NLP of Gods of Mars

Ran Wei

Apr 24, 2022

Part 1 Analysis Preparation

For this analysis, I start by loading the document "*The Gods of Mars*". I use VCorpus function to load the entire document into memory, while ignore case for the longest words and sentences analysis. Calling the str() for the book shows me that there are 9313 lines of text, and also provides additional meta data (Figure 1).

```
> god_of_mars <- VCorpus(DirSource("./txt", ignore.case = TRUE, mode = 'text'))
> str(god_of_mars)
Classes 'VCorpus', 'Corpus'  hidden list of 3
 $ content:List of 1
  ..$ :List of 2
  .. ..$ content: chr [1:9313] "Title: The Gods of Mars" "" "Author: Edgar Rice Burroughs" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 03:43:54"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "GodsOfMars.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ meta   : list()
  ..- attr(*, "class")= chr "CorpusMeta"
 $ dmeta  :'data.frame':        1 obs. of  0 variables
```

Figure 1. Data Structure of Whole Document

Then I extract the text from corpus (Figure 2), which contains 443,046 characters, and see the content of the document (Figure 3) with the input 'mars_text[1]'.

```
> mars_text <- god_of_mars[[1]]
> mars_text
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 443046
```

Figure 2. Extract Text

```
 [3] "Author: Edgar Rice Burroughs"
 [4] ""
 [5] "THE GODS OF MARS"
 [6] ""
 [7] "Edgar Rice Burroughs"
 [8] ""
 [9] "FOREWORD"
[10] ""
[11] ""
[12] "Twelve years had passed since I had laid the body of my great-uncle,"
[13] "Captain John Carter, of Virginia, away from the sight of men in that"
[14] "strange mausoleum in the old cemetery at Richmond."
[15] ""
[16] "Often had I pondered on the odd instructions he had left me governing"
[17] "the construction of his mighty tomb, and especially those parts which"
[18] "directed that he be laid in an OPEN casket and that the ponderous"
[19] "mechanism which controlled the bolts of the vault's huge door be"
[20] "accessible ONLY FROM THE INSIDE."
```

Figure 3. Line 3-20 of the Text Content

The next step in my analysis is to separate the chapters. After creating a new folder named 'chapters', I apply the which() function to identify the starting points for Chapters 1, 2, and 3. Next, the write.table() is used to save my newly sliced chapters into the

'chapters' folder (Figure 4). The total number of lines in Chapter 1&2 is 917 (445 lines in Chapter 1 and 472 lines in Chapter 2) including blank lines (Figure 5). Note that I didn't take the content before Chapter 1 into consideration, like Title, Author, Foreword and Contents, which contains 161 lines in total. The titles of each Chapter, like 'CHAPTER I' and 'CHAPTER II', were also omitted. Therefore, the number of lines before 'CHAPTER III' should be 1080 including blank lines.

```
> dir.create('chapters')
> index_ch1 <- which(mars_text$content == "CHAPTER I", arr.ind = TRUE)
> index_ch2 <- which(mars_text$content == "CHAPTER II", arr.ind = TRUE)
> index_ch3 <- which(mars_text$content == "CHAPTER III", arr.ind = TRUE)
> book_chapter1 <- mars_text$content[(index_ch1+1):(index_ch2-1)]
> book_chapter2 <- mars_text$content[(index_ch2+1):(index_ch3-1)]
> write.table(book_chapter1, file = "chapters/god_of_mars_chapter1.txt", sep = "\t", row.names=FALSE, col.names=FALSE,quote=FALSE)
> write.table(book_chapter2, file = "chapters/god_of_mars_chapter2.txt", sep = "\t", row.names=FALSE, col.names=FALSE,quote=FALSE)
> dir.create('chapters')
```

Figure 4. Separate the First 2 Chapters

```
> god_of_mars_chapters <- VCorpus(DirSource("chapters", ignore.case = TRUE, mode = 'text'))
> str(god_of_mars_chapters)
Classes 'VCorpus', 'Corpus'  hidden list of 3
 $ content:List of 2
  ..$ :List of 2
  .. ..$ content: chr [1:445] "" "THE PLANT MEN" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter1.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
  ..$ :List of 2
  .. ..$ content: chr [1:472] "" "A FOREST BATTLE" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter2.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ meta   : list()
  ..- attr(*, "class")= chr "CorpusMeta"
 $ dmeta  :'data.frame':        2 obs. of  0 variables
```

Figure 5. Data Structure of Chapter 1&2

Part 2 Find the Longest Words and Sentences

I created two functions to get the 10 longest words (Figure 6) and the 10 longest sentences (Figure 7) before removing the punctuation. These are then applied to Chapter 1 and Chapter 2 respectively (Figure 8). The results for 10 longest words is shown in Table 1, and the results for 10 longest sentences is show in Table 2.

```
get_10_longest_words <- function(chapter){
  all_lengths = data.frame(word = character(),
                           length = integer())
  for (line in chapter){
    chars = str_split(line, " ")
    for (char in chars){
      char_length = str_length(char)
      all_lengths = all_lengths %>%
        add_row(word = char, length = char_length)
    }

  }

  ordered = all_lengths[order(all_lengths$length, decreasing = TRUE),]

  top_10 = ordered[1:10,]
  return(top_10)
}
```

Figure 6. Function of Finding 10 Longest Words

```
get_10_longest_sentences <- function(chapter){

  all_lengths = data.frame(sentence = character(),
                           length = integer())

  one_text = paste(chapter, collapse="")
  sentences = str_split(one_text, pattern = '[.!?]')

  for (sentence_ in sentences){
    sent_length = str_length(sentence_)
    all_lengths = all_lengths %>%
      add_row(sentence = sentence_, length = sent_length)

  }

  ordered = all_lengths[order(all_lengths$length, decreasing = TRUE),]

  top_10 = ordered[1:10,]
  return(top_10)
}
```

Figure 7. Function of Finding 10 Longest Sentences

```
> words_ch1 = get_10_longest_words(book_chapter1)
> words_ch2 = get_10_longest_words(book_chapter2)
> sentences_ch1 <- get_10_longest_sentences(book_chapter1)
> sentences_ch2 <- get_10_longest_sentences(book_chapter2)
```

Figure 8. Apply Functions to First two Chapters

Table 1. 10 Longest words in the First two Chapters
((a) chapter 1; (b) chapter 2)

	word	length
2062	Fearsome-looking	16
499	interplanetary	14
776	forgetfulness,	14
2800	demonstrations	14
3025	simultaneously	14
528	close-cropped	13
762	comparatively	13
801	imperceptible	13
825	close-cropped	13
1024	American-made	13

(a)

	word	length
4083	well-proportioned	17
1591	creatures--great	16
835	walls--possible	15
2414	ever-increasing	15
4129	ever-increasing	15
303	disintegration	14
1507	uninterrupted.	14
239	perpendicular	13
951	opportunities	13
966	opportunities	13

(b)

Table 2. 10 Longest sentences in the First two Chapters
((a) chapter 1; (b) chapter 2)

	sentence	length
5	Instantly my brain cleared and there swept back across the threshold ofmy memory the vi...	725
66	There were two men and four females in the party and their ornamentsdenoted them as ...	608
93	A glance in the direction toward which he was looking was sufficient toapprise me of his a...	522
64	Here were the great males towering in all the majesty of their imposingheight; here were t...	494
54	Naked and unarmed, as I was, my end would have been both speedy andhorrible at the h...	474
51	As I had been scrutinizing this weird monstrosity the balance of theherd had fed quite clo...	443
81	Half a dozen great leaps brought me to the spot, and another instantsaw me again in my ...	443
84	For an instant they recoiled before my terrific onslaught, and in thatinstant the green warr...	440
53	Fearsome-looking as they were, I did not know whether to fear them ornot, for they did n...	432
69	Presently the leader of the plant men charged the little party, and hismethod of attack wa...	402

(a)

	sentence	length
55	The great tails of theplant men lashed with tremendous power about us as they charged from...	401
21	It seemed the forest now or nothing, and I was just on the point ofmotioning Tars Tarkas to foll...	394
22	The face of the entire cliff was, as later inspection conclusivelyproved, so shot with veins and pa...	393
31	Our relentless pursuers were now close to us, so close that it seemedthat it would be an utter i...	391
53	What it has taken minutes to write occurred in but a few seconds, butduring that time Tars Tark...	369
81	At length, all but a score, who had apparently been left to prevent ourescape, had left us, and o...	352
36	He was, I should say, a hundred yards in advance of his closestcompanion, and so I called to Tar...	340
1	A FOREST BATTLETars Tarkas and I found no time for an exchange of experiences as westood th...	338
76	With the fear that we would escape them, the creatures redoubled theirefforts to pull me down...	337
50	It was into the eyes of such as these and the terrible plant men that Igazed above the shoulder ...	334

(b)

Part 3 Apply Methods in Rubric

In this section, I apply the methods in *Introduction to Text Analytics* and *text_analysis_in_R* to get better understanding of the first two Chapters.

- I firstly compute document term matrix (DTM) and term document matrix (TDM), as well as examine the structure and inspect the data (Figure 9 and Figure 10).

```
> gom_DTM <- DocumentTermMatrix(god_of_mars_chapters)
> gom_DTM
<<DocumentTermMatrix (documents: 2, terms: 2263)>>
Non-/sparse entries: 2760/1766
Sparsity           : 39%
Maximal term length: 17
Weighting          : term frequency (tf)
> inspect(gom_DTM)
<<DocumentTermMatrix (documents: 2, terms: 2263)>>
Non-/sparse entries: 2760/1766
Sparsity           : 39%
Maximal term length: 17
Weighting          : term frequency (tf)
Sample             :
                    Terms
Docs                 and for from had that the upon was which with
  god_of_mars_chapter1.txt 147  24   41  45   63 328   29  43    38   40
  god_of_mars_chapter2.txt 124  42   29  35   66 309   30  48    24   37
> str(gom_DTM)
List of 6
 $ i       : int [1:2760] 1 1 1 1 1 1 1 1 1 1 ...
 $ j       : int [1:2760] 9 11 14 15 16 20 21 22 24 25 ...
 $ v       : num [1:2760] 1 1 1 1 1 14 7 1 1 1 ...
 $ nrow    : int 2
 $ ncol    : int 2263
 $ dimnames:List of 2
  ..$ Docs : chr [1:2] "god_of_mars_chapter1.txt" "god_of_mars_chapter2.txt"
  ..$ Terms: chr [1:2263] "\"as" "\"but" "\"come,\"" "\"for" ...
 - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

Figure 9. DTM

```
> gom_TDM <- TermDocumentMatrix(god_of_mars_chapters)
> gom_TDM
<<TermDocumentMatrix (terms: 2263, documents: 2)>>
Non-/sparse entries: 2760/1766
Sparsity           : 39%
Maximal term length: 17
Weighting          : term frequency (tf)
> inspect(gom_TDM)
<<TermDocumentMatrix (terms: 2263, documents: 2)>>
Non-/sparse entries: 2760/1766
Sparsity           : 39%
Maximal term length: 17
Weighting          : term frequency (tf)
Sample             :
       Docs
Terms   god_of_mars_chapter1.txt god_of_mars_chapter2.txt
  and                        147                       124
  for                         24                        42
  from                        41                        29
  had                         45                        35
  that                        63                        66
  the                        328                       309
  upon                        29                        30
  was                         43                        48
  which                       38                        24
  with                        40                        37
> str(gom_TDM)
List of 6
 $ i       : int [1:2760] 9 11 14 15 16 20 21 22 24 25 ...
 $ j       : int [1:2760] 1 1 1 1 1 1 1 1 1 1 ...
 $ v       : num [1:2760] 1 1 1 1 1 14 7 1 1 1 ...
 $ nrow    : int 2263
 $ ncol    : int 2
 $ dimnames:List of 2
  ..$ Terms: chr [1:2263] "\"as" "\"but" "\"come,\"" "\"for" ...
  ..$ Docs : chr [1:2] "god_of_mars_chapter1.txt" "god_of_mars_chapter2.txt"
 - attr(*, "class")= chr [1:2] "TermDocumentMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

Figure 10. TDM

- The output signifies that the two documents contain 2263 terms which have appeared at least once.
- 1766 cells in frequencies are 0, 2760 cells have non-zero values. The sparsity is 39%, which means that 39% (=1766/(2760+1766)) of all cells in the matrix are zero .
- Some sample words with high frequency are listed, like ‘the’, which appears 328 times in chapter 1, and 309 times in chapter 2.

- I also convert the content of the document into data frame (Figure 11).

```
> mars_ch1_text <- god_of_mars_chapters[[1]]
> mars_ch2_text <- god_of_mars_chapters[[2]]
> ch1_df <- data.frame(mars_ch1_text[1])
> ch2_df <- data.frame(mars_ch2_text[1])
> ch1_df[1]
                                                                    content
1
2                                                                THE PLANT MEN
3
4
5    As I stood upon the bluff before my cottage on that clear cold night in
6      the early part of March, 1886, the noble Hudson flowing like the grey
7     and silent spectre of a dead river below me, I felt again the strange,
8      compelling influence of the mighty god of war, my beloved Mars, which
9      for ten long and lonesome years I had implored with outstretched arms
10                                          to carry me back to my lost love.
11
12   Not since that other March night in 1866, when I had stood without that
13       Arizona cave in which my still and lifeless body lay wrapped in the
14     similitude of earthly death had I felt the irresistible attraction of
15                                                 the god of my profession.
```

Figure 11. Date Frame of the Content of Chapter 1

```
> god_of_mars_chapters[[1]]$content
 [1] ""
 [2] "THE PLANT MEN"
 [3] ""
 [4] ""
 [5] "As I stood upon the bluff before my cottage on that clear cold night in"
 [6] "the early part of March, 1886, the noble Hudson flowing like the grey"
 [7] "and silent spectre of a dead river below me, I felt again the strange,"
 [8] "compelling influence of the mighty god of war, my beloved Mars, which"
 [9] "for ten long and lonesome years I had implored with outstretched arms"
[10] "to carry me back to my lost love."
[11] ""
[12] "Not since that other March night in 1866, when I had stood without that"
[13] "Arizona cave in which my still and lifeless body lay wrapped in the"
[14] "similitude of earthly death had I felt the irresistible attraction of"
[15] "the god of my profession."
```

Figure 12. Content of Chapter 1

- By comparing the result with the content of chapter 1 (Figure 12), I can notice that the double quote marks in the start and end of each line are deleted while the content is converted to data frame.

- Then I apply functions to wrangle data: First removing numbers and punctuations (Figure 13), then converting every character into lower case (Figure 14).

```
> removeQuote <- function(x) gsub('[\"]', '', x)
> god_of_mars_chapters_cl <- tm::tm_map(god_of_mars_chapters, content_transformer(removeQuote))
> removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*","",x)
> god_of_mars_chapters_cl <- tm::tm_map(god_of_mars_chapters_cl, content_transformer(removeNumPunct))
> str(god_of_mars_chapters_cl)
Classes 'VCorpus', 'Corpus'  hidden list of 3
 $ content:List of 2
  ..$ :List of 2
  .. ..$ content: chr [1:445] "" "THE PLANT MEN" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter1.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
  ..$ :List of 2
  .. ..$ content: chr [1:472] "" "A FOREST BATTLE" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter2.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ meta   : list()
  ..- attr(*, "class")= chr "CorpusMeta"
 $ dmeta  :'data.frame':        2 obs. of  0 variables
> inspect(god_of_mars_chapters)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 22479

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 23382

> inspect(god_of_mars_chapters_cl)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 22077

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 22900
```

Figure 13. Remove Numbers and Punctuation

```
> god_of_mars_chapters_lower <- tm_map(god_of_mars_chapters_cl, content_transformer(tolower))
> str(god_of_mars_chapters_lower)
Classes 'VCorpus', 'Corpus'  hidden list of 3
 $ content:List of 2
  ..$ :List of 2
  .. ..$ content: chr [1:445] "" "the plant men" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter1.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
  ..$ :List of 2
  .. ..$ content: chr [1:472] "" "a forest battle" "" "" ...
  .. ..$ meta   :List of 7
  .. .. ..$ author       : chr(0)
  .. .. ..$ datetimestamp: POSIXlt[1:1], format: "2022-04-25 04:42:28"
  .. .. ..$ description  : chr(0)
  .. .. ..$ heading      : chr(0)
  .. .. ..$ id           : chr "god_of_mars_chapter2.txt"
  .. .. ..$ language     : chr "en"
  .. .. ..$ origin       : chr(0)
  .. .. ..- attr(*, "class")= chr "TextDocumentMeta"
  .. ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ meta   : list()
  ..- attr(*, "class")= chr "CorpusMeta"
 $ dmeta  :'data.frame':        2 obs. of  0 variables
> inspect(god_of_mars_chapters_lower)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 22077

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 22900
```

Figure 14. Convert to Lower Case

- Note that I have removed 402 characters from chapter 1, and 482 characters from chapter 2, and all the characters now are in lower case.

- I compute the DTM again (Figure 15) using the cleaned data.

```
> gom_DTM <- DocumentTermMatrix(god_of_mars_chapters_lower)
> gom_DTM
<<DocumentTermMatrix (documents: 2, terms: 1873)>>
Non-/sparse entries: 2370/1376
Sparsity           : 37%
Maximal term length: 16
Weighting          : term frequency (tf)
> str(gom_DTM)
List of 6
 $ i       : int [1:2370] 1 1 1 1 1 1 1 1 1 1 ...
 $ j       : int [1:2370] 3 4 6 7 10 11 12 13 14 16 ...
 $ v       : num [1:2370] 14 8 1 1 1 4 1 1 1 1 ...
 $ nrow    : int 2
 $ ncol    : int 1873
 $ dimnames:List of 2
  ..$ Docs : chr [1:2] "god_of_mars_chapter1.txt" "god_of_mars_chapter2.txt"
  ..$ Terms: chr [1:1873] "abandoned" "able" "about" "above" ...
 - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

Figure 15. Convert to Lower Case

- After data wrangling, the number of terms reduced by 390 (=2263-1873);
- The sparsity reduced from 39% to 37%.

- Stop words are considered uninformative to my analysis, so I want to remove them. This can be done easily using the tm package. The package provides a list of stop words in English that can be easily filtered out using removeWords (Figure 16).

```
> myStopWords <- c(tm::stopwords("english"))
> myStopWords
  [1] "i"          "me"         "my"         "myself"     "we"         "our"        "ours"       "ourselves"  "you"
 [10] "your"       "yours"      "yourself"   "yourselves" "he"         "him"        "his"        "himself"    "she"
 [19] "her"        "hers"       "herself"    "it"         "its"        "itself"     "they"       "them"       "their"
 [28] "theirs"     "themselves" "what"       "which"      "who"        "whom"       "this"       "that"       "these"
 [37] "those"      "am"         "is"         "are"        "was"        "were"       "be"         "been"       "being"
 [46] "have"       "has"        "had"        "having"     "do"         "does"       "did"        "doing"      "would"
 [55] "should"     "could"      "ought"      "i'm"        "you're"     "he's"       "she's"      "it's"       "we're"
 [64] "they're"    "i've"       "you've"     "we've"      "they've"    "i'd"        "you'd"      "he'd"       "she'd"
 [73] "we'd"       "they'd"     "i'll"       "you'll"     "he'll"      "she'll"     "we'll"      "they'll"    "isn't"
 [82] "aren't"     "wasn't"     "weren't"    "hasn't"     "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"     "couldn't"   "mustn't"    "let's"
[100] "that's"     "who's"      "what's"     "here's"     "there's"    "when's"     "where's"    "why's"      "how's"
[109] "a"          "an"         "the"        "and"        "but"        "if"         "or"         "because"    "as"
[118] "until"      "while"      "of"         "at"         "by"         "for"        "with"       "about"      "against"
[127] "between"    "into"       "through"    "during"     "before"     "after"      "above"      "below"      "to"
[136] "from"       "up"         "down"       "in"         "out"        "on"         "off"        "over"       "under"
[145] "again"      "further"    "then"       "once"       "here"       "there"      "when"       "where"      "why"
[154] "how"        "all"        "any"        "both"       "each"       "few"        "more"       "most"       "other"
[163] "some"       "such"       "no"         "nor"        "not"        "only"       "own"        "same"       "so"
[172] "than"       "too"        "very"
> god_of_mars_chapters_stop <- tm_map(god_of_mars_chapters_lower, removeWords, myStopWords)
> inspect(god_of_mars_chapters_stop[[1]])
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 15808


 plant men


  stood upon  bluff   cottage   clear cold night
 early part  march   noble hudson flowing like  grey
 silent spectre   dead river    felt   strange
compelling influence   mighty god  war  beloved mars
 ten long  lonesome years   implored  outstretched arms
 carry  back   lost love

 since   march night      stood without
arizona cave    still  lifeless body lay wrapped
similitude  earthly death   felt  irresistible attraction
 god   profession
> inspect(god_of_mars_chapters_stop[[2]])
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 16626


 forest battle


tars tarkas   found  time   exchange  experiences
stood    great boulder surrounded   corpses
grotesque assailants    directions   broad valley
streaming  perfect torrent  terrifying creatures  response
weird call   strange figure far  us
```

Figure 16. Remove Stop Words – Step 1

- The ‘english’ list contains 174 words which are used frequently in the English language.
- After removing the words in ‘english’ list, the remaining content of chapter 1 has 15808 characters, and the remaining content of chapter 2 has 16626 characters.

- Now I can create a new TDM using the data without stop words and apply findFreqTerms with a lowFreq parameter of 6 to find the words that appear at least 6 times (Figure 17). The nchar() function is applied to show the number of characters

in the selected word. The termFreq returns the term frequency for all words (Figure 18).

```
> gom_ch_stop_TDM <- TermDocumentMatrix(god_of_mars_chapters_stop)
> freq_terms <- findFreqTerms(gom_ch_stop_TDM, lowfreq = 6)
> freq_terms
  [1] "across"     "another"    "apes"       "attack"     "awful"      "back"       "bars"       "barsoom"    "base"
 [10] "behind"     "beneath"    "body"       "boulder"    "broad"      "came"       "carter"     "cave"       "cliff"
 [19] "cliffs"     "close"      "come"       "creature"   "creatures"  "cruel"      "cut"        "dead"       "death"
 [28] "direction"  "distance"   "earthly"    "either"     "entire"     "ere"        "escape"     "even"       "ever"
 [37] "every"      "eye"        "eyes"       "face"       "far"        "fear"       "feet"       "felt"       "find"
 [46] "first"      "five"       "forest"     "found"      "gorgeous"   "great"      "green"      "ground"     "hands"
 [55] "heads"      "heart"      "height"     "herd"       "hideous"    "however"    "huge"       "hundred"    "instant"
 [64] "john"       "know"       "last"       "lay"        "left"       "length"     "light"      "like"       "little"
 [73] "long"       "longsword"  "lower"      "man"        "manner"     "mars"       "martians"   "may"        "men"
 [82] "met"        "might"      "mighty"     "moment"     "much"       "muscles"    "never"      "now"        "one"
 [91] "opening"    "plant"      "point"      "possible"   "presently"  "quickly"    "quite"      "rapidly"    "reach"
[100] "reached"    "red"        "remarkable" "right"      "river"      "rose"       "saw"        "sea"        "seemed"
[109] "seen"       "set"        "shelter"    "side"       "since"      "single"     "soon"       "sprang"     "still"
[118] "stood"      "strange"    "surface"    "sward"      "tarkas"     "tars"       "ten"        "thark"      "thing"
[127] "though"     "thousand"   "thus"       "time"       "toward"     "tree"       "trees"      "turned"     "two"
[136] "upon"       "valley"     "warrior"    "way"        "weight"     "weird"      "well"       "white"      "will"
[145] "without"
```

Figure 17. Find the Words Appear More Than 6 Times

```
> nchar(freq_terms[8])
[1] 7
> freq_terms[8]
[1] "barsoom"
> gom_ch1_tf <- termFreq(god_of_mars_chapters_stop[[1]])
> gom_ch1_tf
       abutted       accorded      acquainted          across             act          acting          action        addition
             1              1              1               4               1               1               1               1
    additional      admiration           adult          adults       advancing        afforded           agent       aimlessly
             1              2              1               1               1               1               1               1
          aims            air           alike          almost           aloft         already            also       amazement
             1              4              1               2               2               2               1               1
   americanmade          among     anaesthesia           anger          angles      angleworm         another     antagonists
             1              2              1               1               1               1               4               1
   antennaelike       anything       apparent      appearance        appeared     apprehension         apprise        approach
             1              2              1               3               2               1               1               1
> gom_ch2_tf <- termFreq(god_of_mars_chapters_stop[[2]])
> gom_ch2_tf
     abandoned           able       absorbed   accumulation     accustomed         across        actions       admitted
             1              1              1               1               1              5               1              1
           ado        advance       advanced      adventure        african          agile        agility         agreed
             1              3              1               1               1              1               3              1
         ahead            air       alluring         almost          aloft          alone           along        already
             1              1              1               3               2              2               3              1
   alternative     altogether         always         amidst          among        ancient           angle         ankles
             1              1              3               1               1              1               1              1
       another      antagonist    antagonists          apart       apartment            ape        aperture           apes
             7              2              1               1               1              1               1              6
```

Figure 18. Looking into Term Length and Frequency
(The result of term frequency is shown partially due to the limitation of space)

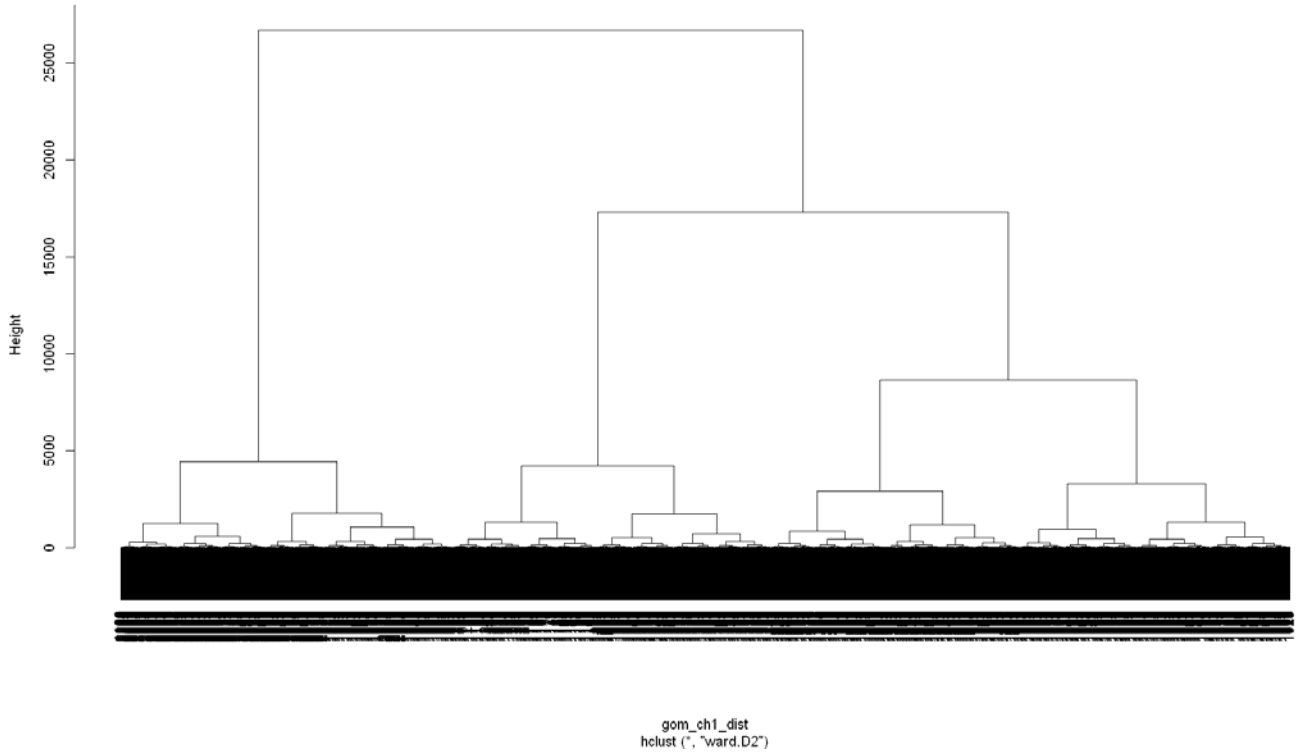
- Then I inspect the TDM again (Figure 19) and draw dendrograms to check the distribution of clusters (Figure 20).

```
> inspect(gom_ch_stop_TDM)
<<TermDocumentMatrix (terms: 1787, documents: 2)>>
Non-/sparse entries: 2210/1364
Sparsity           : 38%
Maximal term length: 16
Weighting          : term frequency (tf)
Sample             :
        Docs
Terms     god_of_mars_chapter1.txt god_of_mars_chapter2.txt
  feet                          10                       21
  great                         20                       15
  green                         15                        6
  men                           15                       12
  now                           10                       13
  one                           16                       11
  strange                       16                        5
  tarkas                         2                       27
  tars                           2                       27
  upon                          29                       31
```

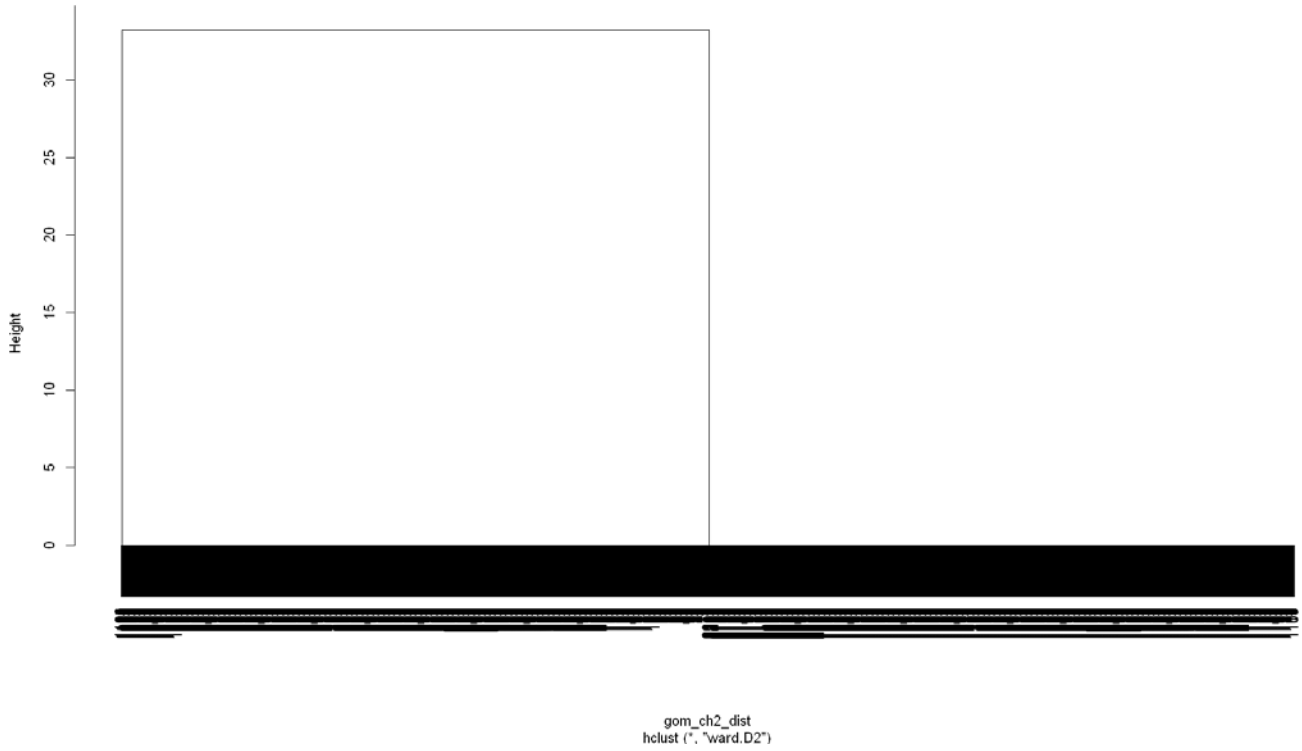
Figure 19. Inspect TDM without Stop words

```
> gom_ch1_df <- as.data.frame(gom_ch_stop_TDM[[1]])
> gom_ch2_df <- as.data.frame(gom_ch_stop_TDM[[2]])
> gom_ch1_dist <- dist(gom_ch1_df)
> gom_ch2_dist <- dist(gom_ch2_df)
> gom_ch1_DG <- hclust(gom_ch1_dist, method = 'ward.D2')
> gom_ch2_DG <- hclust(gom_ch2_dist, method = 'ward.D2')
> str(gom_ch1_DG)
List of 7
 $ merge      : int [1:2209, 1:2] -4 -18 -20 -21 -22 -26 -31 -32 -36 -37 ...
 $ height     : num [1:2209] 0 0 0 0 0 0 0 0 0 0 ...
 $ order      : int [1:2210] 1106 2210 1104 2208 1105 2209 1100 2201 1101 2202 ...
 $ labels     : NULL
 $ method     : chr "ward.D2"
 $ call       : language hclust(d = gom_ch1_dist, method = "ward.D2")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
> plot(gom_ch1_DG)
> str(gom_ch2_DG)
List of 7
 $ merge      : int [1:2209, 1:2] -1 -3 -4 -5 -6 -7 -8 -9 -10 -11 ...
 $ height     : num [1:2209] 0 0 0 0 0 0 0 0 0 0 ...
 $ order      : int [1:2210] 1106 1105 1104 1103 1102 1101 1100 1099 1098 1097 ...
 $ labels     : NULL
 $ method     : chr "ward.D2"
 $ call       : language hclust(d = gom_ch2_dist, method = "ward.D2")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
> plot(gom_ch2_DG)
```

Figure 20. The Process to Draw Dendrograms



(a) Chapter 1



(b) Chapter 2

Figure 21. Cluster Dendrograms

- The resulting dendrogram is quite cluttered, so I need to eliminate more words to get a good dendrogram (Figure 23). I select the words from the rubric as well as

choosing some additional terms manually. The manually selected terms were chosen by looking through the most frequent terms (Figure 22) and identifying those that were uninformative. The result dendrograms is shown as Figure 24.

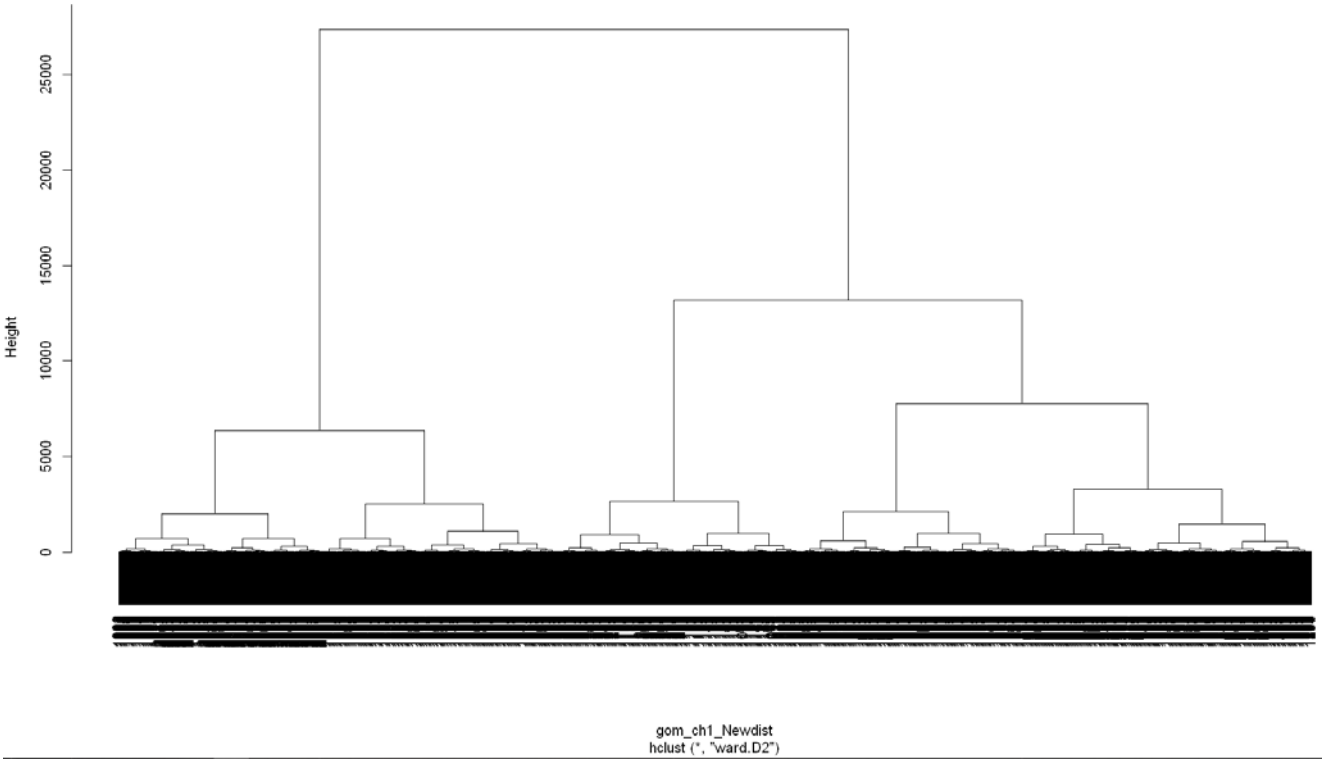
```
> findMostFreqTerms(gom_ch_stop_TDM, n = 100)
$god_of_mars_chapter1.txt
      upon      great        one    strange      green        men       herd      stood     toward       feet     forest 
        29         20         16         16         15         15         11         11         11         10         10 
    little        now      plant       back       eyes       mars     mighty     seemed     though    warrior      trees 
        10         10         10          9          9          9          9          9          9          9          8 
    turned        two  direction        eye        far      quite      river       rose        sea       time       body 
         8          8          7          7          7          7          7          7          7          7          6 
 creatures       dead       face      heads    instant        lay        man       much       side      since     single 
         6          6          6          6          6          6          6          6          6          6          6 
       ten      awful    barsoom      close   distance   earthly       felt     ground      hands     hordes     indeed 
         6          5          5          5          5          5          5          5          5          5          5 
    meadow        met      might        red remarkable      right       seen     across        air    another       arms 
         5          5          5          5          5          5          4          4          4          4          4 
   boulder      broad   creature     either    engaged        ere       even      every  evidently     glance       hair 
         4          4          4          4          4          4          4          4          4          4          4 
     heart     height    however       huge       last       left     length       long  longsword     manner   martians 
         4          4          4          4          4          4          4          4          4          4          4 
   muscles      party     planet      point       race    rapidly        saw     shrill     sprang      sward      thing 
         4          4          4          4          4          4          4          4          4          4          4 
      thus 
         4 

$god_of_mars_chapter2.txt
     upon    tarkas      tars      feet    cliffs     great       now       men      tree       far       one     might     plant 
       31        27        27        21        16        15        13        12        12        11        11        10        10 
    cliff    escape    valley      even      ever      face   opening     reach   reached    seemed      time       way   another 
        9         9         9         8         8         8         8         8         8         8         8         8         7 
     cave     close     cruel      eyes    forest     found   hundred     still      apes      back      bars      base     green 
        7         7         7         7         7         7         7         7         6         6         6         6         6 
      man    mighty     quite       set   shelter     thark  thousand    toward     trees    weight     white    across    behind 
        6         6         6         6         6         6         6         6         6         6         6         5         5 
   carter     death      door     every      fear     first      five  gorgeous    ground  interior      john    ladder      last 
        5         5         5         5         5         5         5         5         5         5         5         5         5 
      lay    length     light      like      limb     lower  possible presently      seen     stood   strange   surface     sward 
        5         5         5         5         5         5         5         5         5         5         5         5         5 
  without    ascent    attack   beneath      best  branches      came     caves      come       cut direction  distance   explore 
        5         4         4         4         4         4         4         4         4         4         4         4         4 
     felt      find  gleaming     hands    height     horde      knew      know     ledge 
        4         4         4         4         4         4         4         4         4 
```

Figure 22. Most Frequent Terms in Each Document

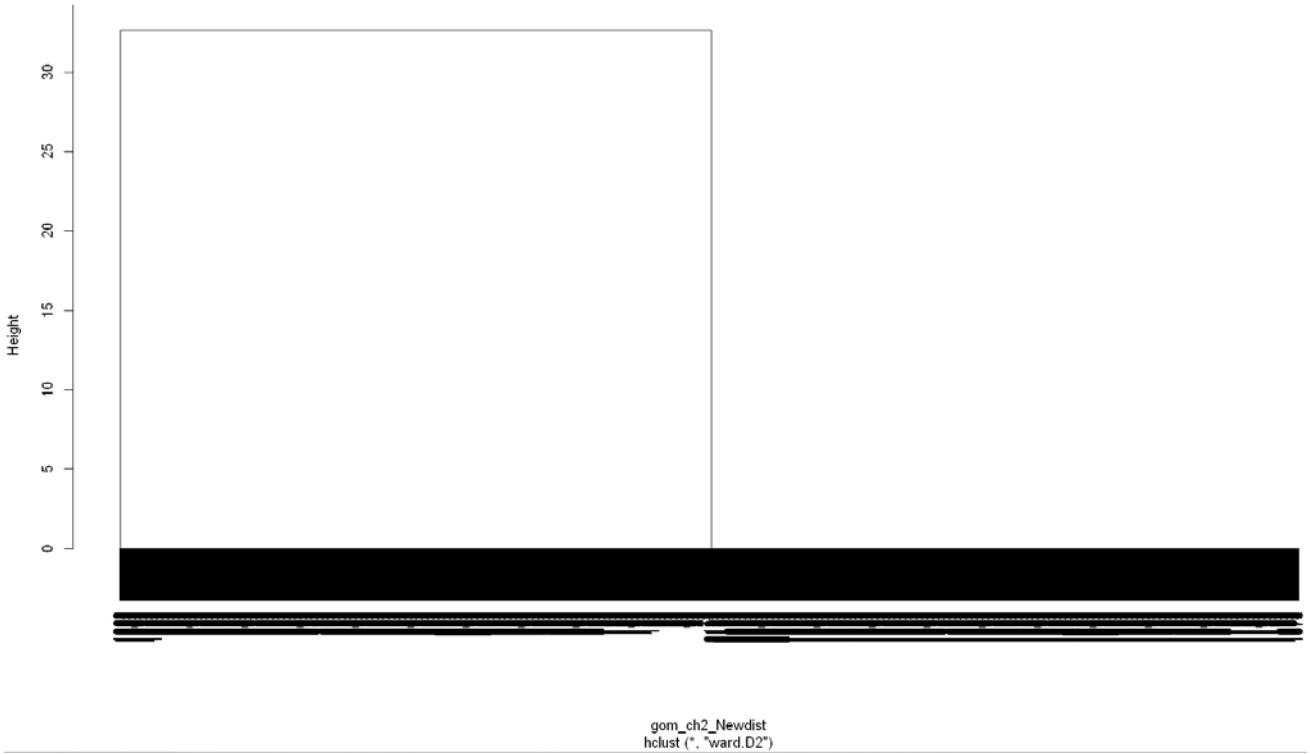
```
> newstopwords = c('one','even','a','will','may','soon',
+                  'can','as','much','just','now','certain',
+                  'quite','merely','shall','take','well','great',
+                  'though','two','quite','ten','three','four','might','right',
+                  'either','last','however','thus','without','like','upon','across',
+                  'almost','another','back','base','become','behind', 'beneath', 'better',
+                  'cause','centre')
> god_of_mars_chapters_stopnew <- tm_map(god_of_mars_chapters_stop, removeWords, newstopwords)
> gom_ch_stop_NewTDM <- TermDocumentMatrix(god_of_mars_chapters_stopnew)
> str(gom_ch_stop_NewTDM)
List of 6
 $ i       : int [1:2135] 4 5 8 9 10 11 13 14 15 18 ...
 $ j       : int [1:2135] 1 1 1 1 1 1 1 1 1 1 ...
 $ v       : num [1:2135] 1 1 1 1 1 1 1 1 2 1 ...
 $ nrow    : int 1747
 $ ncol    : int 2
 $ dimnames:List of 2
  ..$ Terms: chr [1:1747] "abandoned" "able" "absorbed" "abutted" ...
  ..$ Docs : chr [1:2] "god_of_mars_chapter1.txt" "god_of_mars_chapter2.txt"
 - attr(*, "class")= chr [1:2] "TermDocumentMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
> findMostFreqTerms(gom_ch_stop_NewTDM, n = 100)
$god_of_mars_chapter1.txt
   strange      green        men       herd      stood     toward       feet     forest     little      plant       eyes
        16         15         15         11         11         11         10         10         10         10          9
     mars     mighty     seemed    warrior      trees     turned  direction        eye        far      river       rose
         9          9          9          9          8          8          7          7          7          7          7
      sea       time       body  creatures       dead       face      heads    instant        lay        man       side
         7          7          6          6          6          6          6          6          6          6          6
    since     single      awful    barsoom      close   distance    earthly       felt     ground      hands     hordes
         6          6          5          5          5          5          5          5          5          5          5
   indeed     meadow        met        red remarkable       seen        air       arms    boulder      broad   creature
         5          5          5          5          5          5          4          4          4          4          4
  engaged        ere      every  evidently     glance       hair      heart     height       huge       left     length
         4          4          4          4          4          4          4          4          4          4          4
     long  longsword     manner   martians    muscles      party     planet      point       race    rapidly        saw
         4          4          4          4          4          4          4          4          4          4          4
    shrill     sprang      sward      thing       tiny   warriors      years appearance     attack      birds      blade
         4          4          4          4          4          4          4          3          3          3          3
    bluff    brought       came       cave    charged      cruel    crushed        cut      death  different     entire
         3          3          3          3          3          3          3          3          3          3          3
  equipped
         3
$god_of_mars_chapter2.txt
   tarkas      tars      feet    cliffs       men      tree       far     plant     cliff    escape    valley      ever      face
       27        27        21        16        12        12        11        10         9         9         9         8         8
  opening     reach   reached    seemed      time       way      cave     close     cruel      eyes    forest     found   hundred
        8         8         8         8         8         8         7         7         7         7         7         7         7
    still      apes      bars     green       man    mighty       set   shelter     thark  thousand    toward     trees    weight
        7         6         6         6         6         6         6         6         6         6         6         6         6
    white    carter     death      door     every      fear     first      five  gorgeous    ground  interior      john    ladder
        6         5         5         5         5         5         5         5         5         5         5         5         5
      lay    length     light      limb     lower  possible presently      seen     stood   strange   surface     sward    ascent
        5         5         5         5         5         5         5         5         5         5         5         5         4
   attack      best  branches      came     caves      come       cut direction  distance   explore      felt      find  gleaming
        4         4         4         4         4         4         4         4         4         4         4         4         4
    hands    height     horde      knew      know     ledge      left    little      mars     means    moment    narrow     never
        4         4         4         4         4         4         4         4         4         4         4         4         4
  quickly   rapidly       saw      side     solid    sprang  together     weird    within
        4         4         4         4         4         4         4         4         4
> gom_ch1_newdf <- as.data.frame(gom_ch_stop_NewTDM[[1]])
> gom_ch2_newdf <- as.data.frame(gom_ch_stop_NewTDM[[2]])
> gom_ch1_Newdist <- dist(gom_ch1_newdf)
> gom_ch2_Newdist <- dist(gom_ch2_newdf)
> gom_ch1_NewDG <- hclust(gom_ch1_Newdist, method = 'ward.D2')
> gom_ch2_NewDG <- hclust(gom_ch2_Newdist, method = 'ward.D2')
> inspect(gom_ch_stop_NewTDM)
<<TermDocumentMatrix (terms: 1747, documents: 2)>>
Non-/sparse entries: 2135/1359
Sparsity           : 39%
Maximal term length: 16
Weighting          : term frequency (tf)
Sample             :
        Docs
Terms    god_of_mars_chapter1.txt god_of_mars_chapter2.txt
  cliffs                        2                       16
  far                           7                       11
  feet                         10                       21
  forest                       10                        7
  green                        15                        6
  men                          15                       12
  plant                        10                       10
  strange                      16                        5
  tarkas                        2                       27
  tars                          2                       27
> plot(gom_ch1_NewDG)
> plot(gom_ch2_NewDG)
> plot(gom_ch1_NewDG)
```

Figure 23. Remove More Stop Words to Draw New Dendrograms



(a) Chapter 1



(b) Chapter 2

Figure 24. Cluster Dendrogram Result (for New Stop Word List)

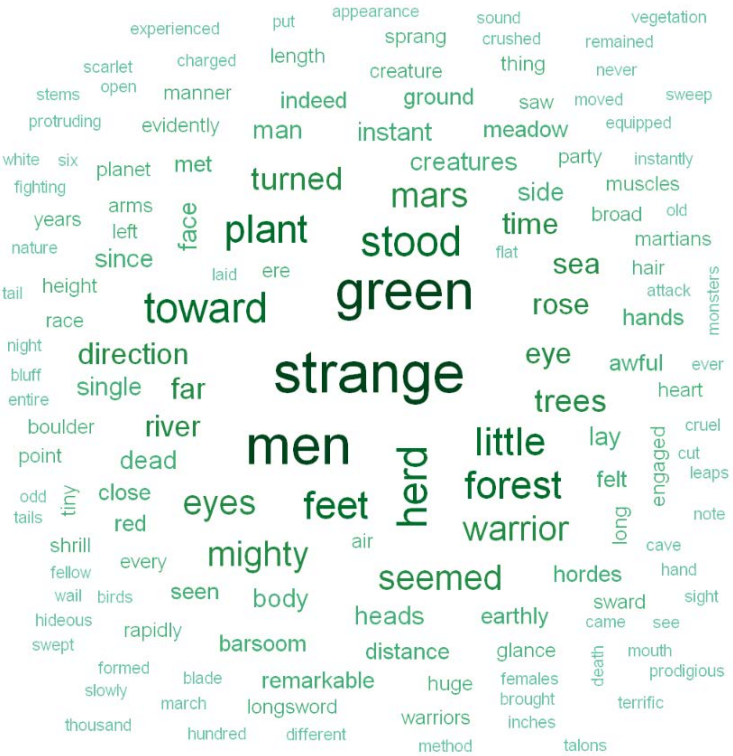
- I remove 86 (=1873-1787) words from the documents and get a new TDM as well as increasing the sparsity from 37% to 39%;

- After removing the new stop words, the most frequent terms in both chapters seem to contribute more to the understanding the theme of the book. For example, ‘Tarkas Tars’ might be a name of a man, since the two words appear the same number of times in chapters 1 and 2.
- The new dendrogram is much better compared with the old one, but it is still too difficult to interpret.

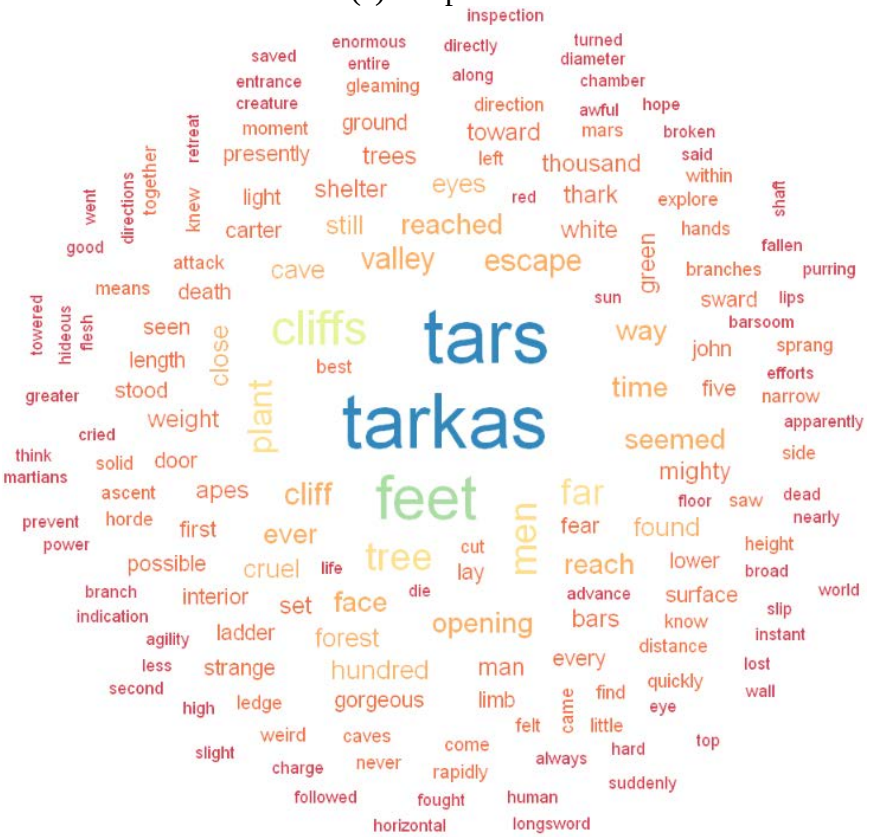
- To quickly perceive the most prominent terms, and try to get and understanding of each chapter from them, I can create word clouds (Figure 25). The result is shown in Figure 26.

```
> gom_ch1_ntf <- termFreq(god_of_mars_chapters_stopnew[[1]])
> gom_ch2_ntf <- termFreq(god_of_mars_chapters_stopnew[[2]])
> words_ch1 <- names(gom_ch1_ntf)
> words_ch2 <- names(gom_ch2_ntf)
> pal1<- brewer.pal(9,'BuGn')
> pal2<- brewer.pal(9,'Spectral')
> str(pal1)
 chr [1:9] "#F7FCFD" "#E5F5F9" "#CCECE6" "#99D8C9" "#66C2A4" "#41AE76" "#238B45" ...
> str(pal2)
 chr [1:9] "#D53E4F" "#F46D43" "#FDAE61" "#FEE08B" "#FFFFBF" "#E6F598" "#ABDDA4" ...
> png("wordcloud_packages.png", width=12,height=8, units='in', res=500)
> gom_WC_ch1 <- wordcloud(words_ch1, gom_ch1_ntf, colors = pal1[-(1:4)], min.freq=3,
+                         max.words=Inf, random.order=FALSE)
> gom_WC_ch2 <- wordcloud(words_ch2, gom_ch2_ntf, colors = pal2, min.freq=3,
+                         max.words=Inf, random.order=FALSE)
```

Figure 25. Create Word Cloud



(a) Chapter 1



(b) Chapter 2

Figure 26. Word Cloud of First Two Chapters

- Next, I explore methods from the ‘quanteda’ package for text analysis. Before continuing, I must eliminate all the blank lines from the document. Then I can use tokens() to tokenized the documents by line and apply dfm() to construct a sparse document-feature matrix (Figure 27).

```
> gomText1 = god_of_mars_chapters_stopnew[[1]]
> gomText2 = god_of_mars_chapters_stopnew[[2]]
> gomText1$content <- gomText1$content[gomText1$content != ""] # delete the blank lines
> gomText2$content <- gomText2$content[gomText2$content != ""]
> gomText1$content
  [1] " plant men"
  [2] "  stood   bluff   cottage   clear cold night "
  [3] " early part  march   noble hudson flowing   grey"
  [4] " silent spectre   dead river    felt   strange"
  [5] "compelling influence   mighty god  war  beloved mars "
  [6] "  long  lonesome years   implored  outstretched arms"
  [7] " carry     lost love"
  [8] " since   march night      stood  "
  [9] "arizona cave    still  lifeless body lay wrapped  "
 [10] "similitude  earthly death   felt  irresistible attraction "
> gomText2$content
  [1] " forest battle"
  [2] "tars tarkas   found  time   exchange  experiences  "
  [3] "stood     boulder surrounded   corpses  "
  [4] "grotesque assailants    directions   broad valley "
  [5] "streaming  perfect torrent  terrifying creatures  response  "
  [6] "weird call   strange figure far  us"
  [7] "come cried tars tarkas  must make   cliffs   lies"
  [8] "  hope   temporary escape    find  cave  "
  [9] "narrow ledge    defend  ever   motley unarmed"
 [10] "horde"
> gomTokens1 = quanteda::tokens(gomText1$content)
> gomTokens2 = quanteda::tokens(gomText2$content)
> str(gomTokens1)
List of 361
 $ text1  : chr [1:2] "plant" "men"
 $ text2  : chr [1:6] "stood" "bluff" "cottage" "clear" ...
 $ text3  : chr [1:7] "early" "part" "march" "noble" ...
 $ text4  : chr [1:6] "silent" "spectre" "dead" "river" ...
 $ text5  : chr [1:7] "compelling" "influence" "mighty" "god" ...
 $ text6  : chr [1:6] "long" "lonesome" "years" "implored" ...
 $ text7  : chr [1:3] "carry" "lost" "love"
 $ text8  : chr [1:4] "since" "march" "night" "stood"
 $ text9  : chr [1:7] "arizona" "cave" "still" "lifeless" ...
 $ text10 : chr [1:6] "similitude" "earthly" "death" "felt" ...
```

```
 [list output truncated]
- attr(*, "types")= chr [1:1071] "plant" "men" "stood" "bluff" ...
- attr(*, "padding")= logi FALSE
- attr(*, "class")= chr "tokens"
- attr(*, "docvars")='data.frame':     361 obs. of  3 variables:
 ..$ docname_: chr [1:361] "text1" "text2" "text3" "text4" ...
 ..$ docid_  : Factor w/ 361 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
 ..$ segid_  : int [1:361] 1 1 1 1 1 1 1 1 1 1 ...
- attr(*, "meta")=List of 3
 ..$ system:List of 5
 .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
 .. .. ..$ : int [1:3] 3 2 1
 .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
 .. .. ..$ : int [1:3] 4 1 3
 .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
 .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
 .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
 .. ..$ created        : Date[1:1], format: "2022-04-28"
 ..$ object:List of 6
 .. ..$ unit        : chr "documents"
 .. ..$ what        : chr "word"
 .. ..$ ngram       : int 1
 .. ..$ skip        : int 0
 .. ..$ concatenator: chr "_"
 .. ..$ summary     :List of 2
 .. .. ..$ hash: chr(0)
 .. .. ..$ data: NULL
 ..$ user  : list()
```

```
> str(gomTokens2)
List of 384
 $ text1  : chr [1:2] "forest" "battle"
 $ text2  : chr [1:6] "tars" "tarkas" "found" "time" ...
 $ text3  : chr [1:4] "stood" "boulder" "surrounded" "corpses"
 $ text4  : chr [1:5] "grotesque" "assailants" "directions" "broad" ...
 $ text5  : chr [1:6] "streaming" "perfect" "torrent" "terrifying" ...
 $ text6  : chr [1:6] "weird" "call" "strange" "figure" ...
 $ text7  : chr [1:8] "come" "cried" "tars" "tarkas" ...
 $ text8  : chr [1:5] "hope" "temporary" "escape" "find" ...
 $ text9  : chr [1:6] "narrow" "ledge" "defend" "ever" ...
 $ text10 : chr "horde"
```

```
 [list output truncated]
- attr(*, "types")= chr [1:1068] "forest" "battle" "tars" "tarkas" ...
- attr(*, "padding")= logi FALSE
- attr(*, "class")= chr "tokens"
- attr(*, "docvars")='data.frame':     384 obs. of  3 variables:
 ..$ docname_: chr [1:384] "text1" "text2" "text3" "text4" ...
 ..$ docid_  : Factor w/ 384 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
 ..$ segid_  : int [1:384] 1 1 1 1 1 1 1 1 1 1 ...
- attr(*, "meta")=List of 3
 ..$ system:List of 5
 .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
 .. .. ..$ : int [1:3] 3 2 1
 .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
 .. .. ..$ : int [1:3] 4 1 3
 .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
 .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
 .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
 .. ..$ created        : Date[1:1], format: "2022-04-28"
 ..$ object:List of 6
 .. ..$ unit        : chr "documents"
 .. ..$ what        : chr "word"
 .. ..$ ngram       : int 1
 .. ..$ skip        : int 0
 .. ..$ concatenator: chr "_"
 .. ..$ summary     :List of 2
 .. .. ..$ hash: chr(0)
 .. .. ..$ data: NULL
 ..$ user  : list()
```

```
> gomDFM1 = quanteda::dfm(gomTokens1)
> gomDFM2 = quanteda::dfm(gomTokens2)
> str(gomDFM1)
Formal class 'dfm' [package "quanteda"] with 8 slots
  ..@ docvars :'data.frame':    361 obs. of  3 variables:
  .. ..$ docname_: chr [1:361] "text1" "text2" "text3" "text4" ...
  .. ..$ docid_  : Factor w/ 361 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ segid_  : int [1:361] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ meta    :List of 3
  .. ..$ system:List of 5
  .. .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 3 2 1
  .. .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 4 1 3
  .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
  .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
  .. .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
  .. .. ..$ created        : Date[1:1], format: "2022-04-28"
  .. ..$ object:List of 9
  .. .. ..$ unit        : chr "documents"
  .. .. ..$ what        : chr "word"
  .. .. ..$ ngram       : int 1
  .. .. ..$ skip        : int 0
  .. .. ..$ concatenator: chr "_"
  .. .. ..$ weight_tf   :List of 3
  .. .. .. ..$ scheme: chr "count"
  .. .. .. ..$ base  : NULL
  .. .. .. ..$ k     : NULL
  .. .. ..$ weight_df   :List of 5
  .. .. .. ..$ scheme   : chr "unary"
  .. .. .. ..$ base     : NULL
  .. .. .. ..$ c        : NULL
  .. .. .. ..$ smoothing: NULL
  .. .. .. ..$ threshold: NULL
  .. .. ..$ smooth      : num 0
  .. .. ..$ summary     :List of 2
  .. .. .. ..$ hash: chr(0)
  .. .. .. ..$ data: NULL
  .. ..$ user  : list()
  ..@ i       : int [1:1806] 0 191 212 242 243 249 264 267 276 319 ...
  ..@ p       : int [1:1072] 0 10 25 36 39 40 42 44 47 48 ...
  ..@ Dim     : int [1:2] 361 1071
  ..@ Dimnames:List of 2
  .. ..$ docs    : chr [1:361] "text1" "text2" "text3" "text4" ...
  .. ..$ features: chr [1:1071] "plant" "men" "stood" "bluff" ...
  ..@ x       : num [1:1806] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ factors : list()
```

```
> str(gomDFM2)
Formal class 'dfm' [package "quanteda"] with 8 slots
  ..@ docvars :'data.frame':    384 obs. of  3 variables:
  .. ..$ docname_: chr [1:384] "text1" "text2" "text3" "text4" ...
  .. ..$ docid_  : Factor w/ 384 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ segid_  : int [1:384] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ meta    :List of 3
  .. ..$ system:List of 5
  .. .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 3 2 1
  .. .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 4 1 3
  .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
  .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
  .. .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
  .. .. ..$ created        : Date[1:1], format: "2022-04-28"
  .. ..$ object:List of 9
  .. .. ..$ unit        : chr "documents"
  .. .. ..$ what        : chr "word"
  .. .. ..$ ngram       : int 1
  .. .. ..$ skip        : int 0
  .. .. ..$ concatenator: chr "_"
  .. .. ..$ weight_tf   :List of 3
  .. .. .. ..$ scheme: chr "count"
  .. .. .. ..$ base  : NULL
  .. .. .. ..$ k     : NULL
  .. .. ..$ weight_df   :List of 5
  .. .. .. ..$ scheme   : chr "unary"
  .. .. .. ..$ base     : NULL
  .. .. .. ..$ c        : NULL
  .. .. .. ..$ smoothing: NULL
  .. .. .. ..$ threshold: NULL
  .. .. ..$ smooth      : num 0
  .. .. ..$ summary     :List of 2
  .. .. .. ..$ hash: chr(0)
  .. .. .. ..$ data: NULL
  .. ..$ user  : list()
  ..@ i       : int [1:1929] 0 33 43 57 97 109 336 0 1 6 ...
  ..@ p       : int [1:1069] 0 7 8 35 62 69 76 77 79 84 ...
  ..@ Dim     : int [1:2] 384 1068
  ..@ Dimnames:List of 2
  .. ..$ docs    : chr [1:384] "text1" "text2" "text3" "text4" ...
  .. ..$ features: chr [1:1068] "forest" "battle" "tars" "tarkas" ...
  ..@ x       : num [1:1929] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ factors : list()
```

Figure 27. Data Tokenization and Create dfm

- Then get the frequency of terms in dmf and assign weights to each words based on frequency(Figure 28).

```
> str(gomDocFreq1)
 Named int [1:1071] 10 15 11 3 1 2 2 3 1 1 ...
 - attr(*, "names")= chr [1:1071] "plant" "men" "stood" "bluff" ...
> str(gomDocFreq2)
 Named int [1:1068] 7 1 27 27 7 7 1 2 5 2 ...
 - attr(*, "names")= chr [1:1068] "forest" "battle" "tars" "tarkas" ...
> gomDocFreq1
    plant       men     stood     bluff   cottage     clear 
       10        15        11         3         1         2 
     cold     night     early      part     march    noble 
        2         3         1         1         3         2 
   hudson   flowing      grey    silent   spectre      dead 
        2         1         1         1         1         6 
    river      felt   strange compelling influence    mighty 
        7         5        16         1         1         9 
      god       war   beloved      mars      long  lonesome 
        2         2         1         9         4         1 
```

```
> gomDocFreq2
         forest          battle            tars          tarkas           found            time
              7               1              27              27               7               7
       exchange     experiences          stood         boulder      surrounded         corpses
              1               2               5               2               1               1
      grotesque      assailants      directions           broad          valley       streaming
              2               2               3               3               9               1
        perfect         torrent      terrifying      creatures        response          weird
              2               1               2               2               2               4
           call         strange          figure             far              us            come
              1               5               1              11              22               4
> gomWeights1 = dfm_weight(gomDFM1)
> gomWeights2 = dfm_weight(gomDFM2)
> str(gomWeights1)
Formal class 'dfm' [package "quanteda"] with 8 slots
  ..@ docvars :'data.frame':    361 obs. of  3 variables:
  .. ..$ docname_: chr [1:361] "text1" "text2" "text3" "text4" ...
  .. ..$ docid_  : Factor w/ 361 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ segid_  : int [1:361] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ meta    :List of 3
  .. ..$ system:List of 5
  .. .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 3 2 1
  .. .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 4 1 3
  .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
  .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
  .. .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
  .. .. ..$ created        : Date[1:1], format: "2022-04-28"
  .. ..$ object:List of 9
  .. .. ..$ unit        : chr "documents"
  .. .. ..$ what        : chr "word"
  .. .. ..$ ngram       : int 1
  .. .. ..$ skip        : int 0
  .. .. ..$ concatenator: chr "_"
  .. .. ..$ weight_tf   :List of 3
  .. .. .. ..$ scheme: chr "count"
  .. .. .. ..$ base  : NULL
  .. .. .. ..$ k     : NULL
  .. .. ..$ weight_df   :List of 5
  .. .. .. ..$ scheme   : chr "unary"
  .. .. .. ..$ base     : NULL
  .. .. .. ..$ c        : NULL
  .. .. .. ..$ smoothing: NULL
  .. .. .. ..$ threshold: NULL
  .. .. ..$ smooth      : num 0
  .. .. ..$ summary     :List of 2
  .. .. .. ..$ hash: chr(0)
  .. .. .. ..$ data: NULL
  .. ..$ user  : list()
  ..@ i       : int [1:1806] 0 191 212 242 243 249 264 267 276 319 ...
  ..@ p       : int [1:1072] 0 10 25 36 39 40 42 44 47 48 ...
  ..@ Dim     : int [1:2] 361 1071
  ..@ Dimnames:List of 2
  .. ..$ docs    : chr [1:361] "text1" "text2" "text3" "text4" ...
  .. ..$ features: chr [1:1071] "plant" "men" "stood" "bluff" ...
  ..@ x       : num [1:1806] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ factors : list()
```

```
> str(gomWeights2)
Formal class 'dfm' [package "quanteda"] with 8 slots
  ..@ docvars :'data.frame':    384 obs. of  3 variables:
  .. ..$ docname_: chr [1:384] "text1" "text2" "text3" "text4" ...
  .. ..$ docid_  : Factor w/ 384 levels "text1","text2",..: 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ segid_  : int [1:384] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ meta    :List of 3
  .. ..$ system:List of 5
  .. .. ..$ package-version:Classes 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 3 2 1
  .. .. ..$ r-version      :Classes 'R_system_version', 'package_version', 'numeric_version'  hidden list of 1
  .. .. .. ..$ : int [1:3] 4 1 3
  .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "46241"
  .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
  .. .. ..$ directory      : chr "C:/Users/46241/Desktop/GWU/courses/Spring 2022/CSCI 6444/R File/Project 3"
  .. .. ..$ created        : Date[1:1], format: "2022-04-28"
  .. ..$ object:List of 9
  .. .. ..$ unit        : chr "documents"
  .. .. ..$ what        : chr "word"
  .. .. ..$ ngram       : int 1
  .. .. ..$ skip        : int 0
  .. .. ..$ concatenator: chr "_"
  .. .. ..$ weight_tf   :List of 3
  .. .. .. ..$ scheme: chr "count"
  .. .. .. ..$ base  : NULL
  .. .. .. ..$ k     : NULL
  .. .. ..$ weight_df   :List of 5
  .. .. .. ..$ scheme   : chr "unary"
  .. .. .. ..$ base     : NULL
  .. .. .. ..$ c        : NULL
  .. .. .. ..$ smoothing: NULL
  .. .. .. ..$ threshold: NULL
  .. .. ..$ smooth      : num 0
  .. .. ..$ summary     :List of 2
  .. .. .. ..$ hash: chr(0)
  .. .. .. ..$ data: NULL
  .. ..$ user  : list()
  ..@ i       : int [1:1929] 0 33 43 57 97 109 336 0 1 6 ...
  ..@ p       : int [1:1069] 0 7 8 35 62 69 76 77 79 84 ...
  ..@ Dim     : int [1:2] 384 1068
  ..@ Dimnames:List of 2
  .. ..$ docs    : chr [1:384] "text1" "text2" "text3" "text4" ...
  .. ..$ features: chr [1:1068] "forest" "battle" "tars" "tarkas" ...
  ..@ x       : num [1:1929] 1 1 1 1 1 1 1 1 1 1 ...
  ..@ factors : list()
> gomWeights1
Document-feature matrix of: 361 documents, 1,071 features (99.53% sparse) and 0 docvars.
       features
docs    plant men stood bluff cottage clear cold night early part
  text1     1   1     0     0       0     0    0     0     0    0
  text2     0   0     1     1       1     1    1     1     0    0
  text3     0   0     0     0       0     0    0     0     1    1
  text4     0   0     0     0       0     0    0     0     0    0
  text5     0   0     0     0       0     0    0     0     0    0
  text6     0   0     0     0       0     0    0     0     0    0
[ reached max_ndoc ... 355 more documents, reached max_nfeat ... 1,061 more features ]
> gomWeights2
Document-feature matrix of: 384 documents, 1,068 features (99.53% sparse) and 0 docvars.
       features
docs    forest battle tars tarkas found time exchange experiences stood boulder
  text1      1      1    0      0     0    0        0           0     0       0
  text2      0      0    1      1     1    1        1           1     0       0
  text3      0      0    0      0     0    0        0           0     1       1
  text4      0      0    0      0     0    0        0           0     0       0
  text5      0      0    0      0     0    0        0           0     0       0
  text6      0      0    0      0     0    0        0           0     0       0
[ reached max_ndoc ... 378 more documents, reached max_nfeat ... 1,058 more features ]
```

Figure 28. Count Frequency and Assign Weights to Terms

- Finally, I compute the term frequency-inverse document frequency (tf-idf) score, with full control over options (Figure 29).

```
> gomTFIDF1 = dfm_tfidf(gomDFM1, scheme_tf = "count", scheme_df = "inverse")
> gomTFIDF2 = dfm_tfidf(gomDFM2, scheme_tf = "count", scheme_df = "inverse")
> gomTFIDF1
Document-feature matrix of: 361 documents, 1,071 features (99.53% sparse) and 0 docvars.
       features
docs       plant      men    stood    bluff  cottage    clear     cold    night    early     part
  text1 1.557507 1.381416 0        0        0        0        0        0        0        0
  text2 0        0        1.516115 2.080386 2.557507 2.256477 2.256477 2.080386 0        0
  text3 0        0        0        0        0        0        0        0        2.557507 2.557507
  text4 0        0        0        0        0        0        0        0        0        0
  text5 0        0        0        0        0        0        0        0        0        0
  text6 0        0        0        0        0        0        0        0        0        0
[ reached max_ndoc ... 355 more documents, reached max_nfeat ... 1,061 more features ]
> gomTFIDF2
Document-feature matrix of: 384 documents, 1,068 features (99.53% sparse) and 0 docvars.
       features
docs      forest   battle     tars   tarkas    found     time exchange experiences    stood  boulder
  text1 1.739233 2.584331 0        0        0        0        0           0        0        0
  text2 0        0        1.152967 1.152967 1.739233 1.739233 2.584331    2.283301 0        0
  text3 0        0        0        0        0        0        0           0        1.885361 2.283301
  text4 0        0        0        0        0        0        0           0        0        0
  text5 0        0        0        0        0        0        0           0        0        0
  text6 0        0        0        0        0        0        0           0        0        0
[ reached max_ndoc ... 378 more documents, reached max_nfeat ... 1,058 more features ]
```

Figure 29. Compute tf-idf Score

  - There are 361 lines with 1071 features in chapter 1 and 384 lines with 1068 features in chapter 2 (with blank lines removed).
  - For interpreting the tf-idf score, the larger the score, the more useless the word. So, by calling dfm_tfidf function, I can filter the noise using these scores. Like in chapter 2, ‘tars’ and ‘tarkas’ are more important than ‘forest’ or ‘battle’.

- Then I apply ‘syuzhet’ package for sentimental analysis. After reading the file as large string, I can easily get each sentence using get_sentences().Next I can get the sentiment using the default 'syuzhet' term weights (Figure 30).

```
> gomAsString1 = get_text_as_string("./chapters/god_of_mars_chapter1.txt")
> gomAsString2 = get_text_as_string("./chapters/god_of_mars_chapter2.txt")
> gomAsString1
 THE PLANT MEN   As I stood upon the bluff before my cottage on that clear cold night in the early part of Marc
h, 1886, the noble Hudson flowing like the grey and silent spectre of a dead river below me, I felt again the st
range, compelling influence of the mighty god of war, my beloved Mars, which for ten long and lonesome years I h
ad implored with outstretched arms to carry me back to my lost love.  Not since that other March night in 1866,
 when I had stood without that Arizona cave in which my still and lifeless body lay wrapped in the similitude of
 earthly death had I felt the irresistible attraction of the god of my profession.  With arms outstretched towar
> gomAsString2
 A FOREST BATTLE   Tars Tarkas and I found no time for an exchange of experiences as we stood there before the g
reat boulder surrounded by the corpses of our grotesque assailants, for from all directions down the broad valle
y was streaming a perfect torrent of terrifying creatures in response to the weird call of the strange figure fa
r above us.  "Come," cried Tars Tarkas, "we must make for the cliffs.  There lies our only hope of even temporar
y escape; there we may find a cave or a narrow ledge which two may defend for ever against this motley, unarmed
 horde."  Together we raced across the scarlet sward, I timing my speed that I might not outdistance my slower c
```

```
> gomS1 = get_sentences(gomAsString1)
> gomS2 = get_sentences(gomAsString2)
> gomS1
 [1] "THE PLANT MEN   As I stood upon the bluff before my cottage on that clear cold night in the early part of
 March, 1886, the noble Hudson flowing like the grey and silent spectre of a dead river below me, I felt again t
he strange, compelling influence of the mighty god of war, my beloved Mars, which for ten long and lonesome year
s I had implored with outstretched arms to carry me back to my lost love."



 [2] "Not since that other March night in 1866, when I had stood without that Arizona cave in which my still and
 lifeless body lay wrapped in the similitude of earthly death had I felt the irresistible attraction of the god
 of my profession."
```

[Here 97 sentences are omitted]

```
> gomS2
  [1] "A FOREST BATTLE   Tars Tarkas and I found no time for an exchange of experiences as we stood there before
 the great boulder surrounded by the corpses of our grotesque assailants, for from all directions down the broad
 valley was streaming a perfect torrent of terrifying creatures in response to the weird call of the strange fig
ure far above us."
  [2] "\"Come,\" cried Tars Tarkas, \"we must make for the cliffs."
```

[Here 122 sentences are omitted]

```
> gomSSentiment1 = get_sentiment(gomS1, "syuzhet")
> gomSSentiment2 = get_sentiment(gomS2, "syuzhet")
> gomSSentiment1
 [1] -0.05  0.50  0.70 -2.65 -3.70 -1.25 -1.90 -0.70 -2.75  0.00 -0.15  1.50  1.75 -0.50  1.20 -0.65  0.60
[18]  0.50  0.40 -0.25  0.25  4.85  1.55  1.40  1.10  0.75 -0.25  0.45  1.30  1.25  2.50  1.10 -1.00  1.20
[35]  0.00  0.65  2.50 -0.65 -1.50 -2.50 -0.50 -0.75 -1.45 -0.50  0.40 -0.75  0.90  0.50  0.20  0.15  0.70
[52]  0.90 -0.40 -2.70  0.00  0.40 -0.90  1.25  1.35  1.20 -1.65  0.50 -3.70  0.35  0.50 -1.25 -0.95 -1.10
[69] -0.15 -0.25  0.50 -1.25 -3.45 -1.60  1.15 -2.15 -0.65  2.45 -1.25 -0.10 -0.85 -1.15 -1.00 -1.00 -4.40
[86]  0.35 -3.55  0.95 -0.80 -2.95  0.05 -0.05  1.30  0.25  0.00  0.40 -1.05  1.00  3.45
> gomSSentiment2
  [1] -1.30 -1.00 -1.50  0.25 -0.10  0.20  3.05  0.40 -0.45 -2.00 -2.10 -0.30 -0.10 -0.50  0.75 -0.75 -0.35
 [18]  1.00 -2.75 -1.50  3.75 -0.05  1.00 -0.20  1.55  2.70  1.85  2.05  0.00 -0.50 -1.55  0.95 -0.90 -1.35
 [35]  0.55  2.45  0.80 -0.50 -2.05  3.65 -1.15 -0.25 -0.10 -2.50  1.25  0.00  0.50  0.25 -0.75 -3.75 -1.75
 [52] -2.20  0.35 -1.05 -0.35 -2.90  0.30 -0.10  0.40 -0.75 -0.50 -2.50 -0.75 -0.25 -4.55  0.80  0.50 -1.00
 [69] -2.50  0.05 -3.50 -2.00 -2.95  1.40 -3.15 -3.35  0.15  1.55 -5.05 -1.80 -3.95 -1.35 -0.75 -0.55  0.00
 [86]  0.90 -0.90  0.00  0.75  0.65  0.70  1.00  1.20 -2.00  0.00  1.30  0.15  0.70  0.70  0.60 -0.05  1.75
[103]  0.00  0.75 -0.40  1.70 -0.15  1.10  1.15  1.10 -1.25  1.35  0.00  0.00 -2.15 -0.15  0.25  0.10  0.00
[120]  0.00  0.50 -3.55 -1.75 -1.45
```

Figure 30. Data Transformation and Get Sentiment

- By calling get_sentiment function, I can get an idea about the sentiment of each sentence in documents. The larger and more positive number is, the more positive sentiment the sentence is. For example, the last sentence of chapter 1 may have the most positive sentiment in the whole chapter.

- I also check the sentiment dictionary and calculate the sum and means of the values of the sentiment vector (Figure 31).

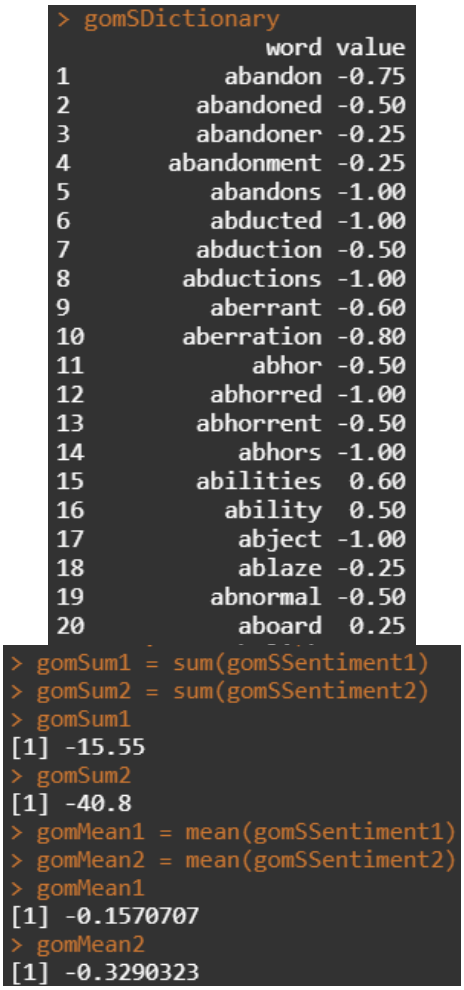


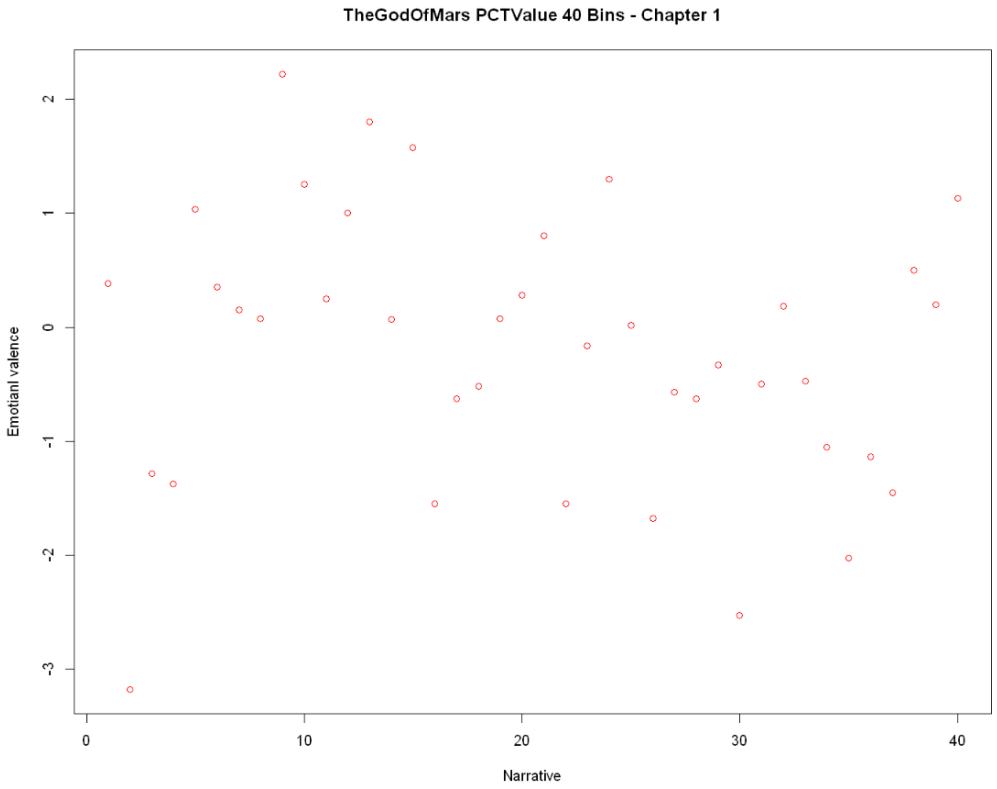
Figure 31. Sentiment Dictionary and Basic Statistic Value of Data

- Through the sentiment dictionary, I can figure out the basis of the sentiment calculation
- Both chapters are in negative sentiment with negative sum and mean value of sentiment vector, while chapter 2 seeming especially negative.

- Finally, I apply get_percentage_values function to compare the shape of one trajectory to another (Figure 32) and draw the plot (Figure 33). The sentences of each chapter are aggregated into 40 equally sized bins. This means in Chapter 2 the average of 3 sentences determines the sentiment for that bin.

```
> gomSSentimentPctvalue1 = get_percentage_values(gomSSentiment1, bin=40)
> gomSSentimentPctvalue2 = get_percentage_values(gomSSentiment2, bin=40)
> structure(gomSSentimentPctvalue1)
         1          2          3          4          5          6          7          8          9
 0.38333333 -3.17500000 -1.28333333 -1.37500000  1.03333333  0.35000000  0.15000000  0.07500000  2.21666667
        10         11         12         13         14         15         16         17         18
 1.25000000  0.25000000  1.00000000  1.80000000  0.06666667  1.57500000 -1.55000000 -0.62500000 -0.51666667
        19         20         21         22         23         24         25         26         27
 0.07500000  0.28333333  0.80000000 -1.55000000 -0.16666667  1.30000000  0.01666667 -1.67500000 -0.56666667
        28         29         30         31         32         33         34         35         36
-0.62500000 -0.33333333 -2.52500000 -0.50000000  0.18333333 -0.47500000 -1.05000000 -2.02500000 -1.13333333
        37         38         39         40
-1.45000000  0.50000000  0.20000000  1.13333333
> structure(gomSSentimentPctvalue2)
         1          2          3          4          5          6          7          8          9
-0.88750000  1.05000000 -0.68333333 -0.83333333 -0.16666667 -0.70000000  0.73333333  0.78333333  2.20000000
        10         11         12         13         14         15         16         17         18
-0.68333333 -0.43333333  1.26666667  0.36666667 -1.00000000  0.58333333 -1.41666667 -1.20000000 -1.43333333
        19         20         21         22         23         24         25         26         27
 0.20000000 -1.25000000 -1.85000000  0.10000000 -1.98333333 -1.18333333 -2.11666667 -1.76666667 -1.65000000
        28         29         30         31         32         33         34         35         36
 0.00000000  0.46666667  0.96666667 -0.23333333  0.51666667  0.76666667  0.11666667  0.88333333  0.33333333
        37         38         39         40
 0.45000000 -0.68333333  0.03333333 -1.56250000
> plot(gomSSentimentPctvalue1, main="TheGodOfMars PCTValue 40 Bins - Chapter 1", xlab="Narrative", ylab="Emotian
l valence", col='red')
> plot(gomSSentimentPctvalue2, main="TheGodOfMars PCTValue 40 Bins - Chapter 2", xlab="Narrative", ylab="Emotian
l valence", col='red')
```

Figure 32. Get Percentage Values and Draw Trajectory Plot



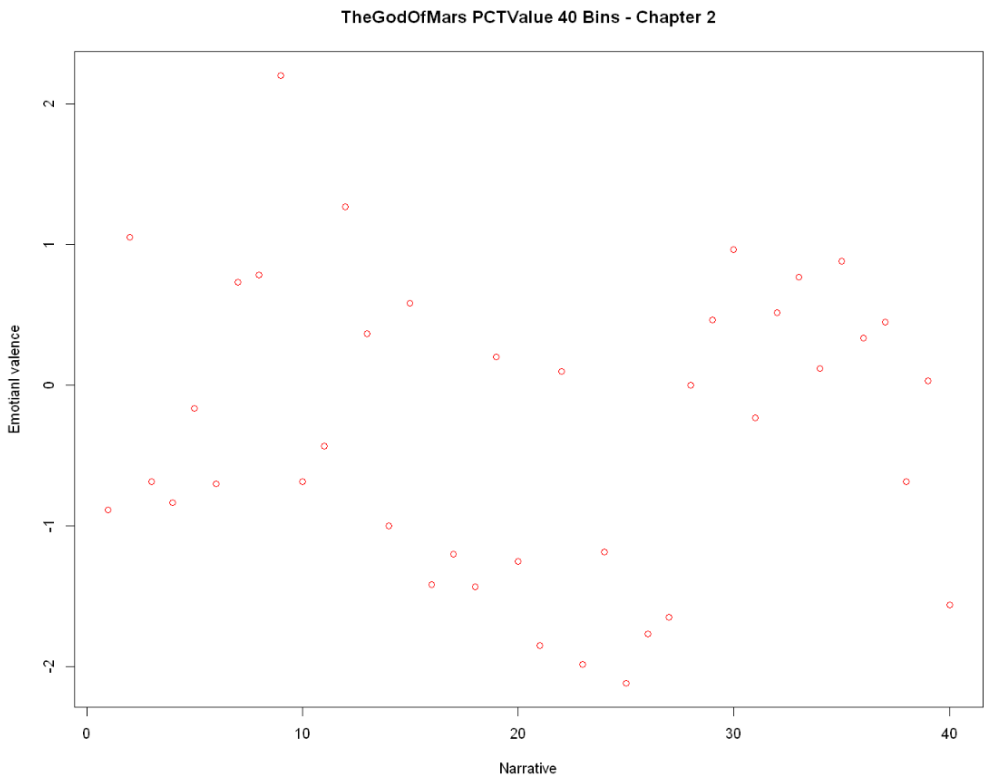


Figure 33. Trajectory Plot by Percentage Value

- From the plots, I can get the distribution of emotions as story goes by.
- There's a positive emotional intensity at the first 5% to 10% of chapter 1, but after that, sentiment becomes negative.
- Chapter 2 appears to contain more negative sentiment than positive, which confirms my results from earlier.

Part 4 Explore Additional Functions

The remaining part of my assignment was to explore additional functions from each package used in the analysis. I started with the tm function, by applying the stripWhiteSpace() (Figure 34) and tm_term_score() (Figure 35) functions to the corpus. The stripWhiteSpace function reduces any extra spaces between words into a maximum of one space. This can be seen easily in representation below. The tm_term_score() function returns the number of times a specified term appears in each chapter. Below are the term scores for 'tree', 'book', and 'cliff'. I can see from these functions that 'tree' and 'cliff' appear much more in chapter 2 than in chapter 1, and 'book' does not appear in either chapter.

```
> god_of_mars_tm_ws <- tm::tm_map(god_of_mars_chapters_stop, content_transformer(stripWhitespace))
> god_of_mars_tm_ws[[1]][1]
$content
  [1] ""                                                  " plant men"
  [3] ""                                                  ""
  [5] " stood upon bluff cottage clear cold night "       " early part march noble hudson flowing like grey"
  [7] " silent spectre dead river felt strange"           "compelling influence mighty god war beloved mars "
  [9] " ten long lonesome years implored outstretched arms"  " carry back lost love"
 [11] ""                                                  " since march night stood without "
 [13] "arizona cave still lifeless body lay wrapped "     "similitude earthly death felt irresistible attraction "
 [15] " god profession"                                   ""
 [17] " arms outstretched toward red eye great star stood"  "praying return strange power twice drawn "
 [19] " immensity space praying prayed thousand"          "nights long ten years waited hoped"
 [21] ""                                                  "suddenly qualm nausea swept senses swam knees gave"
 [23] "beneath pitched headlong ground upon verge "       " dizzy bluff"
```

Figure 34. tm::stripWhiteSpace

```
> # tm_term_score (returns the number of times a term appears in each document)
> tm_term_score(gom_ch_stop_TDM, terms = 'tree')
god_of_mars_chapter1.txt god_of_mars_chapter2.txt
                       2                       12
> tm_term_score(gom_ch_stop_TDM, terms = 'book')
god_of_mars_chapter1.txt god_of_mars_chapter2.txt
                       0                        0
> tm_term_score(gom_ch_stop_TDM, terms = 'cliff')
god_of_mars_chapter1.txt god_of_mars_chapter2.txt
                       1                        9
```

Figure 35. tm::tm_term_score() for 'tree', 'book', and 'cliff'

Next, I explored functions from the quanteda function. The functions explored for quanteda were tokens_sample() (Figure 36), tokens_ngrams() (Figure 37), and topfeatures() (Figure 38). Each of these functions works from the tokens data type from quanteda and was applied to the chapter 1 tokens from earlier.

```
> set.seed(123)
> gom_ch1_sample = tokens_sample(gomTokens1, size = 100)
> gom_ch1_sample
Tokens consisting of 100 documents.
text179 :
[1] "fighting"    "point"       "stepping"    "hidingplace"

text14 :
[1] "immensity" "space"     "praying"   "prayed"    "thousand"

text195 :
[1] "instantly" "every"     "eye"       "turned"    "toward"    "member"    "herd"      "large"
```

Figure 36. quanteda:: tokens_sample() with sample size = 100

```
> n_grams_quanteda = tokens_ngrams(gom_ch1_sample)
> n_grams_quanteda
Tokens consisting of 100 documents.
text179 :
[1] "fighting_point"       "point_stepping"       "stepping_hidingplace"

text14 :
[1] "immensity_space" "space_praying"   "praying_prayed"  "prayed_thousand"

text195 :
[1] "instantly_every" "every_eye"       "eye_turned"      "turned_toward"   "toward_member"   "member_herd"     "herd_large"
```

Figure 37. quanteda::tokens_ngrams() with n=2 (default, from sample)

```
> topfeatures(dfm(n_grams_quanteda))
     plant_men    men_barsoom   mighty_river  green_warrior    warrior_put
             3              3              2              2              2
```

Figure 38. quanteda::topfeatures() from ngrams

- I take a random sample of 100 lines from chapter 1
- Then, the ngrams are created with n=2. The ngrams from the first 3 lines in the sample are shown
- topfeatures() allows me to see the top ngrams in my sample set. In this case I see recognizable phrases such as 'plant men', 'mighty river', and 'green warrior'

Finally, the last package left to explore is the syuzhet package. For this package, I use the get_nrc_sentiment() (Figure 39) and mixed_messages() (Figure 40). Both of these functions are applied to the chapter 1 sentences from earlier in the analysis. The get_nrc_sentiment function returns the sentiment scores for each sentence, using the nrc sentiment definitions. The mixed_messages function calculates the emotional entropy of a string, that this the magnitude to which a sentences of document changes sentiment.

```
> get_nrc_sentiment(gomS1)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     1            2       0    3   3       2        0     3        6        7
2     1            2       1    3   1       2        1     1        2        4
3     0            2       0    0   1       0        0     1        0        1
4     0            1       1    1   0       0        1     1        4        0
5     2            0       2    6   2       6        2     1        8        5
```

Figure 39.syuzhet::get_nrc_sentiment ()

```
> # get the 'emotional entropy'
> mixed_messages(gomS1)
       entropy metric_entropy
  0.9978467548   0.0002383202
> gomS1[4]
[1] "Suddenly a qualm of nausea swept over me, my senses swam, my knees gave
 beneath me and I pitched headlong to the ground upon the very verge of the
 dizzy bluff."
> mixed_messages(gomS1[4])
       entropy metric_entropy
             0              0
```

Figure 40. syuzhet::mixed_messages() for Ch1 and Ch1 4th sentence

- The nrc_sentiment() shows the number of terms of each sentiment in the sentence
- The mixed_messages() score for the chapter is high. This implies there are a wide range of emotions that occur during the chapter.
- The emotional entropy score for sentence 4 is low. Comparing this to the nrc_sentiment() results it makes sense, given most terms are negative.

Part 5 Conclusion

For this project, I loaded the document "*The Gods of Mars*" and separated first two chapters into a VCorpus for text analysis. By calling 'str' and 'inspect' functions, I can easily get the basic idea of the structure and content inside the corpus.

I wrote functions to find 10 longest words and sentences, which developed my skills on manipulating text file. Document term matrices (DTM) and term document matrices (TDM) are two ways to express the text mathematically. I practiced with creating and reading both of these.

For the data wrangling section, removing numbers and punctuations is often necessary, as well as making sure that everything is in lower case. An important indicator 'sparsity' refers to how many of the values in the matrix are zero. Meanwhile, I should try to reduce the noise in the text, i.e., useless but frequently appearing words, by removing stop words. World clouds help me to quickly perceive the most prominent terms and locate the relatively prominent themes. There are many ways to choose stop words, depending on the situation, but the packages used provide easy functionalities. The dendrogram is a good method to check if I removed enough stop words. These methods in data cleaning for text analysis are valuable skills that I have practiced.

I also applied natural language processing (NLP) techniques, like text analysis and sentiment analysis with the 'quanteda' and 'syuzhet' packages respectively. For text analysis, tf-idf score can provide me with a clear sense of which words are most important. The larger the score, the more useless the word. As for sentiment analysis, I can have a feeling about the emotions in a document according to the values after applying a sentiment dictionary. Large, positive number indicate a high amount of positive sentiment. Both the two analyses help me to have a better understanding of the theme and content of the book. These methods can easily be applied to future analysis as well.