

# Text Analysis with R for Students of Literature

Ignacio Almodovar Cárdenas and Alejandra Estrada Sanz

22/03/2022

## Contents

0. Preprocessing . . . . .	2
1. Analyse and study the occurrence of words related with love or positive feelings in general . . .	2
2. Make frequency plots . . . . .	3
3. Compare word frequency data of words like “he”, “she”, “him”, “her” and show also relative frequencies . . . . .	6
4. Make a token distribution analysis . . . . .	7
5. Identify chapter breaks . . . . .	9
7. Show some measures of lexical variety . . . . .	10
8. Calculate the Hapax Richness . . . . .	14

## 0. Preprocessing

First of all, we load our text file, we are going to analyze the book “Macbeth”.

```
library(quanteda)
```

```
Package version: 3.2.1  
Unicode version: 13.0  
ICU version: 69.1
```

Parallel computing: 8 of 8 threads used.

See <https://quanteda.io> for tutorials and examples.

```
library(readtext)  
library(stringi)  
data_macbeth <- texts(readtext("https://www.gutenberg.org/files/1533/1533-0.txt"))  
names(data_macbeth) <- "Macbeth"
```

Then, we separate content from metadata and extract the header and final leftover information.

```
start_v <- stri_locate_first_fixed(data_macbeth, "SCENE I. An open Place.")[1]  
end_v <- stri_locate_last_fixed(data_macbeth, "[_Flourish. Exeunt._]")[1]  
novel_v <- stri_sub(data_macbeth, start_v, end_v)  
novel_v = gsub("€", "", novel_v)  
novel_v = gsub("", "", novel_v)
```

Finally, we reprocess the content for lowercase text.

```
novel_lower_v <- char_tolower(novel_v)  
macbeth_word_v <- tokens(novel_lower_v, remove_punct = TRUE) %>% as.character()  
total_length <- length(macbeth_word_v)
```

## 1. Analyse and study the occurrence of words related with love or positive feelings in general

We count the number of times the word “love” is repeated.

```
length(macbeth_word_v[which(macbeth_word_v == "love")])
```

```
[1] 19
```

We do the same thing using kwic().

```
nrow(kwic(novel_lower_v, pattern = "love"))
```

```
[1] 19
```

We also count the number of times that words similar to “love” are repeated.

```
nrow(kwic(novel_lower_v, pattern = "love*"))
```

```
[1] 25
```

And we get the ratio of “love” to the total number of words.

```
total_love_hits <- nrow(kwic(novel_lower_v, pattern = "^love{0,1}$", valuetype = "regex"))
total_love_hits / ntoken(novel_lower_v, remove_punct = TRUE)
```

```
text1
0.00104453
```

## 2. Make frequency plots

We obtain the ten most frequent words.

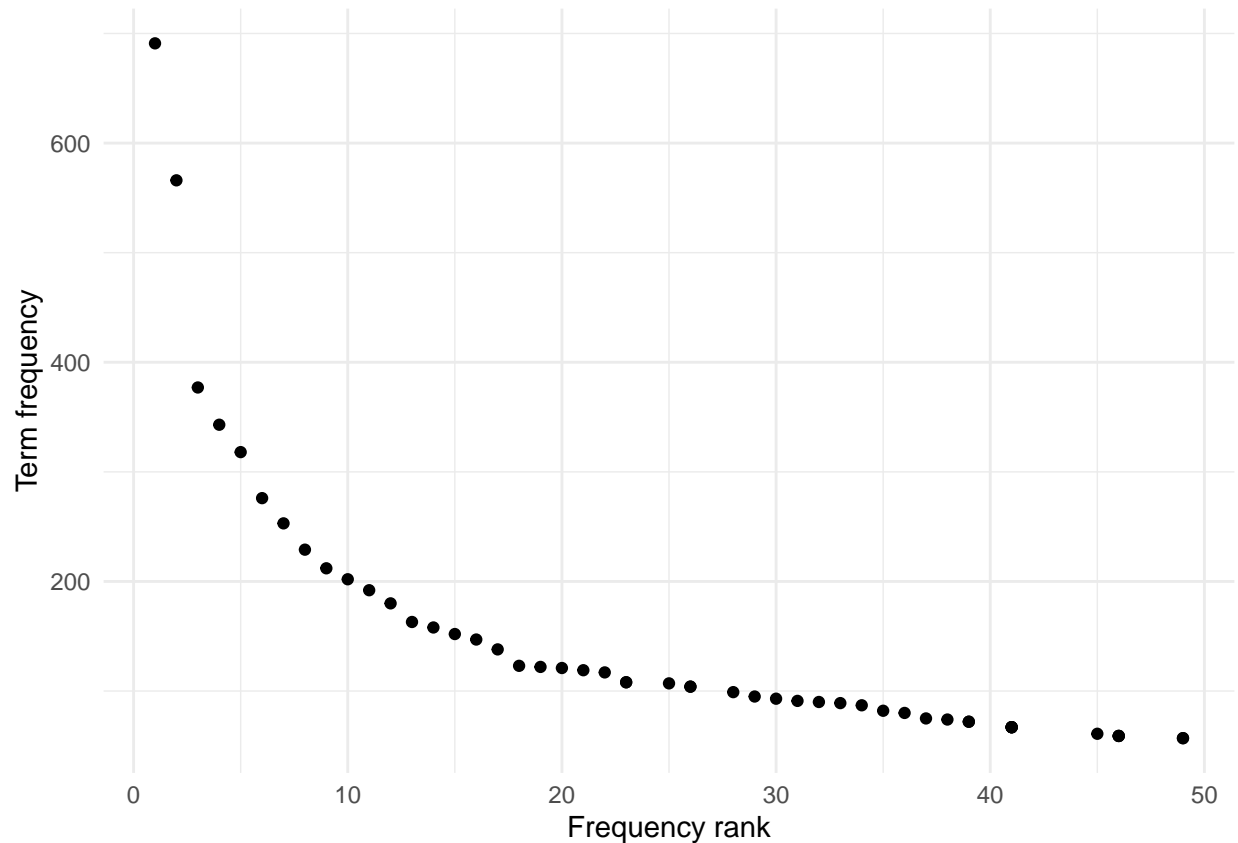
```
macbeth_dfm <- dfm(novel_lower_v, remove_punct = TRUE)
library("quantda.textstats")
textstat_frequency(macbeth_dfm, n = 10)
```

	feature	frequency	rank	docfreq	group
1	the	691	1	1	all
2	and	566	2	1	all
3	to	377	3	1	all
4	of	343	4	1	all
5	i	318	5	1	all
6	macbeth	276	6	1	all
7	a	253	7	1	all
8	that	229	8	1	all
9	in	212	9	1	all
10	you	202	10	1	all

And we plot frequency of 50 most frequent terms.

```
library(ggplot2)
theme_set(theme_minimal())

textstat_frequency(macbeth_dfm, n = 50) %>%
  ggplot(aes(x = rank, y = frequency)) +
  geom_point() +
  labs(x = "Frequency rank", y = "Term frequency")
```

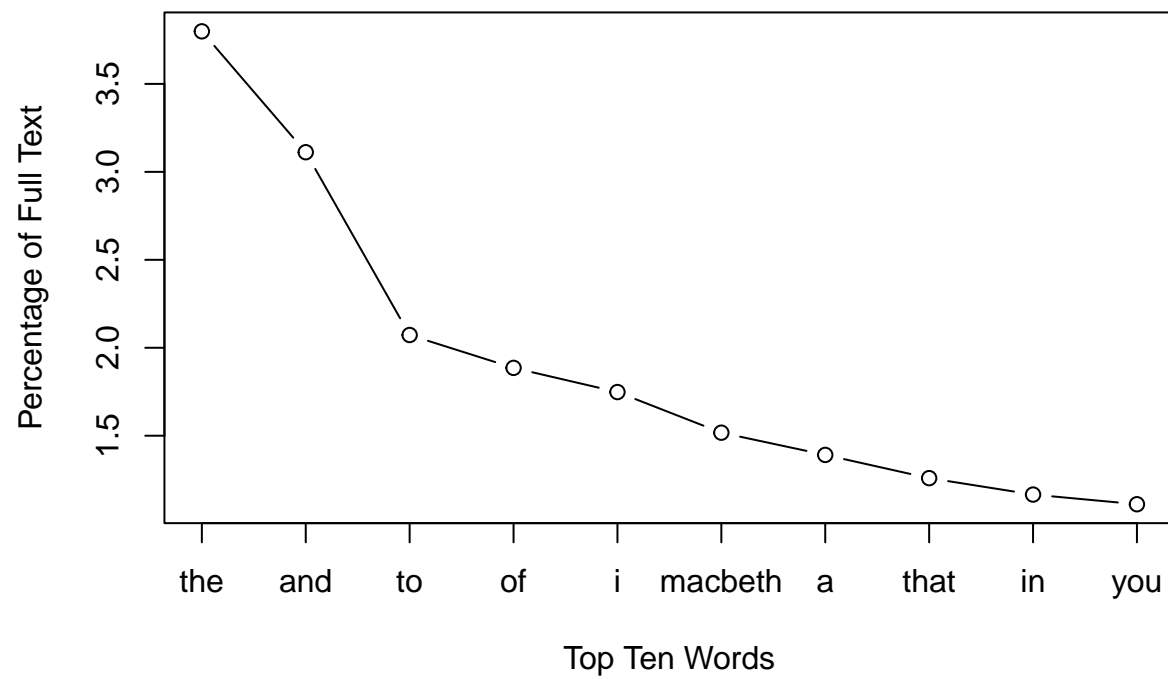


```
sorted_macbeth_freqs_t <- topfeatures(macbeth_dfm, n = nfeat(macbeth_dfm))
```

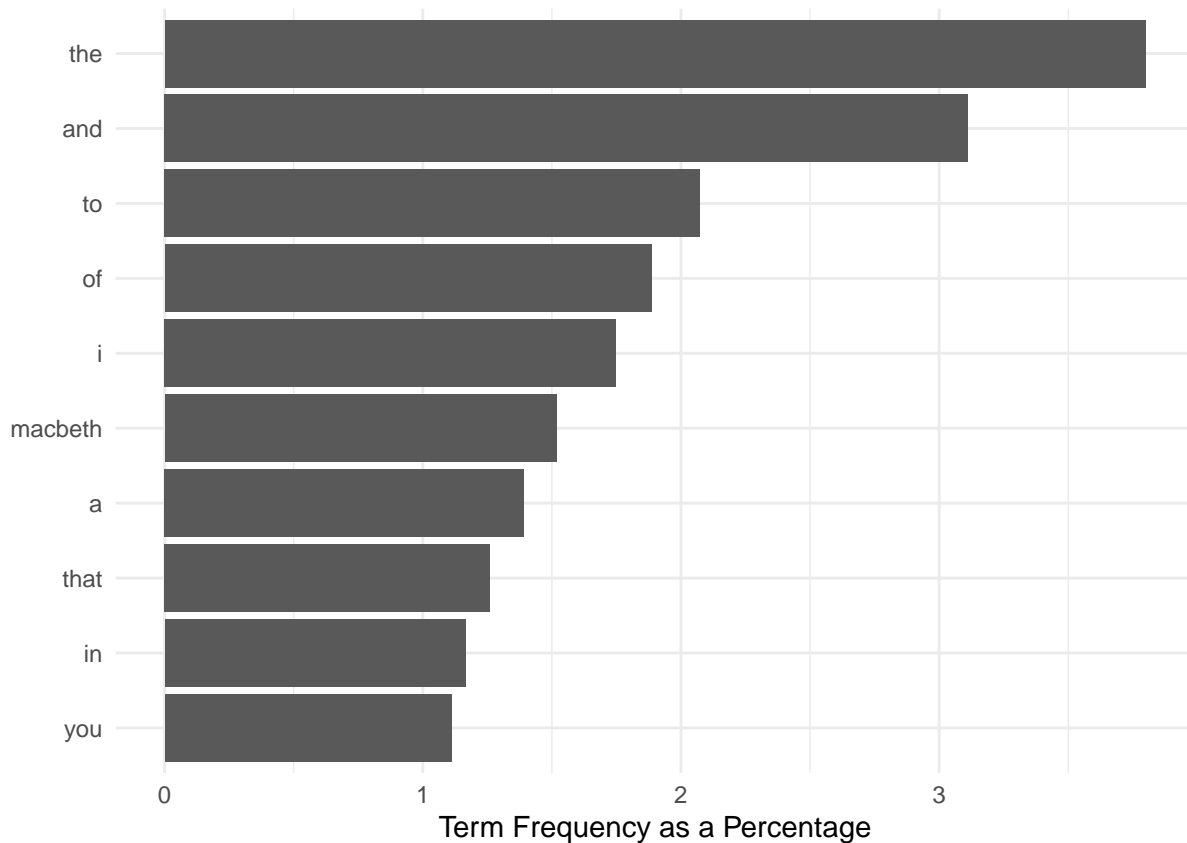
We also plot the percentage of full text and the term frequency as a percentage for the ten most frequent words.

```
sorted_macbeth_rel_freqs_t <- sorted_macbeth_freqs_t / sum(sorted_macbeth_freqs_t) * 100
macbeth_dfm_pct <- dfm_weight(macbeth_dfm, scheme = "prop") * 100

plot(sorted_macbeth_rel_freqs_t[1:10], type = "b",
      xlab = "Top Ten Words", ylab = "Percentage of Full Text", xaxt = "n")
axis(1, 1:10, labels = names(sorted_macbeth_rel_freqs_t[1:10]))
```



```
textstat_frequency(macbeth_dfm_pct, n = 10) %>%  
  ggplot(aes(x = reorder(feature, -rank), y = frequency)) +  
  geom_bar(stat = "identity") + coord_flip() +  
  labs(x = "", y = "Term Frequency as a Percentage")
```



### 3. Compare word frequency data of words like “he”, “she”, “him”, “her” and show also relative frequencies

We calculate frequencies of “he”, “she”, “him” and “her”.

```
sorted_macbeth_freqs_t[c("he", "she", "him", "her")]
```

```
he she him her
117 19 91 43
```

We do the same with another method, indexing the dfm.

```
macbeth_dfm[, c("he", "she", "him", "her")]
```

Document-feature matrix of: 1 document, 4 features (0.00% sparse) and 0 docvars.

```
features
docs    he she him her
text1 117 19 91 43
```

We also estimate relative frequencies.

```
sorted_macbeth_rel_freqs_t["he"]
```

```
      he  
0.6432106
```

```
sorted_macbeth_rel_freqs_t["she"]
```

```
      she  
0.104453
```

```
sorted_macbeth_rel_freqs_t["him"]
```

```
      him  
0.5002749
```

```
sorted_macbeth_rel_freqs_t["her"]
```

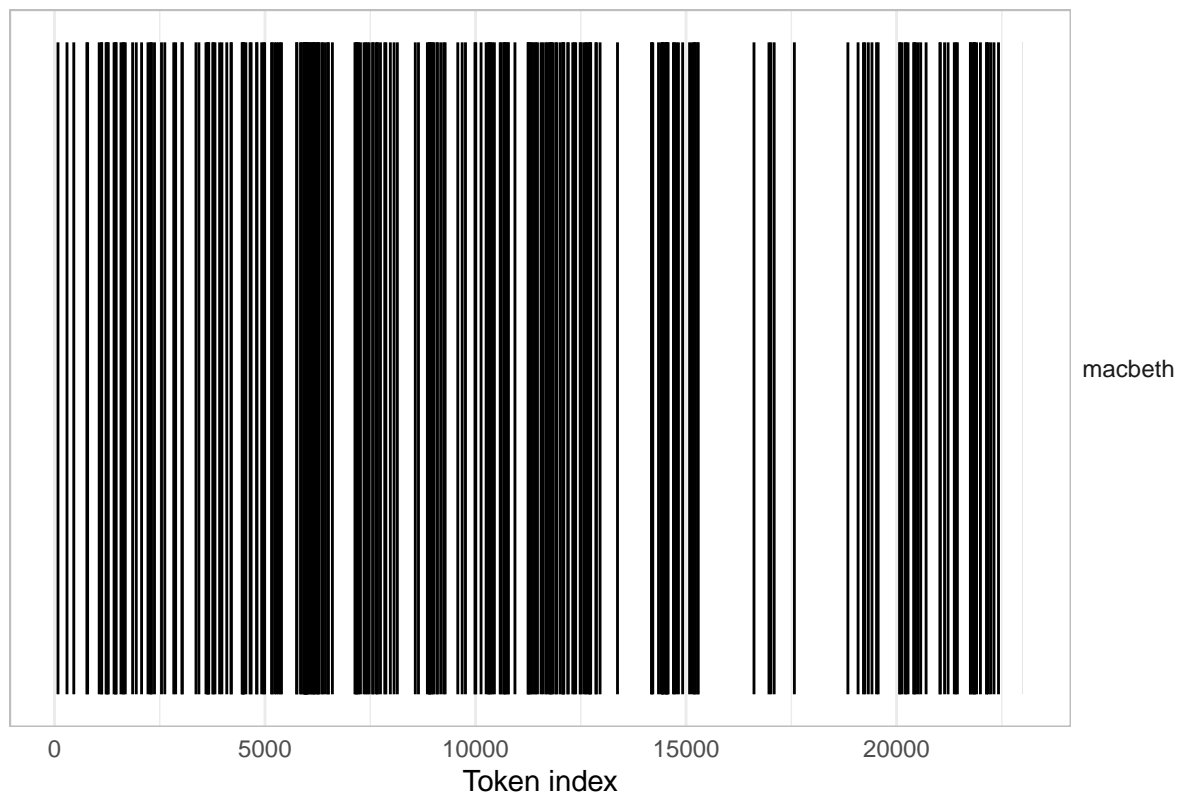
```
      her  
0.2363936
```

#### 4. Make a token distribution analysis

Dispersion plots using words “Macbeth” and “Macduff”.

```
library("quanteda.textplots")  
textplot_xray(kwic(novel_v, pattern = "macbeth")) +  
  ggtitle("Lexical dispersion")
```

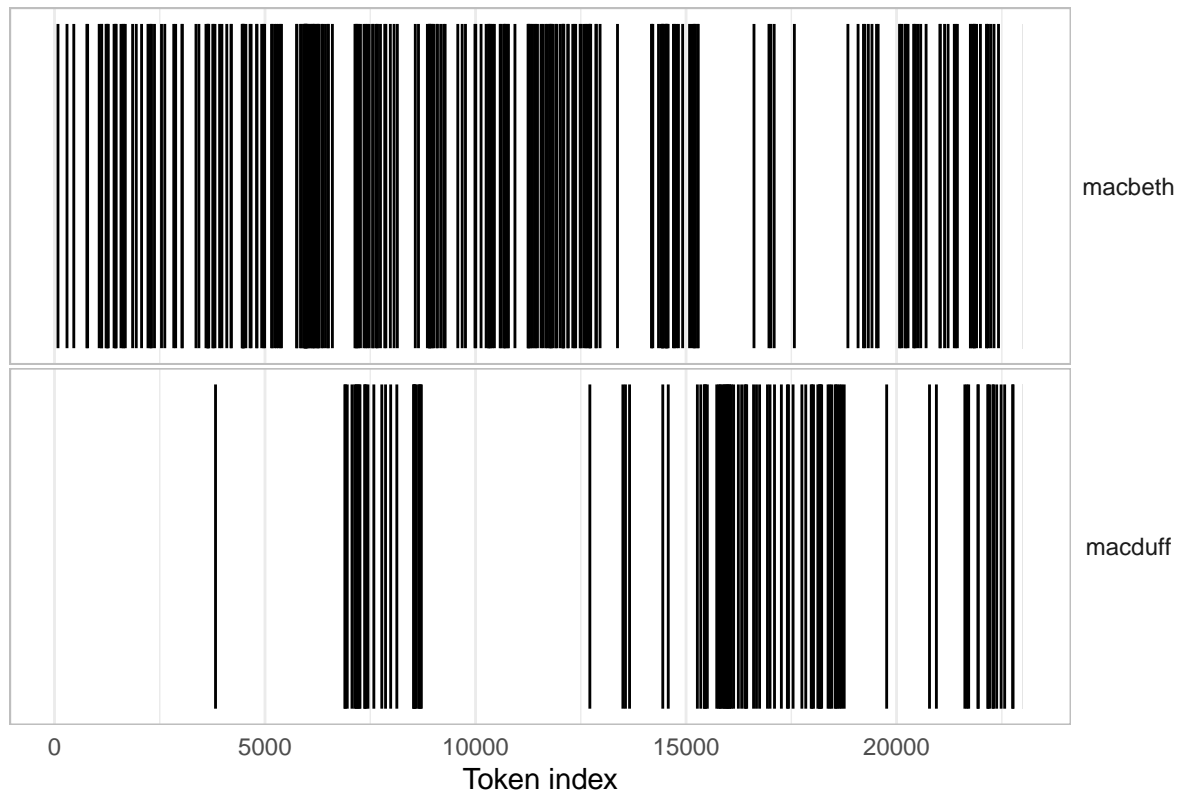
## Lexical dispersion



```
textplot_xray(  
  kwic(novel_v, pattern = "macbeth"),  
  kwic(novel_v, pattern = "macduff")) +  
  ggtitle("Lexical dispersion")
```



## Lexical dispersion



## 5. Identify chapter breaks

We identify the chapter break locations.

```
chap_positions_v <- kwic(novel_v, phrase(c("SCENE")), valuetype = "regex")$from
chap_positions_v
```

```
[1]      1   128   796  2391  3015  3797  4162  5036  5745  6635  8301  8774
[13] 10295 10898 11231 12904 13253 13773 15422 16408 18882 19721 20061 20756
[25] 21016 21598 21738 22125
```

And we save our chapter breaks in the variable “chapters\_corp”.

```
chapters_corp <-
  corpus(novel_v) %>%
  corpus_segment(pattern = "SCENE\\s*.*\\n", valuetype = "regex")
summary(chapters_corp, 10)
```

Corpus consisting of 28 documents, showing 10 documents:

	Text	Types	Tokens	Sentences
text1.1	67	120	25	
text1.2	361	660	52	
text1.3	591	1589	145	

text1.4	316	613	45
text1.5	373	771	54
text1.6	203	355	26
text1.7	416	862	53
text1.8	339	699	52
text1.9	367	884	103
text1.10	618	1660	163

```

pattern
  SCENE I. An open Place.\n
  SCENE II. A Camp near Forres.\n
  SCENE III. A heath.\n
  SCENE IV. Forres. A Room in the Palace.\n
  SCENE V. Inverness. A Room in Macbethâs Castle.\n
  SCENE VI. The same. Before the Castle.\n
  SCENE VII. The same. A Lobby in the Castle.\n
  SCENE I. Inverness. Court within the Castle.\n
  SCENE II. The same.\n
  SCENE III. The same.\n

```

```

docvars(chapters_corp, "pattern") <- stringi::stri_trim_right(docvars(chapters_corp, "pattern"))
docnames(chapters_corp) <- docvars(chapters_corp, "pattern")

```

## 7. Show some measures of lexical variety

We want to calculate the mean word frequency, so first, we obtain the length of the book in chapters.

```
ndoc(chapters_corp)
```

```
[1] 28
```

Then, we obtain chapter names.

```
docnames(chapters_corp) %>% head()
```

```

[1] "SCENE I. An open Place."
[2] "SCENE II. A Camp near Forres."
[3] "SCENE III. A heath."
[4] "SCENE IV. Forres. A Room in the Palace."
[5] "SCENE V. Inverness. A Room in Macbethâs Castle."
[6] "SCENE VI. The same. Before the Castle."

```

And calculate mean word frequency for the first few chapters.

```
ntoken(chapters_corp) %>% head()
```

```

  SCENE I. An open Place.
                        120
  SCENE II. A Camp near Forres.
                        660
  SCENE III. A heath.

```

	1589
SCENE IV. Forres. A Room in the Palace.	
	613
SCENE V. Inverness. A Room in Macbeth's Castle.	
	771
SCENE VI. The same. Before the Castle.	
	355

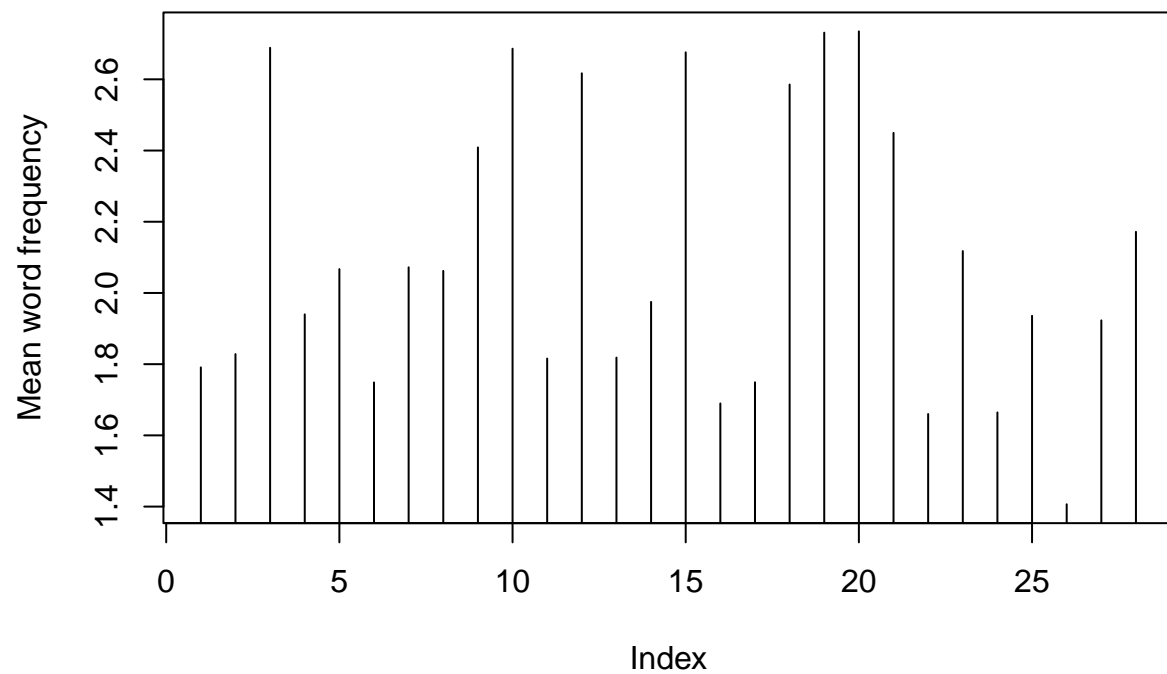
We do the same with the average.

```
(ntoken(chapters_corp) / ntype(chapters_corp)) %>% head()
```

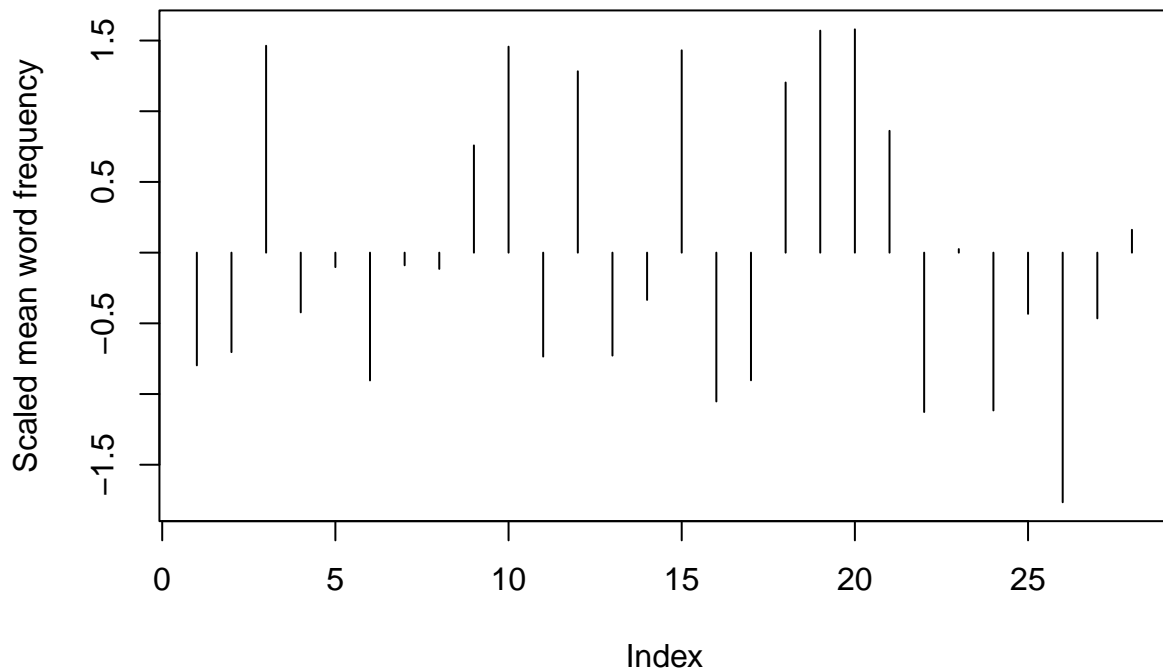
SCENE I. An open Place.	
	1.791045
SCENE II. A Camp near Forres.	
	1.828255
SCENE III. A heath.	
	2.688663
SCENE IV. Forres. A Room in the Palace.	
	1.939873
SCENE V. Inverness. A Room in Macbeth's Castle.	
	2.067024
SCENE VI. The same. Before the Castle.	
	1.748768

Then, we extract word usage means.

```
(ntoken(chapters_corp) / ntype(chapters_corp)) %>%  
plot(type = "h", ylab = "Mean word frequency")
```



```
(ntoken(chapters_corp) / ntype(chapters_corp)) %>%  
  scale() %>%  
  plot(type = "h", ylab = "Scaled mean word frequency")
```



We also rank the values.

```
mean_word_use_m <- (ntoken(chapters_corp) / ntype(chapters_corp))
sort(mean_word_use_m, decreasing = TRUE) %>% head()
```

```
SCENE III. England. Before the Kingâs Palace.
2.734739
SCENE II. Fife. A Room in Macduffâs Castle.
2.731092
SCENE III. A heath.
2.688663
SCENE III. The same.
2.686084
SCENE IV. The same. A Room of state in the Palace.
2.675806
SCENE I. Forres. A Room in the Palace.
2.616984
```

And finally, we calculate the TTR.

```
dfm(chapters_corp) %>%
  textstat_lexdiv(measure = "TTR") %>%
  head(n = 10)
```

document

TTR

```

1          SCENE I. An open Place. 0.6321839
2          SCENE II. A Camp near Forres. 0.6057143
3          SCENE III. A heath. 0.4022436
4          SCENE IV. Forres. A Room in the Palace. 0.5472837
5 SCENE V. Inverness. A Room in Macbethâs Castle. 0.5078616
6          SCENE VI. The same. Before the Castle. 0.6289753
7          SCENE VII. The same. A Lobby in the Castle. 0.4945205
8          SCENE I. Inverness. Court within the Castle. 0.5222816
9          SCENE II. The same. 0.4580925
10         SCENE III. The same. 0.4247439

```

## 8. Calculate the Hapax Richness

First, we create a dfm.

```
chap_dfm <- dfm(chapters_corp)
```

We calculate hapaxes per document.

```
rowSums(chap_dfm == 1) %>% head()
```

```

          SCENE I. An open Place.
                                45
          SCENE II. A Camp near Forres.
                                249
          SCENE III. A heath.
                                329
          SCENE IV. Forres. A Room in the Palace.
                                196
SCENE V. Inverness. A Room in Macbethâs Castle.
                                235
          SCENE VI. The same. Before the Castle.
                                141

```

And we calculate them again as a proportion.

```
hapax_proportion <- rowSums(chap_dfm == 1) / ntoken(chap_dfm)
head(hapax_proportion)
```

```

          SCENE I. An open Place.
                                0.3750000
          SCENE II. A Camp near Forres.
                                0.3772727
          SCENE III. A heath.
                                0.2070485
          SCENE IV. Forres. A Room in the Palace.
                                0.3197390
SCENE V. Inverness. A Room in Macbethâs Castle.
                                0.3047990
          SCENE VI. The same. Before the Castle.
                                0.3971831

```

```
barplot(hapax_proportion, beside = TRUE, col = "grey", names.arg = seq_len(ndoc(chap_dfm)))
```

