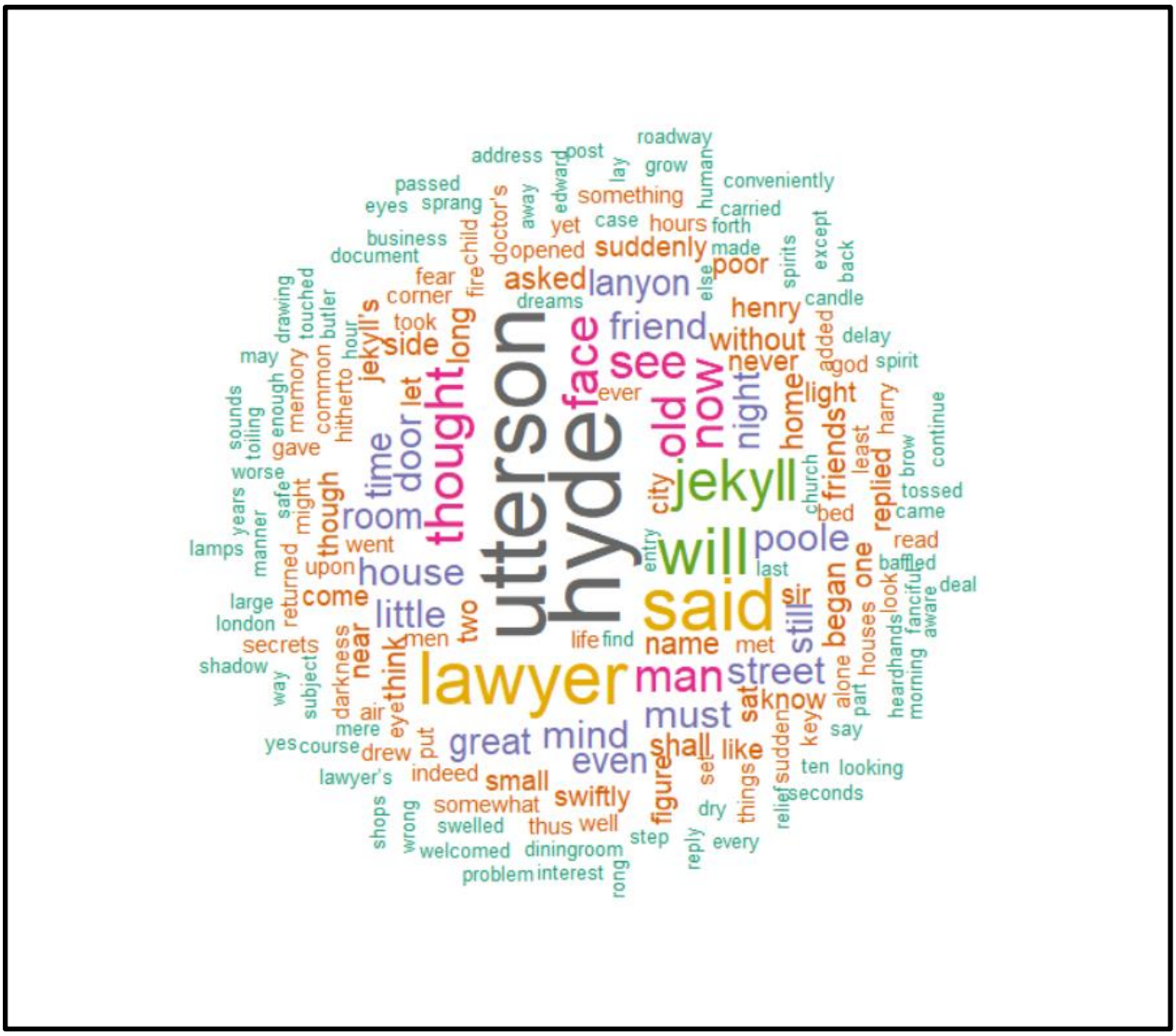
```
text7 <- readLines(file.choose())
docs7 <- Corpus(VectorSource(text7))
inspect(docs7)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs7 <- tm_map(docs7, toSpace, "/")
docs7 <- tm_map(docs7, toSpace, "@")
docs7 <- tm_map(docs7, toSpace, " â ")
docs7 <- tm_map(docs7, toSpace, "\\|")
# Convert the text to lower case
docs7 <- tm_map(docs7, content_transformer(tolower))
# Remove numbers
docs7 <- tm_map(docs7, removeNumbers)
# Remove english common stopwords
docs7 <- tm_map(docs7, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs7 <- tm_map(docs7, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs7 <- tm_map(docs7, removePunctuation)
# Eliminate extra white spaces
docs7 <- tm_map(docs7, stripWhitespace)
dtm7 <- TermDocumentMatrix(docs7)
m7 <- as.matrix(dtm7)
v7 <- sort(rowSums(m7),decreasing=TRUE)
d7 <- data.frame(word = names(v7),freq=v7)
noquote(d7)
head(d7, 10)
set.seed(1237)
wordcloud(words = d7$word, freq = d7$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.37,
        colors=brewer.pal(8, "Dark2"))
```


young
heavy
post
spoke
o'clock
lawyer's
suppose
church
large
memory
eyes
however
darkness
way
touched
master
fanciful
replied
doctor's
past
drew
deal
figure
sudden
lanyon
great
hall
things
tossed
awhile
can
wrong
seems
passed
harry
looking
come
never
room
near
every
heard
key
even
friend
night
hour
morning
part
safe
men
fire
name
thus
came
lawyer
said
poole
light
find
carried
candle
without
henry
side
see
one
houses
secrets
back
yes
subject
london
face
street
air
set
let
man
will
long
eye
lamps
hands
toiling
know
hyde
door
thought
two
corner
child
fear
business
seconds
little
life
met
utterson
house
though
least
put
diningroom
poor
began
home
opened
strong
round
disgrace
jekyll
time
sat
like
hitherto
look
except
since
asked
now
old
shall
suddenly
case
last
grow
lay
city
mind
still
might
servant
upon
think
sir
must
indeed
friends
forth
continue
jekyll's
small
address
swiftly
took
alone
hours
delay
manner
else
goodnight
enough
somewhat
common
left
went
human
worse
added
shadow
returned
read
problem
reply
baffled
square
butler
welcomed
cabinets
dreams
interest
may

Chapter 8

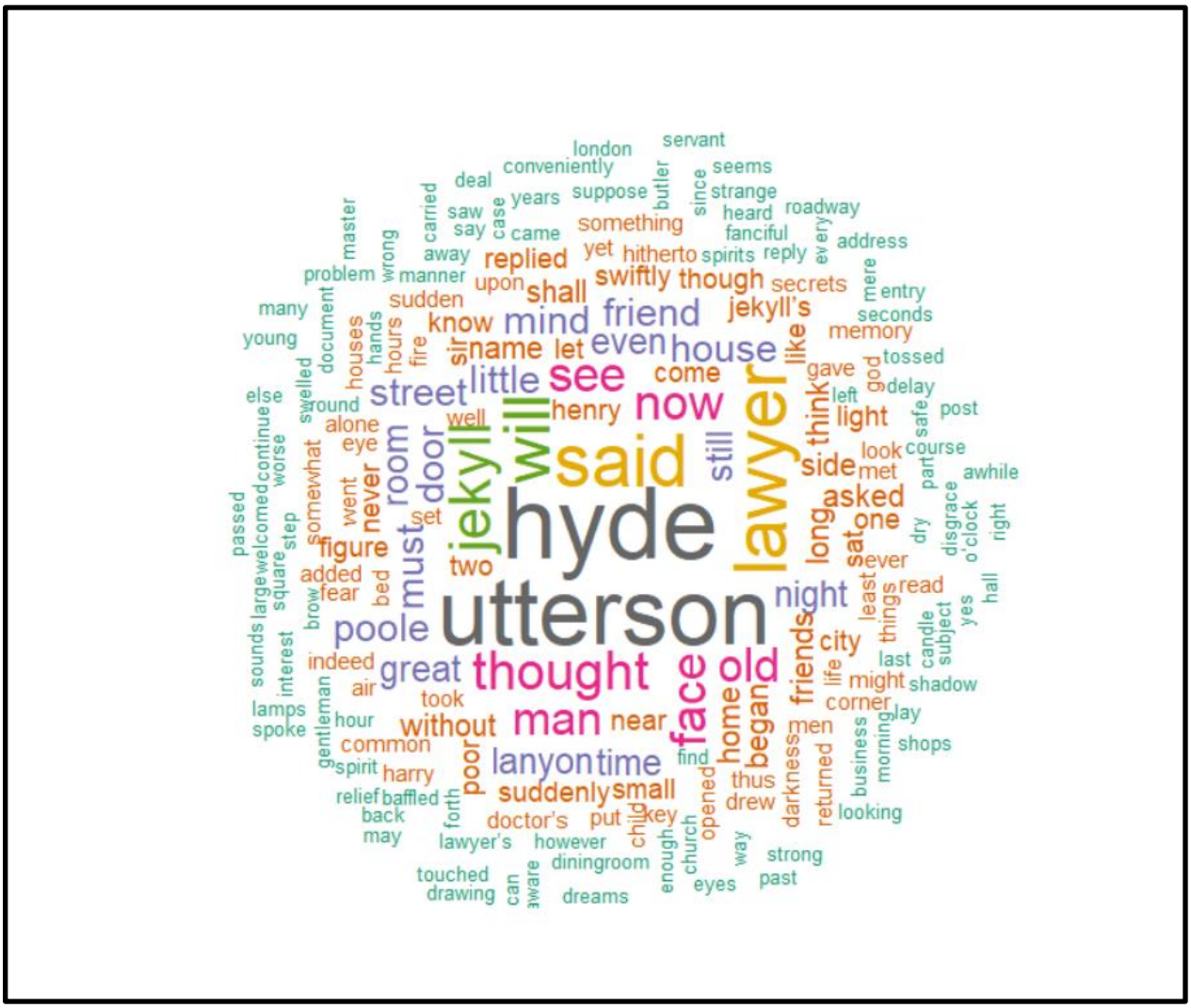
```
text8 <-
readLines("C://Users//aneri//OneDrive//Documents//RProject3//Projec3//Text
Document//Chapter 8.txt")
docs8 <- Corpus(VectorSource(text8))
inspect(docs8)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs8 <- tm_map(docs8, toSpace, "/")
docs8 <- tm_map(docs8, toSpace, "@")
docs8 <- tm_map(docs8, toSpace, " â ")
docs8 <- tm_map(docs8, toSpace, "\\|")
# Convert the text to lower case
docs8 <- tm_map(docs8, content_transformer(tolower))
# Remove numbers
docs8 <- tm_map(docs8, removeNumbers)
# Remove english common stopwords
docs8 <- tm_map(docs8, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs8 <- tm_map(docs8, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs8 <- tm_map(docs8, removePunctuation)
# Eliminate extra white spaces
docs8 <- tm_map(docs8, stripWhitespace)
dtm8 <- TermDocumentMatrix(docs8)
m8 <- as.matrix(dtm8)
v8 <- sort(rowSums(m8),decreasing=TRUE)
d8 <- data.frame(word = names(v8),freq=v8)
noquote(d8)
head(d8, 10)
set.seed(1238)
wordcloud(words = d8$word, freq = d8$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.38,
```

colors=brewer.pal(8, "Dark2"))



Chapter 9

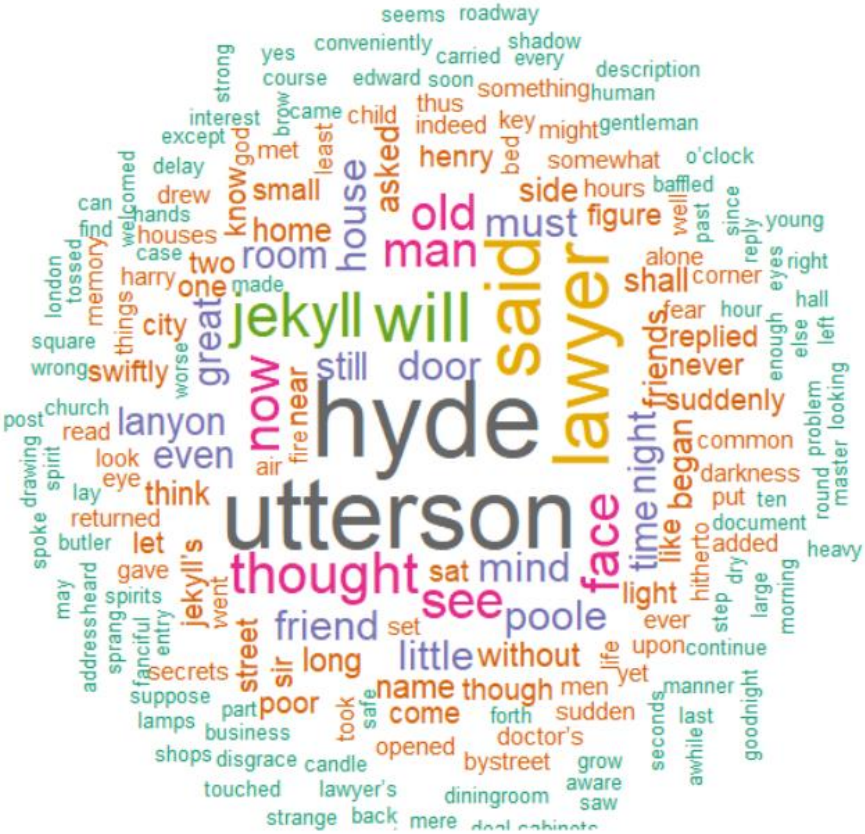
```
text9 <- readLines(file.choose())
docs9 <- Corpus(VectorSource(text9))
inspect(docs9)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs9 <- tm_map(docs9, toSpace, "/")
docs9 <- tm_map(docs9, toSpace, "@")
docs9 <- tm_map(docs9, toSpace, " â ")
docs9 <- tm_map(docs9, toSpace, "\\|")
# Convert the text to lower case
docs9 <- tm_map(docs9, content_transformer(tolower))
# Remove numbers
docs9 <- tm_map(docs9, removeNumbers)
# Remove english common stopwords
docs9 <- tm_map(docs9, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs9 <- tm_map(docs9, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs9 <- tm_map(docs9, removePunctuation)
# Eliminate extra white spaces
docs9 <- tm_map(docs9, stripWhitespace)
dtm9 <- TermDocumentMatrix(docs9)
m9 <- as.matrix(dtm9)
v9 <- sort(rowSums(m9),decreasing=TRUE)
d9 <- data.frame(word = names(v9),freq=v9)
noquote(d9)
head(d9, 10)
set.seed(1239)
wordcloud(words = d9$word, freq = d9$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.39,
        colors=brewer.pal(9, "Dark2"))
```



Chapter 10

```
text10 <- readLines(file.choose())
docs10 <- Corpus(VectorSource(text10))
inspect(docs10)
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs10 <- tm_map(docs10, toSpace, "/")
docs10 <- tm_map(docs10, toSpace, "@")
docs10 <- tm_map(docs10, toSpace, " â ")
docs10 <- tm_map(docs10, toSpace, "\\|")
# Convert the text to lower case
```

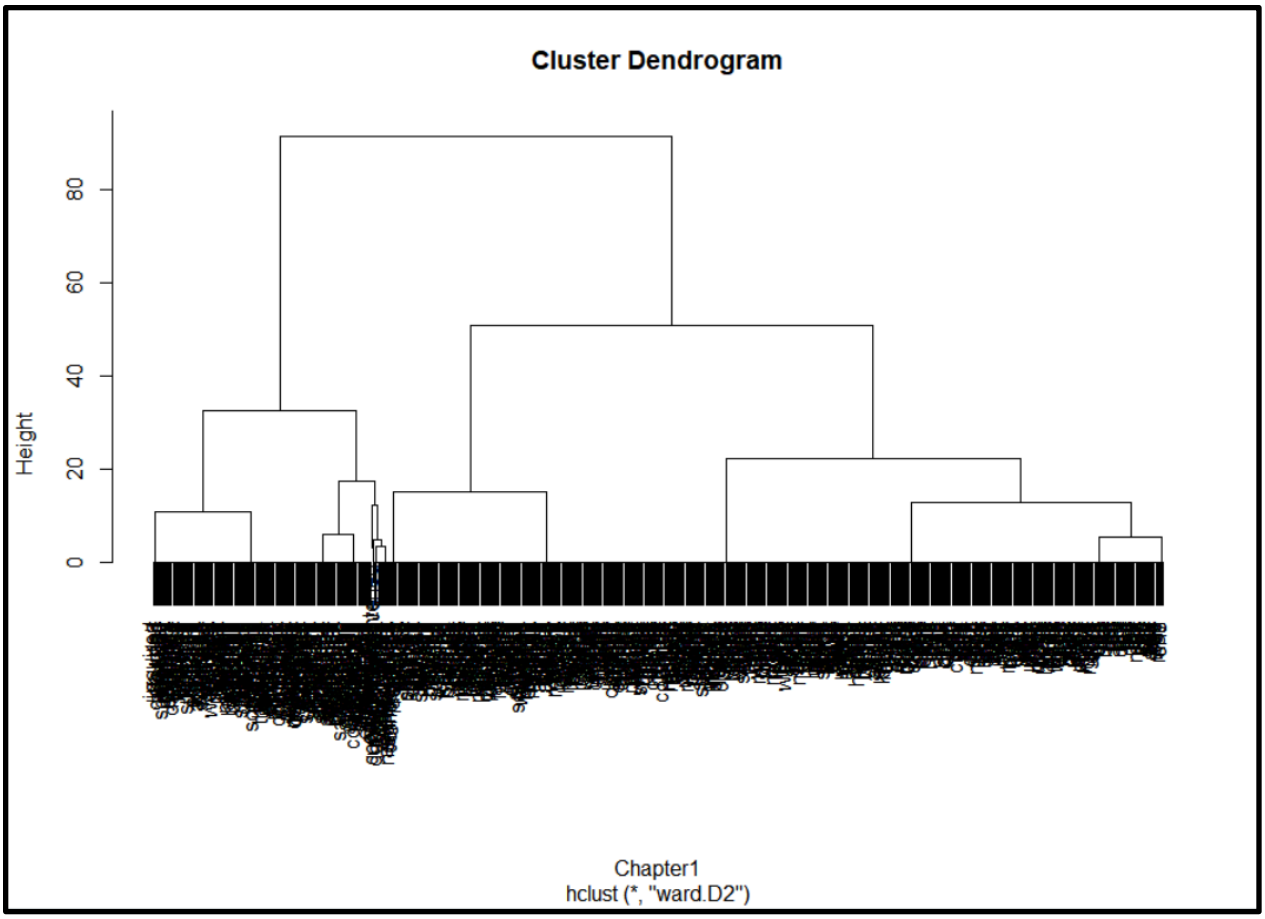
```
docs10 <- tm_map(docs10, content_transformer(tolower))
# Remove numbers
docs10 <- tm_map(docs10, removeNumbers)
# Remove english common stopwords
docs10 <- tm_map(docs10, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
docs10 <- tm_map(docs10, removeWords, c("blabla1", "blabla2"))
# Remove punctuations
docs10 <- tm_map(docs10, removePunctuation)
# Eliminate extra white spaces
docs10 <- tm_map(docs10, stripWhitespace)
dtm10 <- TermDocumentMatrix(docs10)
m10 <- as.matrix(dtm10)
v10 <- sort(rowSums(m10),decreasing=TRUE)
d10 <- data.frame(word = names(v10),freq=v10)
noquote(d10)
head(d10, 10)
set.seed(12310)
wordcloud(words = d10$word, freq = d10$freq, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.310,
        colors=brewer.pal(10, "Dark2"))
```


hyde
utterson
lawyer
said
jekyll
will
thought
face
see
friend
poole
mind
time
night
door
lanyon
even
now
house
asked
henry
old
man
must
figure
room
home
small
side
shall
replied
never
suddenly
began
darkness
common
little
without
though
name
come
long
street
sir
poor
secrets
think
let
returned
swiftly
great
city
one
two
harry
jekyll's
still
near
fire
air
look
eye
lay
church
read
post
spoke
drawing
spirit
butler
gave
spirits
may
heard
sprang
fanciful
entry
address
suppose
lamps
business
shops
disgrace
candle
touched
lawyer's
strange
back
mere
opened
bystreet
doctor's
diningroom
aware
saw
grow
forth
sudden
last
seconds
awhile
goodnight
manner
yet
upon
continue
life
sat
set
light
ever
step
dry
large
morning
hitherto
added
document
put
ten
round
master
problem
looking
heavy
hour
fear
hall
else
enough
left
eyes
right
reply
since
past
well
alone
corner
friends
hours
baffled
somewhat
o'clock
gentleman
might
key
indeed
thus
child
came
brow
interest
except
met
least
god
know
delay
drew
welcomed
can
hands
find
houses
case
tossed
memory
london
things
square
wrong
worse
made
seems
roadway
conveniently
shadow
carried
every
description
yes
course
edward
soon
something
human
strong
bed
young

Dendrogram

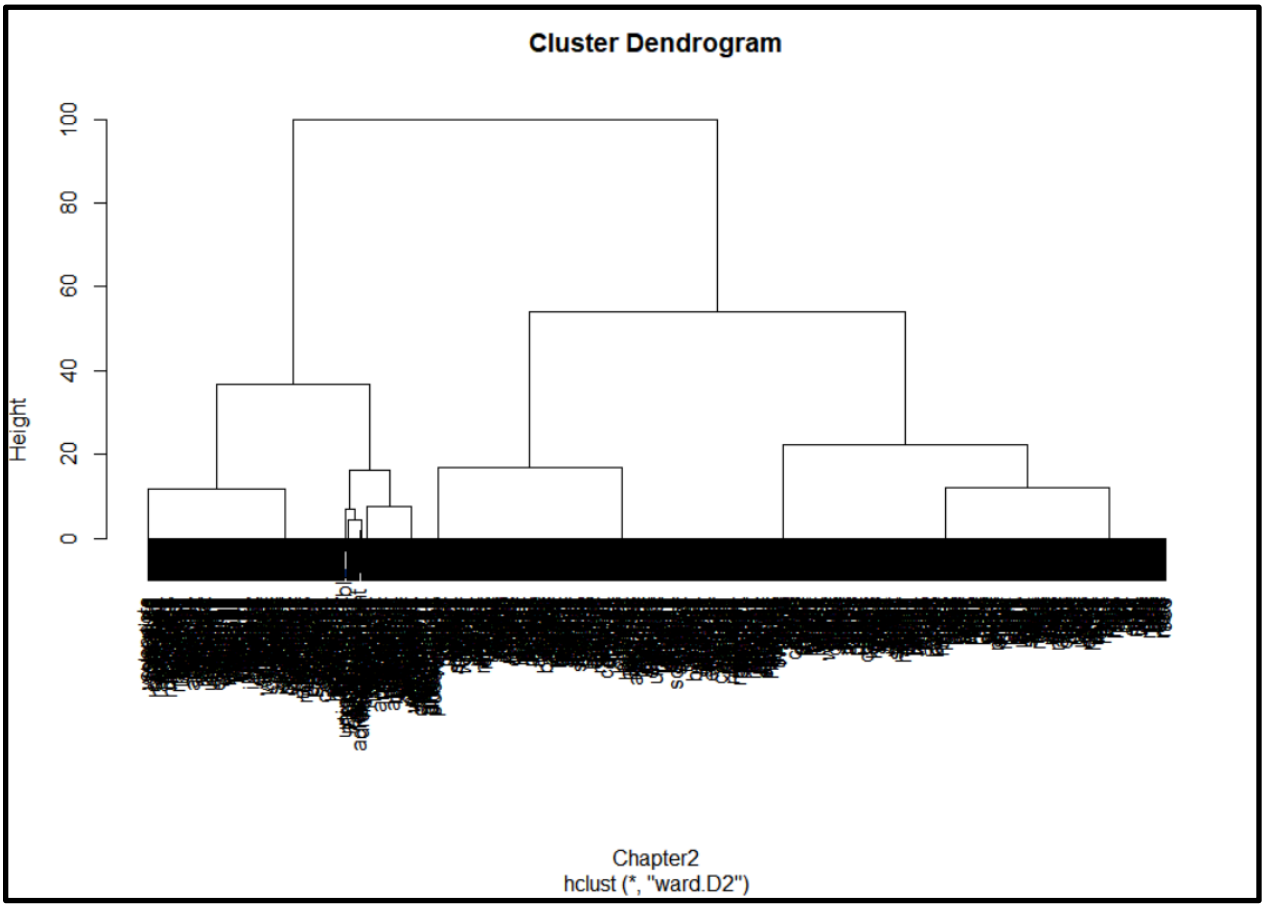
Chapter 1

```
Chapter1 <- dist(a, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
na.omit(Chapter1)
fit <- hclust(Chapter1, method="ward.D2")
plot(fit)
```



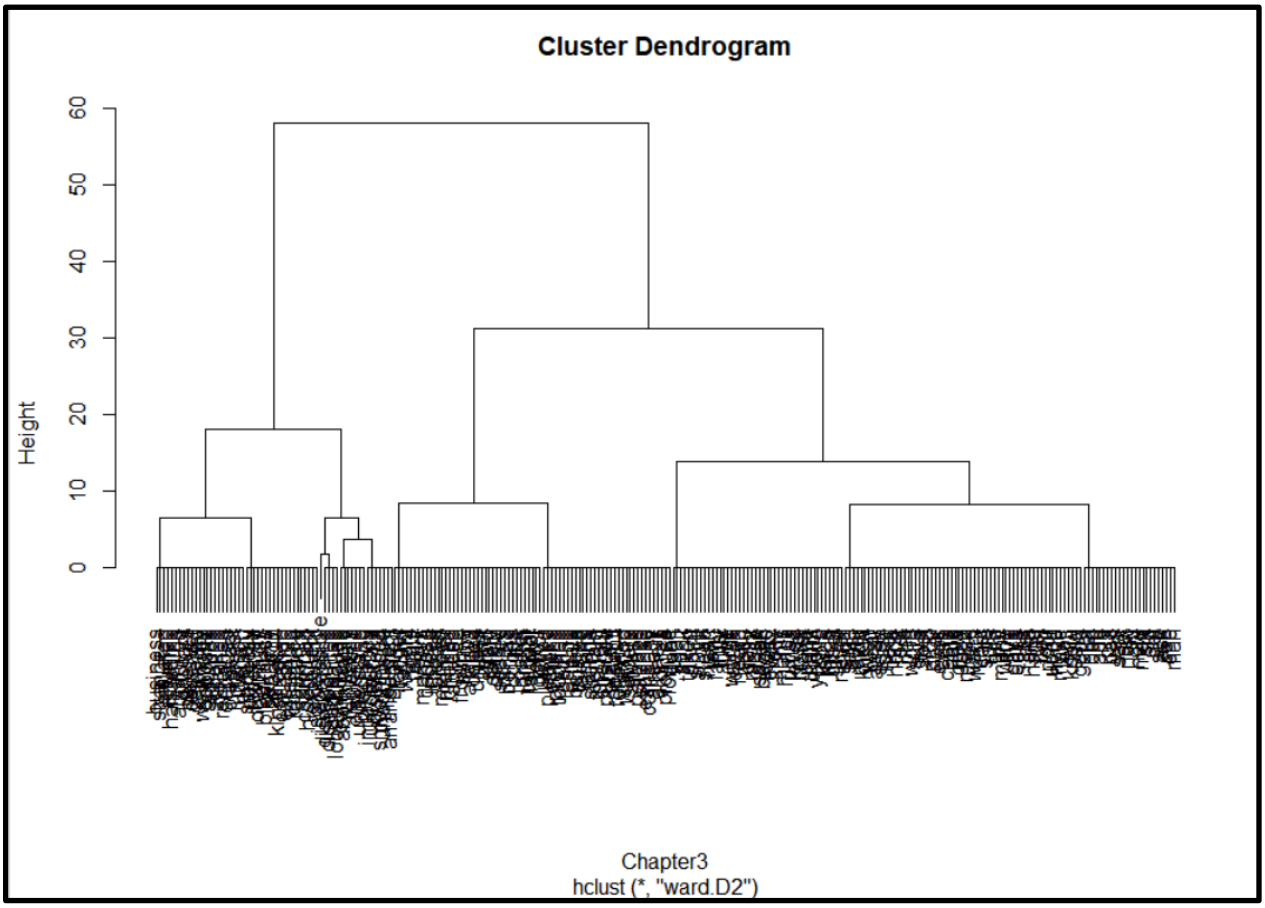
Chapter 2

```
Chapter2 <- dist(b, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
na.omit(Chapter2)
fit <- hclust(Chapter2, method="ward.D2")
plot(fit)
```



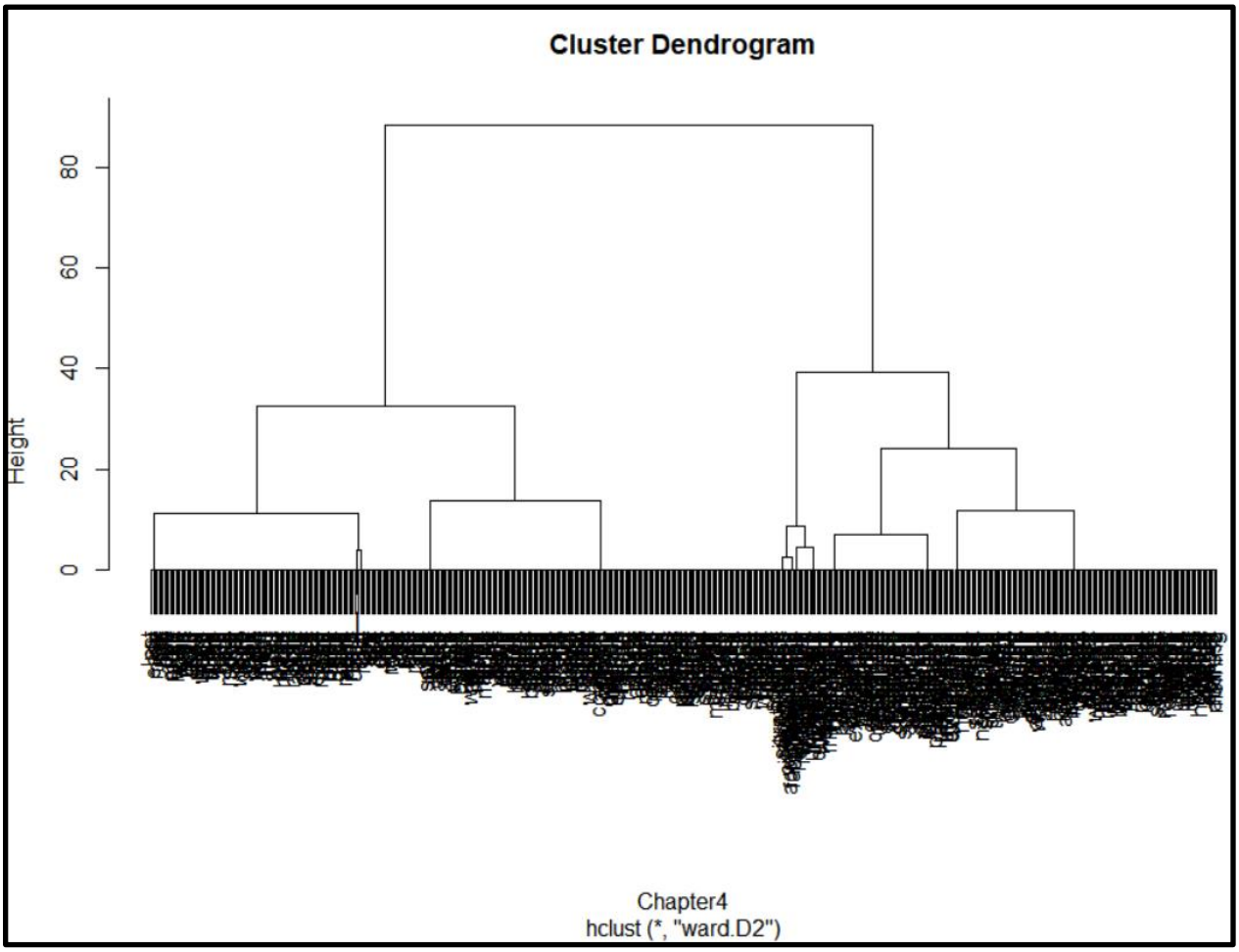
Chapter 3

```
Chapter3 <- dist(c, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
na.omit(Chapter3)
fit <- hclust(Chapter3, method="ward.D2")
plot(fit)
```



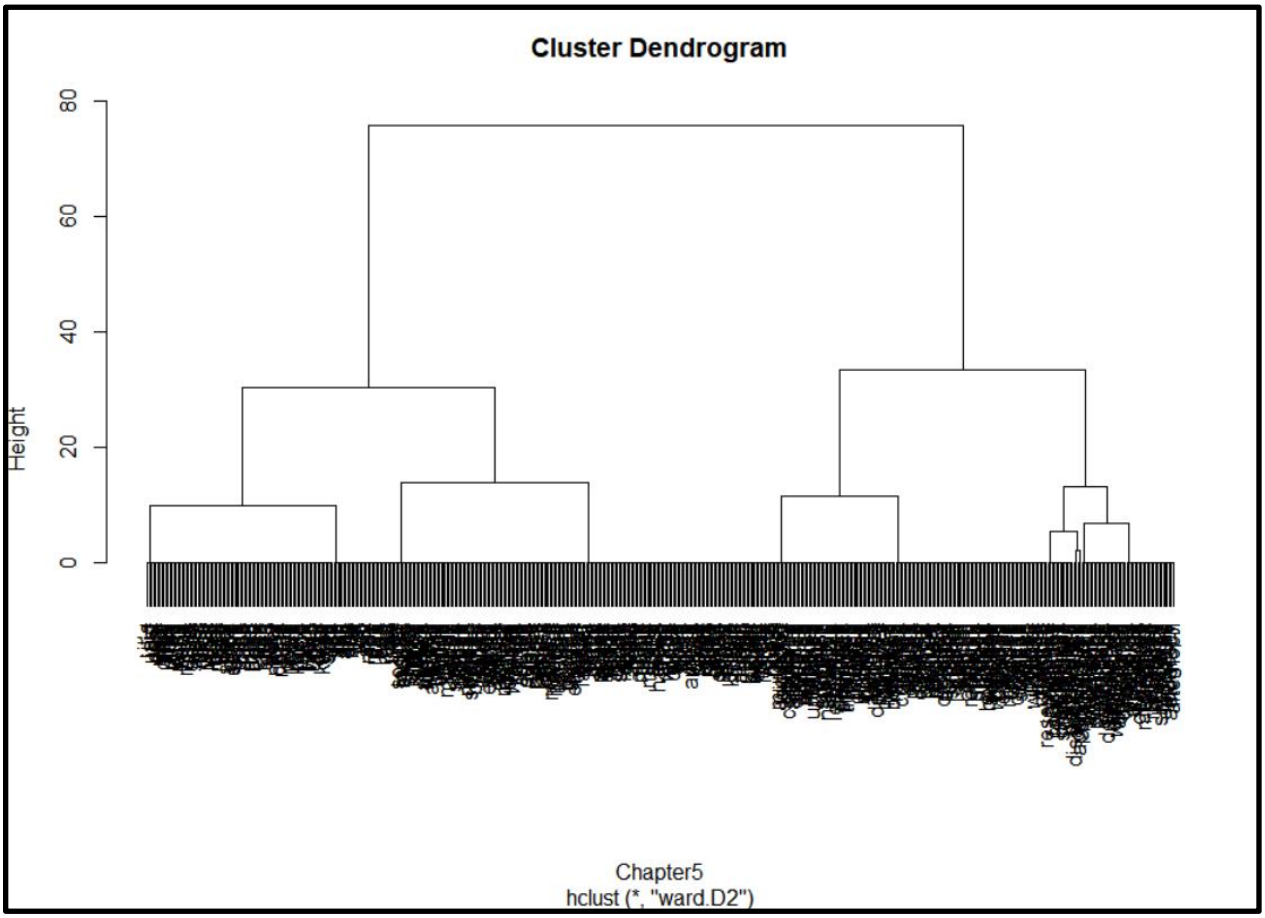
Chapter 4

```
Chapter4 <- dist(d, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter4na <-  na.omit(Chapter4)
fit <- hclust(Chapter4, method="ward.D2")
plot(fit)
```



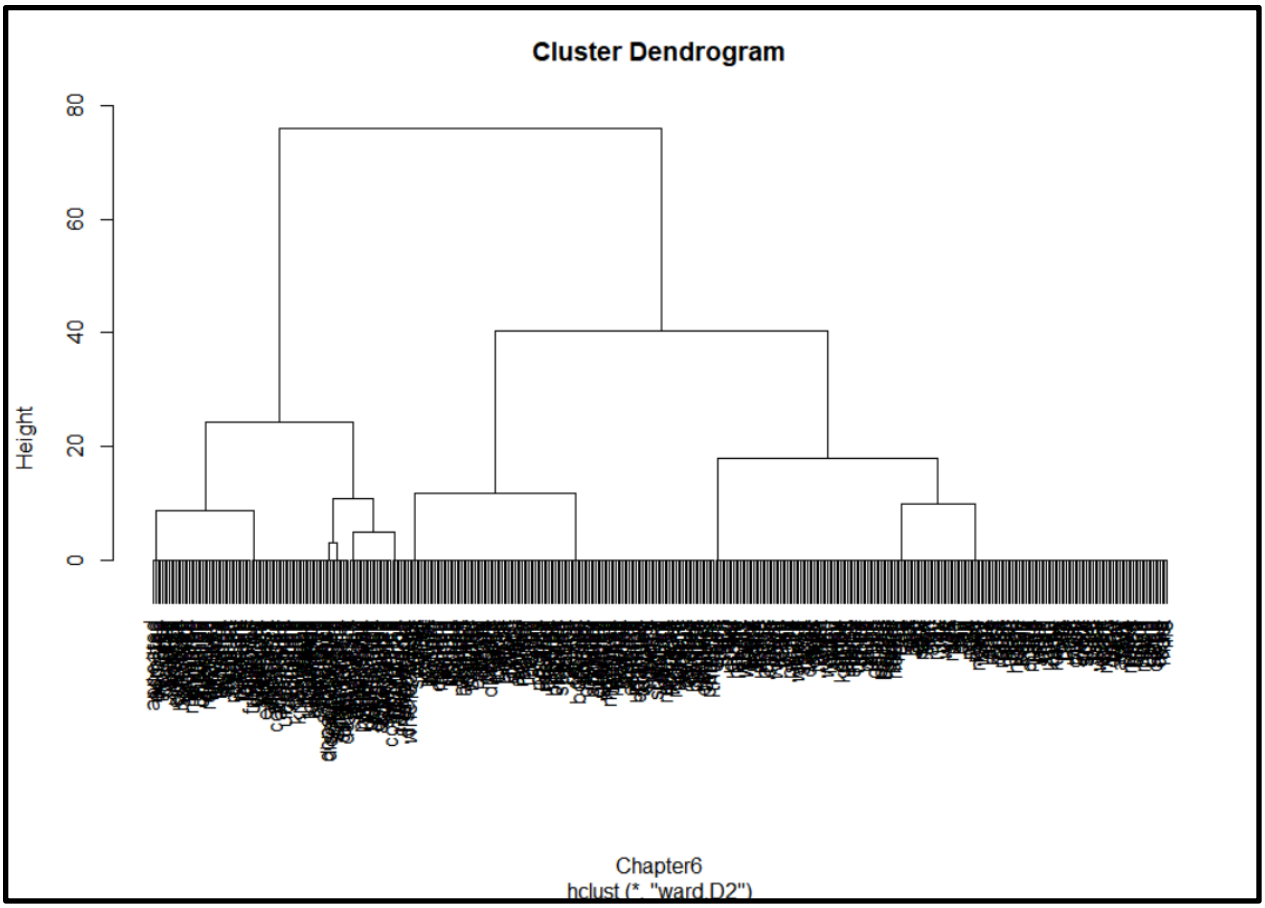
Chapter 5

```
Chapter5 <- dist(e, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter5na <-  na.omit(Chapter5)
fit <- hclust(Chapter5, method="ward.D2")
plot(fit)
```



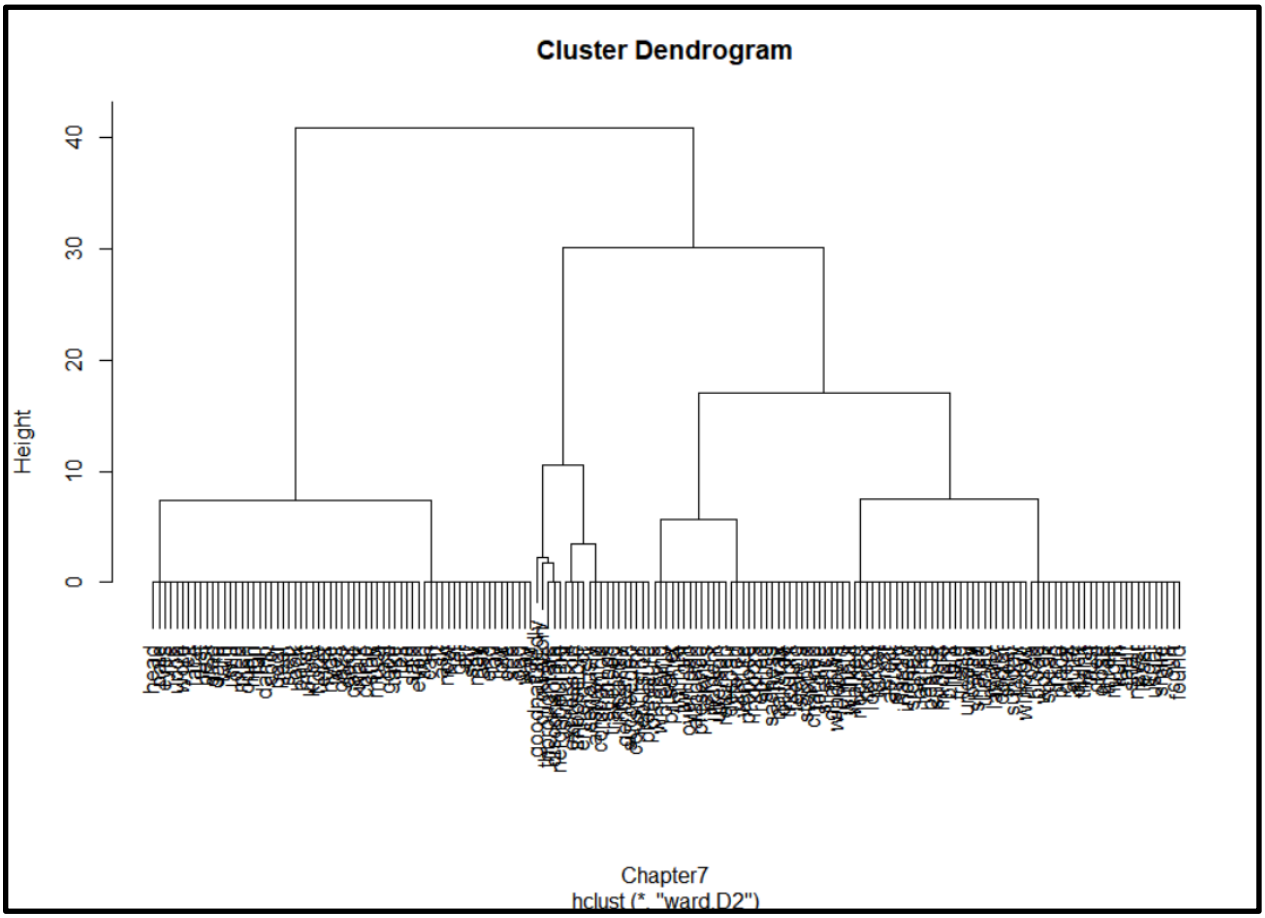
Chapter 6

```
Chapter6 <- dist(f, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter6na <-  na.omit(Chapter6)
fit <- hclust(Chapter6, method="ward.D2")
plot(fit)
```



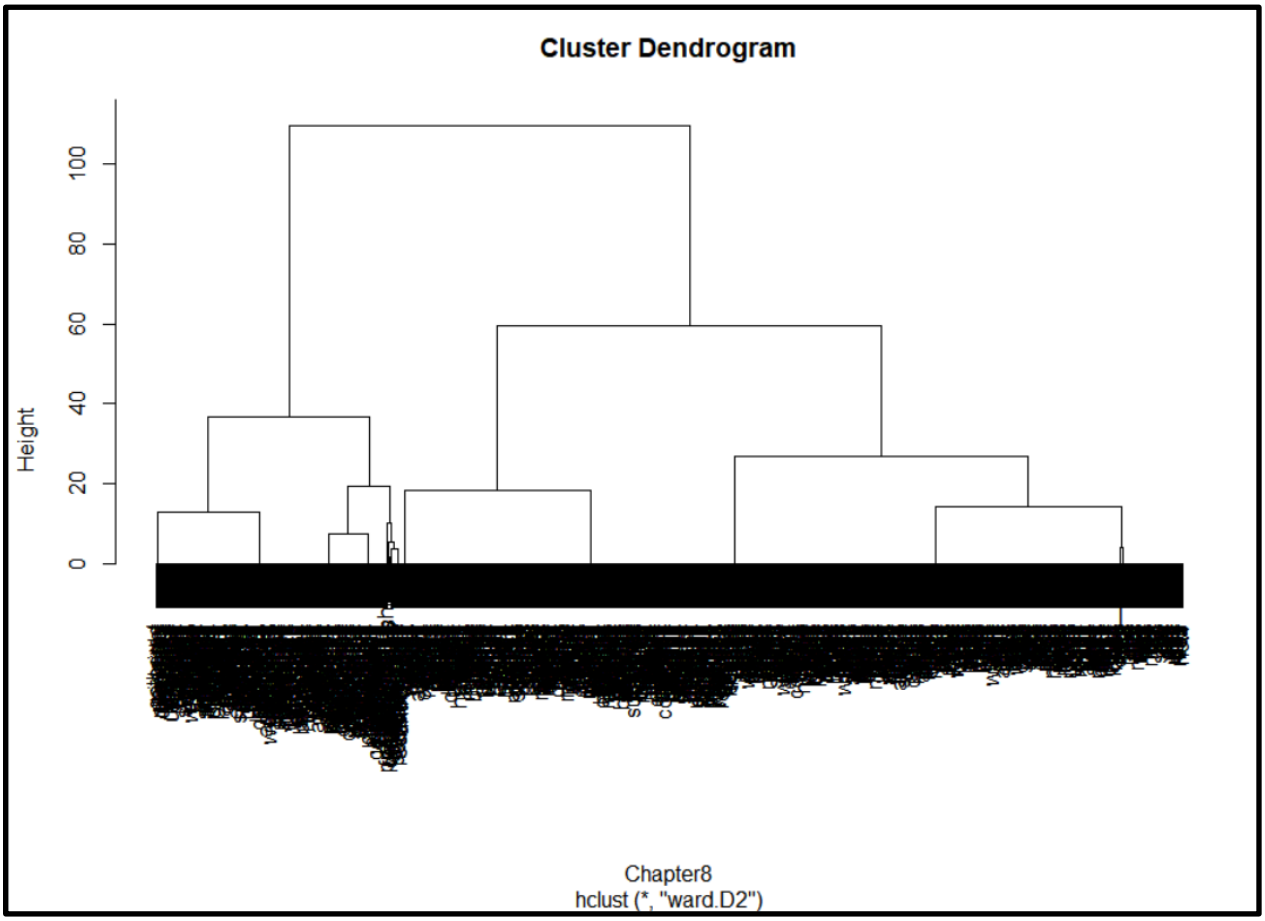
Chapter 7

```
Chapter7 <- dist(g, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter7na <-  na.omit(Chapter7)
fit <- hclust(Chapter7, method="ward.D2")
plot(fit)
```



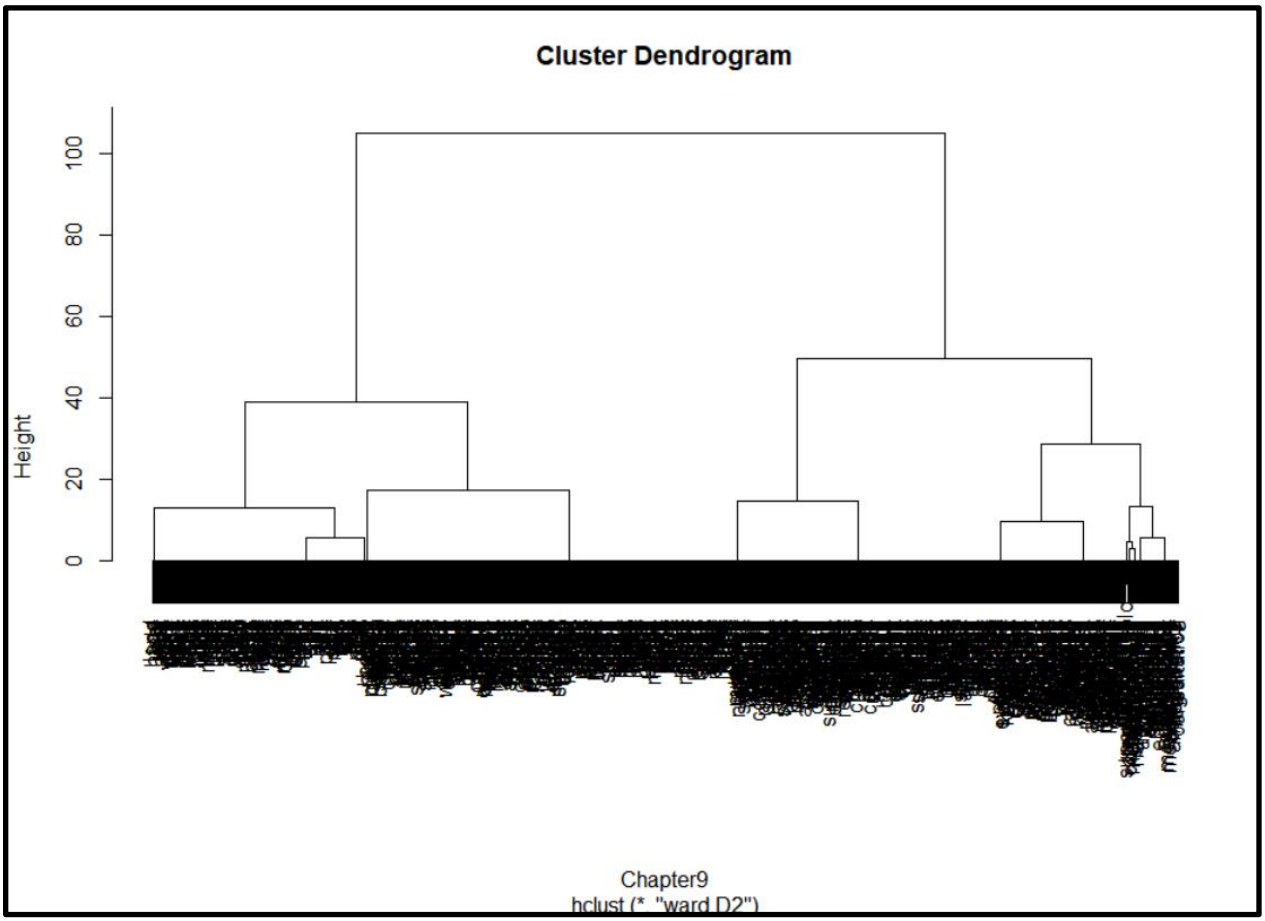
Chapter 8

```
Chapter8 <- dist(h, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter8na <-  na.omit(Chapter8)
fit <- hclust(Chapter8, method="ward.D2")
plot(fit)
```



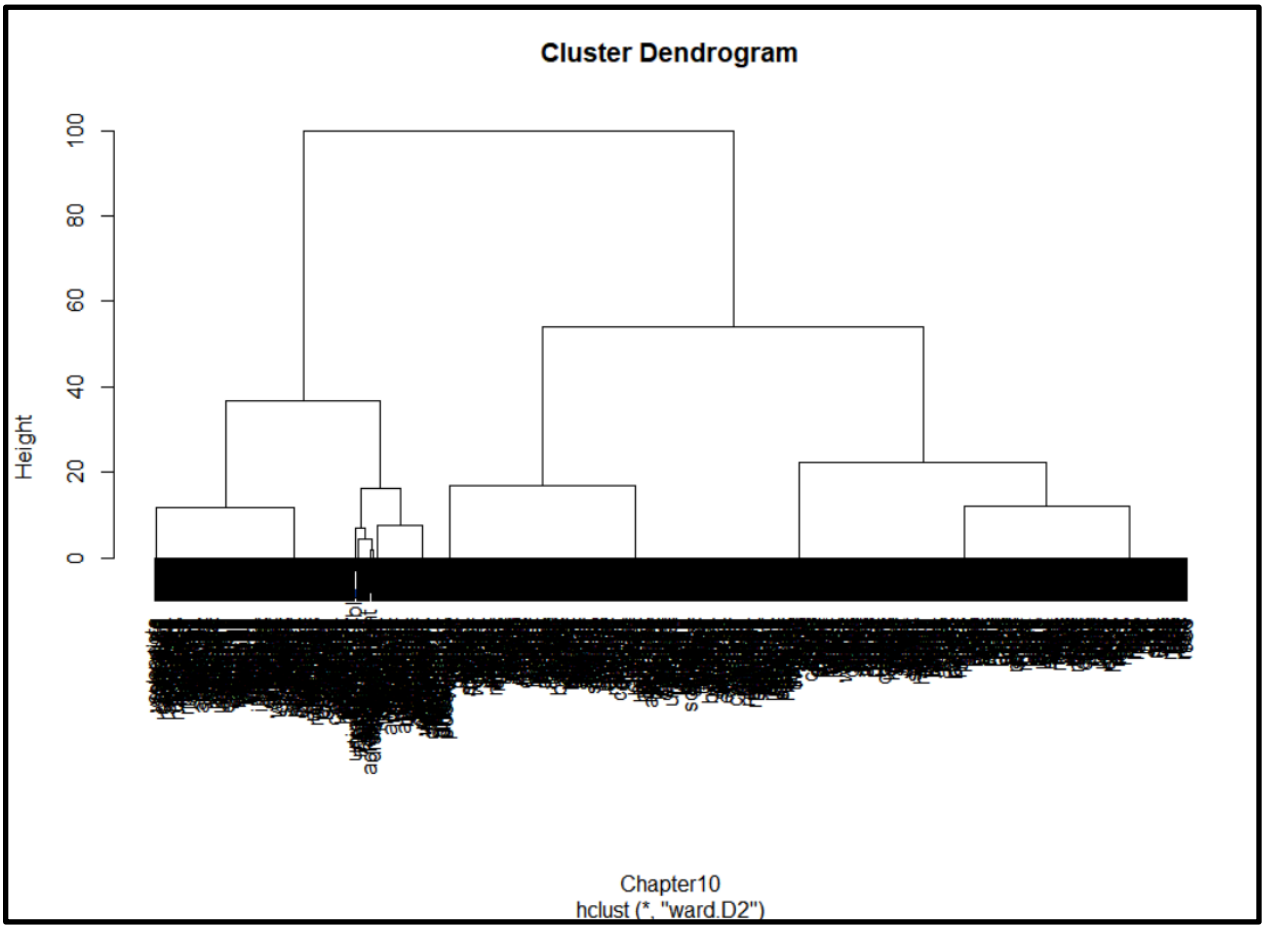
Chapter 9

```
Chapter9 <- dist(i, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter9na <-  na.omit(Chapter9)
fit <- hclust(Chapter9, method="ward.D2")
plot(fit)
```



Chapter 10

```
Chapter10 <- dist(j, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
Chapter10na <-  na.omit(Chapter10)
fit <- hclust(Chapter10, method="ward.D2")
plot(fit)
```



Part d: Find the longest word and longest sentence in each chapter. Print a table of the length of the shortest and longest sentences in each chapter.

Longest Word

For finding the longest word, we convert the clean text into data frame. Dataframe has two columns one being the word and second being the character length of the word. Using the which.max(a$chr), we can find the word with the longest length.

Chapter 1

```
> options(max.print=999999)
> a <- data.frame(names=d1$word,chr=apply(d1,2,nchar)[,1])
> a[which.max(a$chr),]
                           names chr
undemonstrative undemonstrative  15
```

Chapter 2

```
> options(max.print=999999)
> b <- data.frame(names=d2$word,chr=apply(d2,2,nchar)[,1])
> b[which.max(b$chr),]
                          names chr
unimpressionable unimpressionable  16
```

Chapter 3

```
> options(max.print=999999)
> c <- data.frame(names=d3$word,chr=apply(d3,2,nchar)[,1])
> c[which.max(c$chr),]
                        names chr
irrepressible irrepressible  13
```

Chapter 4

```
> d <- data.frame(names=d4$word,chr=apply(d4,2,nchar)[,1])
> d[which.max(d$chr),]
                         names chr
accomplishment accomplishment  14
```

Chapter 5

```
> options(max.print=999999)
> e <- data.frame(names=d5$word,chr=apply(d5,2,nchar)[,1])
> e[which.max(e$chr),]
                       names chr
indifferently indifferently  13
```

Chapter 6

```
> options(max.print=999999)
> f <- data.frame(names=d6$word,chr=apply(d6,2,nchar)[,1])
> f[which.max(f$chr),]
                       names chr
disappearance disappearance  13
```

Chapter 7

```
> options(max.print=999999)
> g <- data.frame(names=d7$word,chr=apply(d7,2,nchar)[,1])
> g[which.max(g$chr),]
                    names chr
disconsolate disconsolate  12
```

Chapter 8

```
> options(max.print=999999)
> h <- data.frame(names=d8$word,chr=apply(d8,2,nchar)[,1])
> h[which.max(h$chr),]
                         names chr
unquestionably unquestionably  14
```

Chapter 9

```
> options(max.print=999999)
> i <- data.frame(names=d9$word,chr=apply(d9,2,nchar)[,1])
> i[which.max(i$chr),]
                       names chr
correspondence correspondence  14
```

Chapter 10

```
> options(max.print=999999)
> j <- data.frame(names=d10$word,chr=apply(d10,2,nchar)[,1])
> j[which.max(j$chr),]
                          names chr
transformations transformations  15
```

Chapter No.	Longest Word	Characters
1	undemonstrative	15
2	unimpressionable	16
3	irrepressible	13
4	accomplishment	14
5	indifferently	13
6	disappearance	13
7	disconsolate	12
8	unquestionably	14
9	correspondence	14
10	transformation	15

Longest Sentence

To find the longest sentence we read the chapter and split it into sentences based on period. Then we remove the extra spaces and find the word count of each sentence in the chapter. Next, we arrange the word count in descending order. The sentence with the highest word count is the longest. The lowest word count that is greater than 1 is the word count of the shortest sentence. Below, is the code for Chapter 1. The same was done for all the other chapters as well.

#Use collapse paste to store the text as one string and then split at period(.).

```
str = paste(Chp1,collapse=" ")
splitstr = strsplit(str, ". ", fixed = TRUE)

str_new = list()

#Remove extra spaces
i = 1
for(s in splitstr){
  s = gsub("^\\s+|\\s+$", "", s)
  str_new[[i]] = s
  i = i + 1
}
str2 = str_new[[1]]

str2 = gsub("  ", " ", str2)

lengthSents = list()
#get length of words for each sentence
for (s in 1:length(str2)) {
  lengthSents[s] = sapply(strsplit(str2[s], " "), length)
}

#Function to sort sentences by number of words in decreasing order
sortnumlist = function(x) {
  n = length(x)
  for (k in n:2) {
    i = 1
    while (i < k) {
      if (x[[i]] < x[[i+1]]) {
        tmp = x[[i+1]]
        x[[i+1]] = x[[i]]
        x[[i]] = tmp
      }
      i = i+1
```

```
    }
  }
  x
}

#Store list of sorted sentences by words.
wordlist = sortnumlist(lengthSents)
wordlist = unique(wordlist)

longestSentence = list()

k = 1

  for (s in 1:length(str2)) {
    if (sapply(strsplit(str2[s], " "), length) == wordlist[1]) {
      longestSentence[k] = str2[s]
      print(wordlist[1])
      k = k + 1
    }
  }

#Shortest sentence with more than 1 word
for (s in length(wordlist):1) {
  if (wordlist[s] > 1) {
    shortestSentence = wordlist[s]
    break;
  }
}
longestSentence
```

Chapter 1

```
> longestSentence
[[1]]
```

[1] "The next thing was to get the money; and where do you think he carried us but to that place with the door?—whipped out a key, went in, and presently came back with the matter of ten pounds in gold and a cheque for the balance on Coutts's, drawn payable to bearer and signed with a name that I can't mention, though it's one of the points of my story, but it was a name at least very well known and often printed"

Chapter 2

> longestSentence
[[1]]
[1] "Fell? or is it the mere radiance of a foul soul that thus transpires through, and transfigures, its clay continent? The last, I think; for, O my poor old Harry Jekyll, if ever I read Satan's signature upon a face, it is on that of your new friend." Round the corner from the by-street, there was a square of ancient, handsome houses, now for the most part decayed from their high estate and let in flats and chambers to all sorts and conditions of men; map-engravers, architects, shady lawyers and the agents of obscure enterprises"

Chapter 3

> longestSentence
[[1]]
[1] "I believe you fully; I would trust you before any man alive, ay, before myself, if I could make the choice; but indeed it isn't what you fancy; it is not as bad as that; and just to put your good heart at rest, I will tell you one thing: the moment I choose, I can be rid of Mr"

Chapter 4

> longestSentence
[[1]]
[1] "As the cab drew up before the address indicated, the fog lifted a little and showed him a dingy street, a gin palace, a low French eating house, a shop for the

retail of penny numbers and twopenny salads, many ragged children huddled in the doorways, and many women of many different nationalities passing out, key in hand, to have a morning glass; and the next moment the fog settled down again upon that part, as brown as umber, and cut him off from his blackguardly surroundings"

Chapter 5

> longestSentence
[[1]]
[1] "It was the first time that the lawyer had been received in that part of his friend’s quarters; and he eyed the dingy, windowless structure with curiosity, and gazed round with a distasteful sense of strangeness as he crossed the theatre, once crowded with eager students and now lying gaunt and silent, the tables laden with chemical apparatus, the floor strewn with crates and littered with packing straw, and the light falling dimly through the foggy cupola"

Chapter 6

> longestSentence
[[1]]
[1] "I could not think that this earth contained a place for sufferings and terrors so unmanning; and you can do but one thing, Utterson, to lighten this destiny, and that is to respect my silence.” Utterson was amazed; the dark influence of Hyde had been withdrawn, the doctor had returned to his old tasks and amities; a week ago, the prospect had smiled with every promise of a cheerful and an honoured age; and now in a moment, friendship, and peace of mind, and the whole tenor of his life were wrecked"

Chapter 7

> longestSentence
[[1]]

[1] "To tell you the truth, I am uneasy about poor Jekyll; and even outside, I feel as if the presence of a friend might do him good.” The court was very cool and a little damp, and full of premature twilight, although the sky, high up overhead, was still bright with sunset"

Chapter 8

> longestSentence
[[1]]
[1] "Your master, Poole, is plainly seized with one of those maladies that both torture and deform the sufferer; hence, for aught I know, the alteration of his voice; hence the mask and the avoidance of his friends; hence his eagerness to find this drug, by means of which the poor soul retains some hope of ultimate recovery—God grant that he be not deceived! There is my explanation; it is sad enough, Poole, ay, and appalling to consider; but it is plain and natural, hangs well together, and delivers us from all exorbitant alarms.” “Sir,” said the butler, turning to a sort of mottled pallor, “that thing was not my master, and there’s the truth"

Chapter 9

> longestSentence
[[1]]
[1] "How could the presence of these articles in my house affect either the honour, the sanity, or the life of my flighty colleague? If his messenger could go to one place, why could he not go to another? And even granting some impediment, why was this gentleman to be received by me in secret? The more I reflected the more convinced I grew that I was dealing with a case of cerebral disease; and though I dismissed my servants to bed, I loaded an old revolver, that I might be found in some posture of self-defence"

Chapter 10

> longestSentence
[[1]]

[1] "It was on the moral side, and in my own person, that I learned to recognise the thorough and primitive duality of man; I saw that, of the two natures that contended in the field of my consciousness, even if I could rightly be said to be either, it was only because I was radically both; and from an early date, even before the course of my scientific discoveries had begun to suggest the most naked possibility of such a miracle, I had learned to dwell with pleasure, as a beloved daydream, on the thought of the separation of these elements"

Length of the shortest and longest sentences in each chapter by word

We stored the length of the longest sentences and shortest sentences for each chapter in a data frame.

```
compData <- data.frame(Longest= numeric(0), Shortest= numeric(0))
compData[1, ] <- c(wordlist[1], shortestSentence)
compData[2, ] <- c(wordlist[1], shortestSentence)
compData[3, ] <- c(wordlist[1], shortestSentence)
compData[4, ] <- c(wordlist[1], shortestSentence)
compData[5, ] <- c(wordlist[1], shortestSentence)
compData[6, ] <- c(wordlist[1], shortestSentence)
compData[7, ] <- c(wordlist[1], shortestSentence)
compData[8, ] <- c(wordlist[1], shortestSentence)
compData[9, ] <- c(wordlist[1], shortestSentence)
compData[10, ] <- c(wordlist[1], shortestSentence)
compData
```

```
> compData
   Longest Shortest
1       83        5
2       95        2
3       60        4
4       87        3
5       77        3
6       92        3
7       52        2
8      114        2
9       96        2
10     101        4
```

Part e: WordNet to mark the parts of speech

```
install.packages("wordnet")
df<-read.delim("C:/Intro to Big Data/Project 3/DrJekyllAndMrHyde.txt")
d2<-paste(unlist(df),collapse=' ')
df2<-gsub(".*CHAPTER I\\s*|CHAPTER II.*","",df2)
df3<-strsplit(df2,      "  ")
doWordnet<-function(w,pos=c("ADJECTIVE","ADVERB","NOUN","VERB")){
          for(x   in pos){
filter<-getTermFilter("ExactMatchFilter",w,TRUE)
terms<-getIndexTerms(x,5,filter)
if(!is.null(terms)){
return(x)
}
}
return("None")
}
for(i in df3){
  for(j in i){
    if( nchar(j) > 4)
    {
    sink("C:/Intro to Big Data/Project 3/output.txt",        append          =        TRUE)
    cat(j)
    cat("          -          ")
    cat(doWordnet(j))
    cat("\n")
    sink()
  }
  }
}
```

Output:-

```

Utterson    -   None
lawyer  -   NOUN
rugged  -   ADJECTIVE
countenance -   NOUN
wasnever    -   None
lighted -   ADJECTIVE
smile;  -   None
cold,   -   None
scanty  -   ADJECTIVE
embarrassed -   ADJECTIVE
discourse;backward  -   None
sentiment;  -   None
lean,   -   None
long,   -   None
dusty,  -   None
dreary  -   ADJECTIVE
somehowlovable. -   None
friendly    -   ADJECTIVE
meetings,   -   None
taste,something -   None
eminently   -   ADVERB
human   -   ADJECTIVE
beaconed    -   None
something   -   NOUN
indeed  -   ADVERB
whichnever  -   None
```

Part f: Analyze word frequency using functions from package zipfR

ItaRi.tfl<-read.tfl("C:\Users\aneri\OneDrive\Documents\RProject3\Projec3\Text Document.txt")
ItaUltra.tfl<-read.tfl("C:\Users\aneri\OneDrive\Documents\RProject3\Projec3\Text Document.txt")
ItaRi2.tfl<-read.tfl("C:\Users\aneri\OneDrive\Documents\RProject3\Projec3\Text Document.txt")

ItaRi.spc<-tfl2spc(ItaRi.tfl)
ItaUltra.spc<-tfl2spc(ItaUltra.tfl)
ItaRi2.spc<- tfl2spc(ItaRi2.tfl)

```
> ItaRi.spc
    m  Vm
1   1 346
2   2 105
3   3  74
4   4  43
5   5  39
6   6  25
7   7  27
8   8  15
9   9  17
10 10   9
        ...

       N    V
 1399898 1098
```

summary(ItaRi.spc)

```
> summary(ItaRi.spc)
zipfR object for frequency spectrum
Sample size:     N  = 1399898
Vocabulary size: V  = 1098
Class sizes:     Vm = 346 105 74 43 39 25 27 15 ...
```

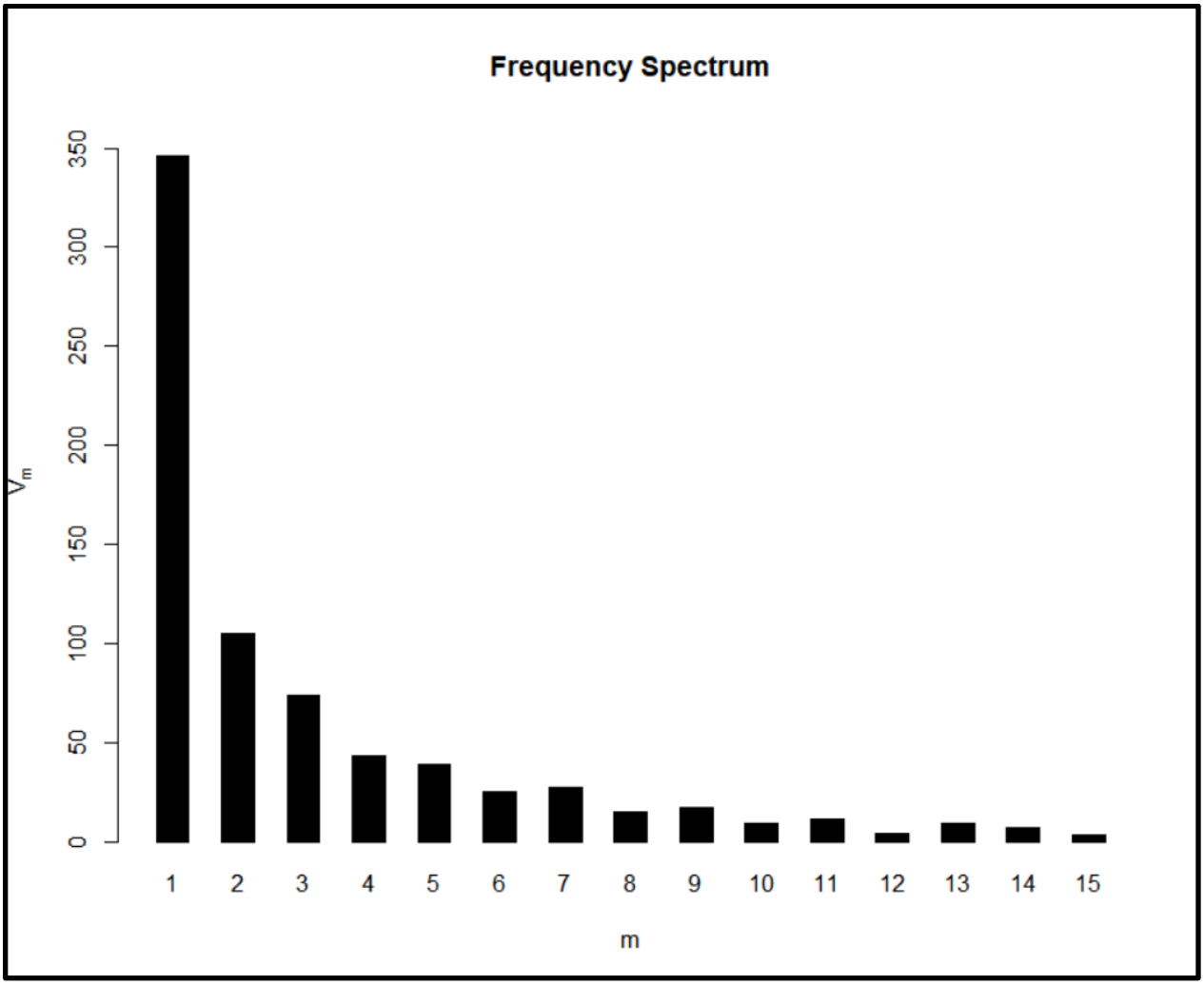
N(ItaRi.spc)
V(ItaRi.spc)

```
> N(ItaRi.spc)
[1] 1399898
> V(ItaRi.spc)
[1] 1098
```

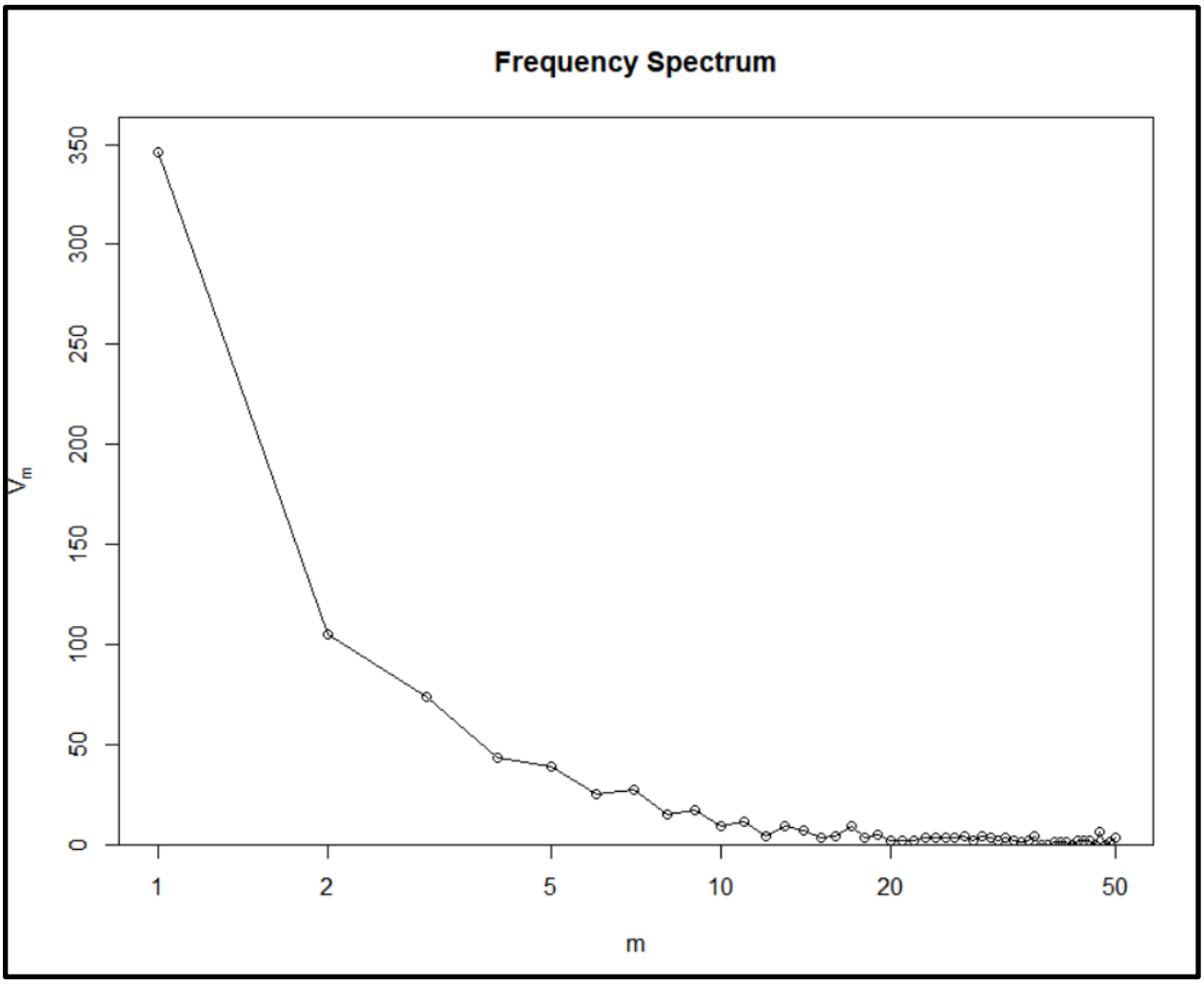
Vm(ItaRi.spc, 1) / N(ItaRi.spc)

```
> Vm(ItaRi.spc, 1:5)
[1] 346 105  74  43  39
```

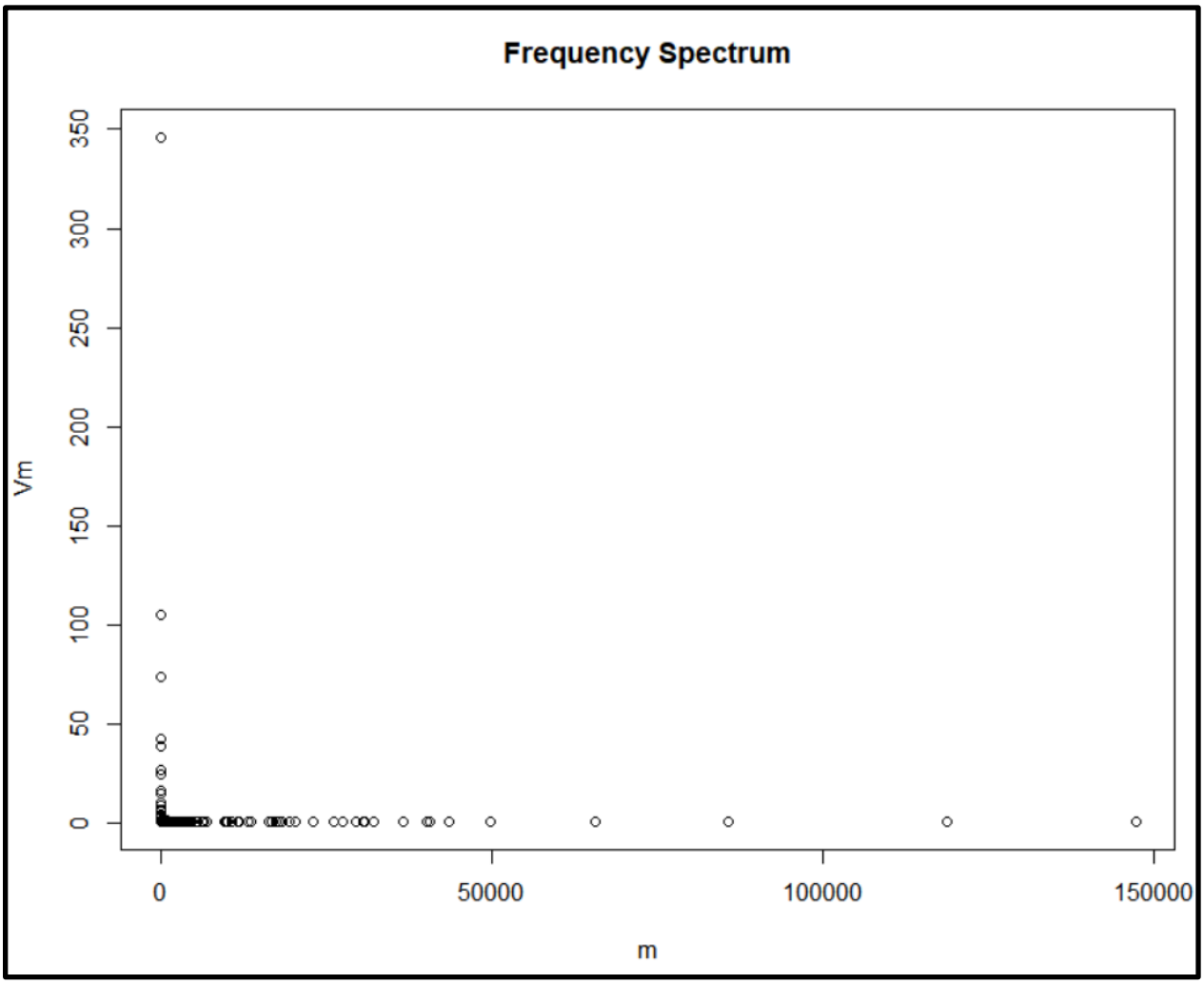
plot(ItaRi.spc)



plot(ItaRi.spc, log="x")



```
with(ItaRi.spc, plot(m, Vm, main="Frequency Spectrum"))
```



Part g: Bigrams and Trigrams for all words whose length is greater than 6 characters in Chapter 1

For the Bigrams and Trigrams for all words with length greater than 6 characters in chapter 1, we start with scan the text in chapter 1. Next, we install the package tau for further analysis. Next, textcnt does the counting and converting the text into lowercase. Then, we convert the vector into data frame.

After converting it to the dataframe, then we remove punctuations, remove the stop words, stem the words, and convert the text into lowercase. Using the tm_map, we convert the corpus text into clean text. Then, nchar is used to include the character lengths greater than 6.The vapply function is used to calculate the bigrams and trigrams for the chapter 1

```
textlines = readLines(file.choose())
text = scan("C://Users//aneri//OneDrive//Documents//RProject3//Projec3//Text
Document//Chapter 1.txt", quote=NULL, what="x")
head(text)
counts = as.data.frame(xtabs(~text))
#installed tau
chapter1.str = paste(text, collapse=" ")
# this does the counting, lowercasing everything first
chapter1.counts = textcnt(chapter1.str, n=1, method="string", tolower=T)
# chapter1.counts is a vector with names on the entries.
# Here is how you access entries:
names(chapter1.counts)
chapter1.counts.df = data.frame(word = names(chapter1.counts), count =
c(chapter1.counts))

chapter1.counts.df[chapter1.counts.df$word == "chapter1",]
#tm library
chapter1 <- Corpus(VectorSource(textlines))
```

```
# normalization of the text:
chapter1 <- tm_map(chapter1, tolower) #lowercase
chapter1 <- tm_map(chapter1, removePunctuation, preserve_intra_word_dashes =
FALSE) # remove punctuation
chapter1 <- tm_map(chapter1, removeWords, stopwords("english")) # remove
stopwords
chapter1 <- tm_map(chapter1, stemDocument) # reduce word forms to stems
chapter1.tdm.1 <- TermDocumentMatrix(chapter1[1])
findFreqTerms(oz.tdm.1, 100)
chapter1.tdm.2 <- TermDocumentMatrix(chapter1[2])
findFreqTerms(chapter1.tdm.2, 50)

# count how often the term appears in each of the documents in the collection
tdm = TermDocumentMatrix(chapter1)
chapter1.str = paste(text, collapse = " ")
chapter1.corpus = Corpus(VectorSource(chapter1.str))
chapter1.corpus = tm_map(chapter1.corpus, tolower)
chapter1.corpus = tm_map(chapter1.corpus, removePunctuation,
preserve_intra_word_dashes = FALSE)
cleaned.chapter1.str = as.character(chapter1.corpus)[1]
chapter1.words = strsplit(cleaned.chapter1.str, " ", fixed = T)[[1]]
a=chapter1.words[nchar(chapter1.words)>6]
chapter1.bigrams = vapply(ngrams(a, 2), paste, "", collapse = " ")
chapter1.Trigrams = vapply(ngrams(a, 3), paste, "", collapse = " ")
chapter1.bigrams
chapter1.Trigrams
```

Bigrams for Chapter 1

```
 [1] "utterson countenance"          "countenance lighted"
 [3] "lighted embarrassed"           "embarrassed discourse"
 [5] "discourse backward"            "backward sentiment"
 [7] "sentiment somehow"             "somehow lovable"
 [9] "lovable friendly"              "friendly meetings"
[11] "meetings something"            "something eminently"
[13] "eminently beaconed"            "beaconed something"
[15] "something symbols"             "symbols afterdinner"
[17] "afterdinner austere"           "austere himself"
[19] "himself mortify"               "mortify vintages"
[21] "vintages enjoyed"              "enjoyed theatre"
[23] "theatre crossed"               "crossed approved"
[25] "approved tolerance"            "tolerance sometimes"
[27] "sometimes wondering"           "wondering pressure"
[29] "pressure spirits"              "spirits involved"
[31] "involved misdeeds"             "misdeeds extremity"
[33] "extremity inclined"            "inclined reprove"
[35] "reprove incline"               "incline quaintly"
[37] "quaintly brother"              "brother character"
[39] "character frequently"          "frequently fortune"
[41] "fortune reputable"             "reputable acquaintance"
[43] "acquaintance influence"        "influence downgoing"
[45] "downgoing chambers"            "chambers demeanour"
[47] "demeanour utterson"            "utterson undemonstrative"
[49] "undemonstrative friendship"    "friendship founded"
[51] "founded similar"               "similar catholicity"
[53] "catholicity goodnature"        "goodnature friendly"
[55] "friendly readymade"            "readymade opportunity"
[57] "opportunity lawyers"           "lawyers friends"
[59] "friends longest"               "longest affections"
[61] "affections implied"            "implied aptness"
[63] "aptness richard"               "richard enfield"
[65] "enfield distant"               "distant kinsman"
```

```
 [45] "downgoing chambers"          "chambers demeanour"
 [47] "demeanour utterson"          "utterson undemonstrative"
 [49] "undemonstrative friendship"  "friendship founded"
 [51] "founded similar"             "similar catholicity"
 [53] "catholicity goodnature"      "goodnature friendly"
 [55] "friendly readymade"          "readymade opportunity"
 [57] "opportunity lawyers"         "lawyers friends"
 [59] "friends longest"             "longest affections"
 [61] "affections implied"          "implied aptness"
 [63] "aptness richard"             "richard enfield"
 [65] "enfield distant"             "distant kinsman"
 [67] "kinsman wellknown"           "wellknown subject"
 [69] "subject reported"            "reported encountered"
 [71] "encountered nothing"         "nothing singularly"
 [73] "singularly obvious"          "obvious appearance"
 [75] "appearance greatest"         "greatest excursions"
 [77] "excursions counted"          "counted occasions"
 [79] "occasions pleasure"          "pleasure resisted"
 [81] "resisted business"           "business uninterrupted"
 [83] "uninterrupted chanced"       "chanced rambles"
 [85] "rambles bystreet"            "bystreet quarter"
 [87] "quarter thriving"            "thriving weekdays"
 [89] "weekdays inhabitants"        "inhabitants emulously"
 [91] "emulously surplus"           "surplus coquetry"
 [93] "coquetry thoroughfare"       "thoroughfare invitation"
 [95] "invitation smiling"          "smiling saleswomen"
 [97] "saleswomen comparatively"    "comparatively passage"
 [99] "passage contrast"            "contrast neighbourhood"
[101] "neighbourhood freshly"       "freshly painted"
[103] "painted shutters"            "shutters wellpolished"
[105] "wellpolished brasses"        "brasses general"
[107] "general cleanliness"         "cleanliness instantly"
[109] "instantly pleased"           "pleased passenger"
[111] "passenger certain"           "certain sinister"
[113] "sinister building"           "building forward"
[115] "forward storeys"             "storeys nothing"
[117] "nothing forehead"            "forehead discoloured"
[119] "discoloured feature"         "feature prolonged"
[121] "prolonged negligence"        "negligence equipped"
[123] "equipped neither"            "neither knocker"
```

```
 [95] "invitation smiling"          "smiling saleswomen"
 [97] "saleswomen comparatively"    "comparatively passage"
 [99] "passage contrast"            "contrast neighbourhood"
[101] "neighbourhood freshly"       "freshly painted"
[103] "painted shutters"            "shutters wellpolished"
[105] "wellpolished brasses"        "brasses general"
[107] "general cleanliness"         "cleanliness instantly"
[109] "instantly pleased"           "pleased passenger"
[111] "passenger certain"           "certain sinister"
[113] "sinister building"           "building forward"
[115] "forward storeys"             "storeys nothing"
[117] "nothing forehead"            "forehead discoloured"
[119] "discoloured feature"         "feature prolonged"
[121] "prolonged negligence"        "negligence equipped"
[123] "equipped neither"            "neither knocker"
[125] "knocker blistered"           "blistered distained"
[127] "distained slouched"          "slouched matches"
[129] "matches children"            "children schoolboy"
[131] "schoolboy mouldings"         "mouldings generation"
[133] "generation appeared"         "appeared visitors"
[135] "visitors ravages"            "ravages enfield"
[137] "enfield bystreet"            "bystreet abreast"
[139] "abreast pointed"             "pointed companion"
[141] "companion replied"           "replied affirmative"
[143] "affirmative connected"       "connected utterson"
[145] "utterson returned"           "returned enfield"
[147] "enfield morning"             "morning through"
[149] "through literally"           "literally nothing"
[151] "nothing asleep"street"       "asleep"street lighted"
[153] "lighted procession"          "procession church"till"
[155] "church"till listens"         "listens listens"
[157] "listens policeman"           "policeman figures"
[159] "figures stumping"            "stumping eastward"
[161] "eastward running"            "running another"
[163] "another naturally"           "naturally horrible"
[165] "horrible trampled"           "trampled screaming"
[167] "screaming nothing"           "nothing hellish"
[169] "hellish juggernaut"          "juggernaut collared"
[171] "collared gentleman"          "gentleman brought"
[173] "brought already"             "already screaming"
```

```
[145] "utterson returned"                "returned enfield"
[147] "enfield morning"                  "morning through"
[149] "through literally"                "literally nothing"
[151] "nothing asleep"street"            "asleep"street lighted"
[153] "lighted procession"               "procession church"till"
[155] "church"till listens"              "listens listens"
[157] "listens policeman"                "policeman figures"
[159] "figures stumping"                 "stumping eastward"
[161] "eastward running"                 "running another"
[163] "another naturally"                "naturally horrible"
[165] "horrible trampled"                "trampled screaming"
[167] "screaming nothing"                "nothing hellish"
[169] "hellish juggernaut"               "juggernaut collared"
[171] "collared gentleman"               "gentleman brought"
[173] "brought already"                  "already screaming"
[175] "screaming perfectly"              "perfectly resistance"
[177] "resistance brought"               "brought running"
[179] "running appearance"               "appearance frightened"
[181] "frightened according"             "according sawbones"
[183] "sawbones supposed"                "supposed curious"
[185] "curious circumstance"             "circumstance loathing"
[187] "loathing gentleman"               "gentleman natural"
[189] "natural doctors"                  "doctors apothecary"
[191] "apothecary particular"            "particular edinburgh"
[193] "edinburgh emotional"              "emotional bagpipe"
[195] "bagpipe prisoner"                 "prisoner sawbones"
[197] "sawbones killing"                 "killing question"
[199] "question scandal"                 "scandal friends"
[201] "friends undertook"                "undertook pitching"
[203] "pitching keeping"                 "keeping harpies"
[205] "harpies hateful"                  "hateful sneering"
[207] "sneering coolness"frightened"     "coolness"frightened that"but"
[209] "that"but carrying"                "carrying capital"
[211] "capital accidentsaid"             "accidentsaid naturally"
[213] "naturally helpless"               "helpless gentleman"
[215] "gentleman screwed"                "screwed hundred"
[217] "hundred clearly"                  "clearly something"
[219] "something mischief"               "mischief carried"
[221] "carried "whipped"                 ""whipped presently"
[223] "presently balance"                "balance couttss"
[225] "couttss payable"                  "payable mention"
```

```
[193] "edinburgh emotional"          "emotional bagpipe"
[195] "bagpipe prisoner"             "prisoner sawbones"
[197] "sawbones killing"             "killing question"
[199] "question scandal"             "scandal friends"
[201] "friends undertook"            "undertook pitching"
[203] "pitching keeping"             "keeping harpies"
[205] "harpies hateful"              "hateful sneering"
[207] "sneering coolness"frightened" "coolness"frightened that"but"
[209] "that"but carrying"            "carrying capital"
[211] "capital accidentsaid"         "accidentsaid naturally"
[213] "naturally helpless"           "helpless gentleman"
[215] "gentleman screwed"            "screwed hundred"
[217] "hundred clearly"              "clearly something"
[219] "something mischief"           "mischief carried"
[221] "carried "whipped"             ""whipped presently"
[223] "presently balance"            "balance couttss"
[225] "couttss payable"              "payable mention"
[227] "mention printed"              "printed signature"
[229] "signature genuine"            "genuine liberty"
[231] "liberty pointing"             "pointing gentleman"
[233] "gentleman business"           "business apocryphal"
[235] "apocryphal morning"           "morning another"
[237] "another hundred"              "hundred sneering"
[239] "sneering chambers"            "chambers breakfasted"
[241] "breakfasted believe"          "believe forgery"
[243] "forgery genuine"              "genuine utterson"
[245] "utterson enfield"             "enfield damnable"
[247] "damnable proprieties"         "proprieties celebrated"
[249] "celebrated fellows"           "fellows blackmail"
[251] "blackmail suppose"            "suppose through"
[253] "through consequence"          "consequence explaining"
[255] "explaining recalled"          "recalled utterson"
[257] "utterson suddenly"            "suddenly returned"
[259] "returned enfield"             "enfield noticed"
[261] "noticed address"              "address utterson"
[263] "utterson delicacy"            "delicacy strongly"
[265] "strongly putting"             "putting questions"
[267] "questions partakes"           "partakes judgment"
[269] "judgment question"            "question starting"
[271] "starting quietly"             "quietly starting"
```

```
Console ~/RProject3/Projec3/
[247] "damnable proprieties"        "proprieties celebrated"
[249] "celebrated fellows"          "fellows blackmail"
[251] "blackmail suppose"           "suppose through"
[253] "through consequence"         "consequence explaining"
[255] "explaining recalled"         "recalled utterson"
[257] "utterson suddenly"           "suddenly returned"
[259] "returned enfield"            "enfield noticed"
[261] "noticed address"             "address utterson"
[263] "utterson delicacy"           "delicacy strongly"
[265] "strongly putting"            "putting questions"
[267] "questions partakes"          "partakes judgment"
[269] "judgment question"           "question starting"
[271] "starting quietly"            "quietly starting"
[273] "starting presently"          "presently thought"
[275] "thought knocked"             "knocked studied"
[277] "studied continued"           "continued enfield"
[279] "enfield scarcely"            "scarcely gentleman"
[281] "gentleman adventure"         "adventure windows"
[283] "windows looking"             "looking windows"
[285] "windows chimney"             "chimney generally"
[287] "generally smoking"           "smoking somebody"
[289] "somebody buildings"          "buildings together"
[291] "together another"            "another silence"
[293] "silence enfield"             "enfield utterson"
[295] "utterson returned"           "returned enfield"
[297] "enfield continued"           "continued enfield"
[299] "enfield utterson"            "utterson describe"
[301] "describe something"          "something appearance"
[303] "appearance something"        "something displeasing"
[305] "displeasing something"       "something downright"
[307] "downright detestable"        "detestable disliked"
[309] "disliked deformed"           "deformed somewhere"
[311] "somewhere feeling"           "feeling deformity"
[313] "deformity although"          "although couldnt"
[315] "couldnt specify"             "specify extraordinary"
[317] "extraordinary looking"       "looking nothing"
[319] "nothing describe"            "describe declare"
[321] "declare utterson"            "utterson silence"
[323] "silence obviously"           "obviously consideration"
[325] "consideration inquired"      "inquired enfield"
```

```
[269] "judgment question"          "question starting"
[271] "starting quietly"           "quietly starting"
[273] "starting presently"         "presently thought"
[275] "thought knocked"            "knocked studied"
[277] "studied continued"          "continued enfield"
[279] "enfield scarcely"           "scarcely gentleman"
[281] "gentleman adventure"        "adventure windows"
[283] "windows looking"            "looking windows"
[285] "windows chimney"            "chimney generally"
[287] "generally smoking"          "smoking somebody"
[289] "somebody buildings"         "buildings together"
[291] "together another"           "another silence"
[293] "silence enfield"            "enfield utterson"
[295] "utterson returned"          "returned enfield"
[297] "enfield continued"          "continued enfield"
[299] "enfield utterson"           "utterson describe"
[301] "describe something"         "something appearance"
[303] "appearance something"       "something displeasing"
[305] "displeasing something"      "something downright"
[307] "downright detestable"       "detestable disliked"
[309] "disliked deformed"          "deformed somewhere"
[311] "somewhere feeling"          "feeling deformity"
[313] "deformity although"         "although couldnt"
[315] "couldnt specify"            "specify extraordinary"
[317] "extraordinary looking"      "looking nothing"
[319] "nothing describe"           "describe declare"
[321] "declare utterson"           "utterson silence"
[323] "silence obviously"          "obviously consideration"
[325] "consideration inquired"     "inquired enfield"
[327] "enfield surprised"          "surprised himself"
[329] "himself utterson"           "utterson strange"
[331] "strange because"            "because already"
[333] "already richard"            "richard inexact"
[335] "inexact correct"            "correct returned"
[337] "returned sullenness"        "sullenness pedantically"
[339] "pedantically utterson"      "utterson presently"
[341] "presently resumed"          "resumed another"
[343] "another nothing"            "nothing ashamed"
[345] "ashamed bargain"            "bargain richard"
```

Trigrams for Chapter 1

```
 [1] "utterson countenance lighted"          "countenance lighted embarrassed"
 [3] "lighted embarrassed discourse"         "embarrassed discourse backward"
 [5] "discourse backward sentiment"          "backward sentiment somehow"
 [7] "sentiment somehow lovable"             "somehow lovable friendly"
 [9] "lovable friendly meetings"             "friendly meetings something"
[11] "meetings something eminently"          "something eminently beaconed"
[13] "eminently beaconed something"          "beaconed something symbols"
[15] "something symbols afterdinner"         "symbols afterdinner austere"
[17] "afterdinner austere himself"           "austere himself mortify"
[19] "himself mortify vintages"              "mortify vintages enjoyed"
[21] "vintages enjoyed theatre"              "enjoyed theatre crossed"
[23] "theatre crossed approved"              "crossed approved tolerance"
[25] "approved tolerance sometimes"          "tolerance sometimes wondering"
[27] "sometimes wondering pressure"          "wondering pressure spirits"
[29] "pressure spirits involved"             "spirits involved misdeeds"
[31] "involved misdeeds extremity"           "misdeeds extremity inclined"
[33] "extremity inclined reprove"            "inclined reprove incline"
[35] "reprove incline quaintly"              "incline quaintly brother"
[37] "quaintly brother character"            "brother character frequently"
[39] "character frequently fortune"          "frequently fortune reputable"
[41] "fortune reputable acquaintance"        "reputable acquaintance influence"
[43] "acquaintance influence downgoing"      "influence downgoing chambers"
[45] "downgoing chambers demeanour"          "chambers demeanour utterson"
[47] "demeanour utterson undemonstrative"    "utterson undemonstrative friendship"
[49] "undemonstrative friendship founded"    "friendship founded similar"
[51] "founded similar catholicity"           "similar catholicity goodnature"
[53] "catholicity goodnature friendly"       "goodnature friendly readymade"
[55] "friendly readymade opportunity"        "readymade opportunity lawyers"
[57] "opportunity lawyers friends"           "lawyers friends longest"
[59] "friends longest affections"            "longest affections implied"
[61] "affections implied aptness"            "implied aptness richard"
[63] "aptness richard enfield"               "richard enfield distant"
[65] "enfield distant kinsman"               "distant kinsman wellknown"
[67] "kinsman wellknown subject"             "wellknown subject reported"
[69] "subject reported encountered"          "reported encountered nothing"
[71] "encountered nothing singularly"        "nothing singularly obvious"
[73] "singularly obvious appearance"         "obvious appearance greatest"
[75] "appearance greatest excursions"        "greatest excursions counted"
[77] "excursions counted occasions"          "counted occasions pleasure"
```

```
 [49] "undemonstrative friendship founded"    "friendship founded similar"
 [51] "founded similar catholicity"           "similar catholicity goodnature"
 [53] "catholicity goodnature friendly"       "goodnature friendly readymade"
 [55] "friendly readymade opportunity"        "readymade opportunity lawyers"
 [57] "opportunity lawyers friends"           "lawyers friends longest"
 [59] "friends longest affections"            "longest affections implied"
 [61] "affections implied aptness"            "implied aptness richard"
 [63] "aptness richard enfield"               "richard enfield distant"
 [65] "enfield distant kinsman"               "distant kinsman wellknown"
 [67] "kinsman wellknown subject"             "wellknown subject reported"
 [69] "subject reported encountered"          "reported encountered nothing"
 [71] "encountered nothing singularly"        "nothing singularly obvious"
 [73] "singularly obvious appearance"         "obvious appearance greatest"
 [75] "appearance greatest excursions"        "greatest excursions counted"
 [77] "excursions counted occasions"          "counted occasions pleasure"
 [79] "occasions pleasure resisted"           "pleasure resisted business"
 [81] "resisted business uninterrupted"       "business uninterrupted chanced"
 [83] "uninterrupted chanced rambles"         "chanced rambles bystreet"
 [85] "rambles bystreet quarter"              "bystreet quarter thriving"
 [87] "quarter thriving weekdays"             "thriving weekdays inhabitants"
 [89] "weekdays inhabitants emulously"        "inhabitants emulously surplus"
 [91] "emulously surplus coquetry"            "surplus coquetry thoroughfare"
 [93] "coquetry thoroughfare invitation"      "thoroughfare invitation smiling"
 [95] "invitation smiling saleswomen"         "smiling saleswomen comparatively"
 [97] "saleswomen comparatively passage"      "comparatively passage contrast"
 [99] "passage contrast neighbourhood"        "contrast neighbourhood freshly"
[101] "neighbourhood freshly painted"         "freshly painted shutters"
[103] "painted shutters wellpolished"         "shutters wellpolished brasses"
[105] "wellpolished brasses general"          "brasses general cleanliness"
[107] "general cleanliness instantly"         "cleanliness instantly pleased"
[109] "instantly pleased passenger"           "pleased passenger certain"
[111] "passenger certain sinister"            "certain sinister building"
[113] "sinister building forward"             "building forward storeys"
[115] "forward storeys nothing"               "storeys nothing forehead"
[117] "nothing forehead discoloured"          "forehead discoloured feature"
[119] "discoloured feature prolonged"         "feature prolonged negligence"
[121] "prolonged negligence equipped"         "negligence equipped neither"
[123] "equipped neither knocker"              "neither knocker blistered"
[125] "knocker blistered distained"           "blistered distained slouched"
[127] "distained slouched matches"            "slouched matches children"
```

```
 [83] "uninterrupted chanced rambles"        "chanced rambles bystreet"
 [85] "rambles bystreet quarter"             "bystreet quarter thriving"
 [87] "quarter thriving weekdays"            "thriving weekdays inhabitants"
 [89] "weekdays inhabitants emulously"       "inhabitants emulously surplus"
 [91] "emulously surplus coquetry"           "surplus coquetry thoroughfare"
 [93] "coquetry thoroughfare invitation"     "thoroughfare invitation smiling"
 [95] "invitation smiling saleswomen"        "smiling saleswomen comparatively"
 [97] "saleswomen comparatively passage"     "comparatively passage contrast"
 [99] "passage contrast neighbourhood"       "contrast neighbourhood freshly"
[101] "neighbourhood freshly painted"        "freshly painted shutters"
[103] "painted shutters wellpolished"        "shutters wellpolished brasses"
[105] "wellpolished brasses general"         "brasses general cleanliness"
[107] "general cleanliness instantly"        "cleanliness instantly pleased"
[109] "instantly pleased passenger"          "pleased passenger certain"
[111] "passenger certain sinister"           "certain sinister building"
[113] "sinister building forward"            "building forward storeys"
[115] "forward storeys nothing"              "storeys nothing forehead"
[117] "nothing forehead discoloured"         "forehead discoloured feature"
[119] "discoloured feature prolonged"        "feature prolonged negligence"
[121] "prolonged negligence equipped"        "negligence equipped neither"
[123] "equipped neither knocker"             "neither knocker blistered"
[125] "knocker blistered distained"          "blistered distained slouched"
[127] "distained slouched matches"           "slouched matches children"
[129] "matches children schoolboy"           "children schoolboy mouldings"
[131] "schoolboy mouldings generation"       "mouldings generation appeared"
[133] "generation appeared visitors"         "appeared visitors ravages"
[135] "visitors ravages enfield"             "ravages enfield bystreet"
[137] "enfield bystreet abreast"             "bystreet abreast pointed"
[139] "abreast pointed companion"            "pointed companion replied"
[141] "companion replied affirmative"        "replied affirmative connected"
[143] "affirmative connected utterson"       "connected utterson returned"
[145] "utterson returned enfield"            "returned enfield morning"
[147] "enfield morning through"              "morning through literally"
[149] "through literally nothing"            "literally nothing asleep"street"
[151] "nothing asleep"street lighted"        "asleep"street lighted procession"
[153] "lighted procession church"till"       "procession church"till listens"
[155] "church"till listens listens"          "listens listens policeman"
[157] "listens policeman figures"            "policeman figures stumping"
[159] "figures stumping eastward"            "stumping eastward running"
[161] "eastward running another"             "running another naturally"
```

```
[163] "another naturally horrible"            "naturally horrible trampled"
[165] "horrible trampled screaming"           "trampled screaming nothing"
[167] "screaming nothing hellish"             "nothing hellish juggernaut"
[169] "hellish juggernaut collared"           "juggernaut collared gentleman"
[171] "collared gentleman brought"            "gentleman brought already"
[173] "brought already screaming"             "already screaming perfectly"
[175] "screaming perfectly resistance"        "perfectly resistance brought"
[177] "resistance brought running"            "brought running appearance"
[179] "running appearance frightened"         "appearance frightened according"
[181] "frightened according sawbones"         "according sawbones supposed"
[183] "sawbones supposed curious"             "supposed curious circumstance"
[185] "curious circumstance loathing"         "circumstance loathing gentleman"
[187] "loathing gentleman natural"            "gentleman natural doctors"
[189] "natural doctors apothecary"            "doctors apothecary particular"
[191] "apothecary particular edinburgh"       "particular edinburgh emotional"
[193] "edinburgh emotional bagpipe"           "emotional bagpipe prisoner"
[195] "bagpipe prisoner sawbones"             "prisoner sawbones killing"
[197] "sawbones killing question"             "killing question scandal"
[199] "question scandal friends"              "scandal friends undertook"
[201] "friends undertook pitching"            "undertook pitching keeping"
[203] "pitching keeping harpies"              "keeping harpies hateful"
[205] "harpies hateful sneering"              "hateful sneering coolness"frightened"
[207] "sneering coolness"frightened that"but" "coolness"frightened that"but carrying"
[209] "that"but carrying capital"             "carrying capital accidentsaid"
[211] "capital accidentsaid naturally"        "accidentsaid naturally helpless"
[213] "naturally helpless gentleman"          "helpless gentleman screwed"
[215] "gentleman screwed hundred"             "screwed hundred clearly"
[217] "hundred clearly something"             "clearly something mischief"
[219] "something mischief carried"            "mischief carried "whipped"
[221] "carried "whipped presently"            ""whipped presently balance"
[223] "presently balance couttss"             "balance couttss payable"
[225] "couttss payable mention"               "payable mention printed"
[227] "mention printed signature"             "printed signature genuine"
[229] "signature genuine liberty"             "genuine liberty pointing"
[231] "liberty pointing gentleman"            "pointing gentleman business"
[233] "gentleman business apocryphal"         "business apocryphal morning"
[235] "apocryphal morning another"            "morning another hundred"
[237] "another hundred sneering"              "hundred sneering chambers"
[239] "sneering chambers breakfasted"         "chambers breakfasted believe"
```

```
[225] "couttss payable mention"              "payable mention printed"
[227] "mention printed signature"            "printed signature genuine"
[229] "signature genuine liberty"            "genuine liberty pointing"
[231] "liberty pointing gentleman"           "pointing gentleman business"
[233] "gentleman business apocryphal"        "business apocryphal morning"
[235] "apocryphal morning another"           "morning another hundred"
[237] "another hundred sneering"             "hundred sneering chambers"
[239] "sneering chambers breakfasted"        "chambers breakfasted believe"
[241] "breakfasted believe forgery"          "believe forgery genuine"
[243] "forgery genuine utterson"             "genuine utterson enfield"
[245] "utterson enfield damnable"            "enfield damnable proprieties"
[247] "damnable proprieties celebrated"      "proprieties celebrated fellows"
[249] "celebrated fellows blackmail"         "fellows blackmail suppose"
[251] "blackmail suppose through"            "suppose through consequence"
[253] "through consequence explaining"       "consequence explaining recalled"
[255] "explaining recalled utterson"         "recalled utterson suddenly"
[257] "utterson suddenly returned"           "suddenly returned enfield"
[259] "returned enfield noticed"             "enfield noticed address"
[261] "noticed address utterson"             "address utterson delicacy"
[263] "utterson delicacy strongly"           "delicacy strongly putting"
[265] "strongly putting questions"           "putting questions partakes"
[267] "questions partakes judgment"          "partakes judgment question"
[269] "judgment question starting"           "question starting quietly"
[271] "starting quietly starting"            "quietly starting presently"
[273] "starting presently thought"           "presently thought knocked"
[275] "thought knocked studied"              "knocked studied continued"
[277] "studied continued enfield"            "continued enfield scarcely"
[279] "enfield scarcely gentleman"           "scarcely gentleman adventure"
[281] "gentleman adventure windows"          "adventure windows looking"
[283] "windows looking windows"              "looking windows chimney"
[285] "windows chimney generally"            "chimney generally smoking"
[287] "generally smoking somebody"           "smoking somebody buildings"
[289] "somebody buildings together"          "buildings together another"
[291] "together another silence"             "another silence enfield"
[293] "silence enfield utterson"             "enfield utterson returned"
[295] "utterson returned enfield"            "returned enfield continued"
[297] "enfield continued enfield"            "continued enfield utterson"
[299] "enfield utterson describe"            "utterson describe something"
[301] "describe something appearance"        "something appearance something"
[303] "appearance something displeasing"     "something displeasing something"
[305] "displeasing something downright"      "something downright detestable"
```

```
[247] "damnable proprieties celebrated"      "proprieties celebrated fellows"
[249] "celebrated fellows blackmail"         "fellows blackmail suppose"
[251] "blackmail suppose through"            "suppose through consequence"
[253] "through consequence explaining"       "consequence explaining recalled"
[255] "explaining recalled utterson"         "recalled utterson suddenly"
[257] "utterson suddenly returned"           "suddenly returned enfield"
[259] "returned enfield noticed"             "enfield noticed address"
[261] "noticed address utterson"             "address utterson delicacy"
[263] "utterson delicacy strongly"           "delicacy strongly putting"
[265] "strongly putting questions"           "putting questions partakes"
[267] "questions partakes judgment"          "partakes judgment question"
[269] "judgment question starting"           "question starting quietly"
[271] "starting quietly starting"            "quietly starting presently"
[273] "starting presently thought"           "presently thought knocked"
[275] "thought knocked studied"              "knocked studied continued"
[277] "studied continued enfield"            "continued enfield scarcely"
[279] "enfield scarcely gentleman"           "scarcely gentleman adventure"
[281] "gentleman adventure windows"          "adventure windows looking"
[283] "windows looking windows"              "looking windows chimney"
[285] "windows chimney generally"            "chimney generally smoking"
[287] "generally smoking somebody"           "smoking somebody buildings"
[289] "somebody buildings together"          "buildings together another"
[291] "together another silence"             "another silence enfield"
[293] "silence enfield utterson"             "enfield utterson returned"
[295] "utterson returned enfield"            "returned enfield continued"
[297] "enfield continued enfield"            "continued enfield utterson"
[299] "enfield utterson describe"            "utterson describe something"
[301] "describe something appearance"        "something appearance something"
[303] "appearance something displeasing"     "something displeasing something"
[305] "displeasing something downright"      "something downright detestable"
[307] "downright detestable disliked"        "detestable disliked deformed"
[309] "disliked deformed somewhere"          "deformed somewhere feeling"
[311] "somewhere feeling deformity"          "feeling deformity although"
[313] "deformity although couldnt"           "although couldnt specify"
[315] "couldnt specify extraordinary"        "specify extraordinary looking"
[317] "extraordinary looking nothing"        "looking nothing describe"
[319] "nothing describe declare"             "describe declare utterson"
[321] "declare utterson silence"             "utterson silence obviously"
```

```
[267] "questions partakes judgment"          "partakes judgment question"
[269] "judgment question starting"           "question starting quietly"
[271] "starting quietly starting"            "quietly starting presently"
[273] "starting presently thought"           "presently thought knocked"
[275] "thought knocked studied"              "knocked studied continued"
[277] "studied continued enfield"            "continued enfield scarcely"
[279] "enfield scarcely gentleman"           "scarcely gentleman adventure"
[281] "gentleman adventure windows"          "adventure windows looking"
[283] "windows looking windows"              "looking windows chimney"
[285] "windows chimney generally"            "chimney generally smoking"
[287] "generally smoking somebody"           "smoking somebody buildings"
[289] "somebody buildings together"          "buildings together another"
[291] "together another silence"             "another silence enfield"
[293] "silence enfield utterson"             "enfield utterson returned"
[295] "utterson returned enfield"            "returned enfield continued"
[297] "enfield continued enfield"            "continued enfield utterson"
[299] "enfield utterson describe"            "utterson describe something"
[301] "describe something appearance"        "something appearance something"
[303] "appearance something displeasing"     "something displeasing something"
[305] "displeasing something downright"      "something downright detestable"
[307] "downright detestable disliked"        "detestable disliked deformed"
[309] "disliked deformed somewhere"          "deformed somewhere feeling"
[311] "somewhere feeling deformity"          "feeling deformity although"
[313] "deformity although couldnt"           "although couldnt specify"
[315] "couldnt specify extraordinary"        "specify extraordinary looking"
[317] "extraordinary looking nothing"        "looking nothing describe"
[319] "nothing describe declare"             "describe declare utterson"
[321] "declare utterson silence"             "utterson silence obviously"
[323] "silence obviously consideration"      "obviously consideration inquired"
[325] "consideration inquired enfield"       "inquired enfield surprised"
[327] "enfield surprised himself"            "surprised himself utterson"
[329] "himself utterson strange"             "utterson strange because"
[331] "strange because already"              "because already richard"
[333] "already richard inexact"              "richard inexact correct"
[335] "inexact correct returned"             "correct returned sullenness"
[337] "returned sullenness pedantically"     "sullenness pedantically utterson"
[339] "pedantically utterson presently"      "utterson presently resumed"
[341] "presently resumed another"            "resumed another nothing"
[343] "another nothing ashamed"              "nothing ashamed bargain"
[345] "ashamed bargain richard"
```

As seen, after applying the vapply function we were able to calculate the bigrams and trigrams for the chapter 1 where the character length is greater than 6. The above screenshots displays the bigrams and trigrams for the chapter 1.Library tau and tm library are used for analyzing and cleaning the text. Thus, this is how the bigrams and trigrams for chapter 1 were generated.

Part h:Process the data from Chapter 1 using methods from stringi, corpustools, quanteda, and tidytext packages

```
df<-read.delim
("/Users/ishaterdal/Dropbox/BigData/Project3/DrJekyllAndMrHyde.txt")
df2<-paste(unlist(df),collapse=")
df2<-gsub(".*STORY OF THE DOOR\\s*|SEARCH FOR MR. HYDE.*","",df2)
```

Stringi functions

1) Remove html tags with stri_replace_all( )

```
df2 <- stri_replace_all(df2, "", regex = "<.*?>")
```

2) Remove surrounding whitespace with stri_trim( )

```
df2 <- stri_trim(df2)
```

3) Transform all characters to lowercase with stri_trans_tolower( )

```
df2 <- stri_trans_tolower(df2)
```

```
> df2
[1] "mr. utterson the lawyer was a man of a rugged countenance that wasnever lighted by
a smile; cold, scanty and embarrassed in discourse;backward in sentiment; lean, long, d
usty, dreary and yet somehowlovable. at friendly meetings, and when the wine was to his
taste,something eminently human beaconed from his eye; something indeed whichnever foun
d its way into his talk, but which spoke not only in thesesilent symbols of the after-d
inner face, but more often and loudly inthe acts of his life. he was austere with himse
lf; drank gin when hewas alone, to mortify a taste for vintages; and though he enjoyed
thetheatre, had not crossed the doors of one for twenty years. but he hadan approved to
lerance for others; sometimes wondering, almost withenvy, at the high pressure of spiri
ts involved in their misdeeds; andin any extremity inclined to help rather than to repr
ove. “i incline tocain’s heresy,” he used to say quaintly: “i let my brother go to thed
evil in his own way.” in this character, it was frequently his fortuneto be the last re
```

The above screenshot shows Chapter 1 in all lowercase letters, with all HTML tags and whitespaces surrounding the text removed.

Quanteda functions

1) The tokens( ) function splits a single sentence into words or unigrams
tokschap1 <- tokens(df2)

2) The tokens_remove( ) function in this case eliminates all the english stopwords such as ‘the’, ‘I’, and ‘or’ that do not prove useful in analysis
sw <- stopwords("english")
tokens_remove(tokschap1, sw)

```
> sw <- stopwords("english")
> tokens_remove(tokschap1, sw)
tokens from 1 document.
text1 :
   [1] "mr"                  "."                   "utterson"            "lawyer"
   [5] "man"                 "rugged"              "countenance"         "wasnever"
   [9] "lighted"             "smile"               ";"                   "cold"
  [13] ","                   "scanty"              "embarrassed"         "discourse"
  [17] ";"                   "backward"            "sentiment"           ";"
  [21] "lean"                ","                   "long"                ","
  [25] "dusty"               ","                   "dreary"              "yet"
  [29] "somehowlovable"      "."                   "friendly"            "meetings"
  [33] ","                   "wine"                "taste"               ","
  [37] "something"           "eminently"           "human"               "beaconed"
  [41] "eye"                 ";"                   "something"           "indeed"
  [45] "whichnever"          "found"               "way"                 "talk"
  [49] ","                   "spoke"               "thesesilent"         "symbols"
  [53] "after-dinner"        "face"                ","                   "often"
```

3) The corpus( ) function creates a corpus of Chapter 1, while the dfm( ) function creates a *document-feature-matrix* which is a matrix where rows are documents, columns are terms, and cells indicate how often each term occurred in each document.
text <- corpus(df2)
dtm <- dfm(text, remove = sw, remove_punct=TRUE)

```
> text <- corpus(df2)
> dtm <- dfm(text, remove = sw, remove_punct=TRUE)
> dtm
Document-feature matrix of: 1 document, 753 features (0.0% sparse).
```

Corpustools functions
1)The create_tcorpus function creates a corpus

tc <- create_tcorpus(df2)

2) The search_features( ) function queries ‘loathing’ and ‘gentleman’ within 5 words of each other
hits <- tc$search_features('"loathing gentleman"~5')

3) Key Word In Context kwic( ) listing
kwic <- tc$kwic(hits, ntokens = 3)
head(kwic$kwic, 3)

```
> tc <- create_tcorpus(df2)
> hits <- tc$search_features('"loathing gentleman"~5')
created index for "token" column
> kwic <- tc$kwic(hits, ntokens = 3)
> head(kwic$kwic, 3)
[1] "...had taken a <loathing> to my <gentleman> at firstsight.
..."
```

Tidytext functions

1) The first step in obtaining tidy text is to get it into *one-token-per-row* format through the unnest_tokens( ) function
chap1_df <- tibble(text=df2)
tidy <- chap1_df %>% unnest_tokens(word, text)

```
> tidy <- chap1_df %>% unnest_tokens(word, text)
> tidy
# A tibble: 2,251 x 1
   word
   <chr>
 1 mr
 2 utterson
 3 the
 4 lawyer
 5 was
 6 a
 7 man
 8 of
 9 a
10 rugged
# … with 2,241 more rows
```

2) The anti_join( ) function works to remove stopwords from the text
data("stop_words")
tidy <- tidy %>% anti_join(stop_words)

```
> tidy <- tidy %>% anti_join(stop_words)
Joining, by = "word"
> tidy
# A tibble: 769 x 1
   word
   <chr>
 1 utterson
 2 lawyer
 3 rugged
 4 countenance
 5 wasnever
 6 lighted
 7 smile
 8 cold
 9 scanty
10 embarrassed
# … with 759 more rows
```

3) The count( ) function finds the most common words in Chapter 1
common <- tidy %>% count(word, sort=TRUE)

```
> common <- tidy %>% count(word, sort=TRUE)
> common
# A tibble: 615 x 2
   word          n
   <chr>     <int>
 1 utterson     11
 2 enfield      10
 3 street       10
 4 door          7
 5 sir           7
 6 cheque        6
 7 it's          6
 8 gentleman     5
 9 lawyer        5
10 child's       4
# … with 605 more rows
```

The above function results indicate that the book has been written quite long ago, probably centuries ago, given the words most commonly used in the first chapter alone. It also appears to have a dark theme given its scattering of negative words in the text i.e. first chapter.

What this project helped in learning about text analytics?

This project helped us in understanding the important aspects of text analytics. We could infer a great amount of information about the book from this project. Although it was difficult, but text analytics can help us in understanding a larger perspective of words. We could explore a good amount of text and its semantics. This project helped us in learning about the various ways in which text could be analyzed, processed and different meaning derivations. The idea and concept of the book were very well analyzed through this project.