Text Analysis with R for Students of Literature

0. Preprocessing

Loading the first text file

```
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)
```

```
Warning: package 'readtext' was built under R version 4.1.2
```

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

library(stringi)
stri_sub(data_macbeth, 1, 65)
```

```
[1] "ï»¿The Project Gutenberg eBook of Macbeth, by William Shakespeare"
```

Separate content from metadata

Extract the header information

```
(start_v <- stri_locate_first_fixed(data_macbeth, "SCENE I. An open Place.")[1])
```

```
[1] 3032
```

```
(end_v <- stri_locate_last_fixed(data_macbeth, "[_Flourish. Exeunt._]")[1])
```

```
[1] 107169
```

Verify that "[*Flourish. Exeunt.*]" is the end of the novel

```
kwic(tokens(data_macbeth), "[_Flourish. Exeunt._]")
```

```
Keyword-in-context with 0 matches.
```

```
stri_count_fixed(data_macbeth, "\n")
```

```
[1] 4528
```

```
stri_sub(data_macbeth, from = start_v, to = end_v) %>%
  stri_count_fixed("\n")
```

```
[1] 4053
```

```
novel_v <- stri_sub(data_macbeth, start_v, end_v)
novel_v = gsub("€", "", novel_v)
novel_v = gsub("", "", novel_v)
length(novel_v)
```

```
[1] 1
```

```
stri_sub(novel_v, 1, 70) %>% cat()
```

```
SCENE I. An open Place.

 Thunder and Lightning. Enter three Witches.
```

Reprocessing the content

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] 18190
```

```
macbeth_word_v[1:11]
```

```
 [1] "scene"     "i"         "an"        "open"      "place"     "thunder"  
 [7] "and"       "lightning" "enter"     "three"     "witches"  
```

```
macbeth_word_v[9999]
```

```
[1] "once"
```

```
macbeth_word_v[c(6,7,8)]
```

```
[1] "thunder"   "and"       "lightning"
```

Check positions of “love”

```
which(macbeth_word_v == "love") %>% head()
```

```
[1] 2114 2902 3132 3145 3242 3302
```

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

Beginning the analysis

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

```
[1] 19
```

Same thing using kwic()

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

```
Warning: ’kwic.character()’ is deprecated. Use ’tokens()’ first.
```

```
[1] 19
```

```
nrow(kwic(novel_lower_v, pattern = "love*")) # Includes words like "whalemen"
```

```
Warning: ’kwic.character()’ is deprecated. Use ’tokens()’ first.
```

```
[1] 25
```

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

```
Warning: ’kwic.character()’ is deprecated. Use ’tokens()’ first.
```

```
[1] 19
```

```
total_love_hits / ntoken(novel_lower_v, remove_punct = TRUE)
```

```
     text1
0.00104453
```

Total unique words

```
length(unique(macbeth_word_v))
```

```
[1] 3503
```

```
ntype(char_tolower(novel_v), remove_punct = TRUE)
```

```
text1
 3503
```

2. Make frequency plots.

Ten most frequent words

```
macbeth_dfm <- dfm(novel_lower_v, remove_punct = TRUE)
```

```
Warning: 'dfm.character()' is deprecated. Use 'tokens()' first.

Warning: '...' should not be used for tokens() arguments; use 'tokens()' first.
```

```
head(macbeth_dfm, nf = 10)
```

```
Warning: nf argument is not used.

Document-feature matrix of: 1 document, 3,503 features (0.00% sparse) and 0 docvars.
       features
docs    scene   i an open place thunder and lightning enter three
  text1    28 318 32    4    11       6 566         2    72    12
[ reached max_nfeat ... 3,493 more features ]
```

```
library("quanteda.textstats")
```

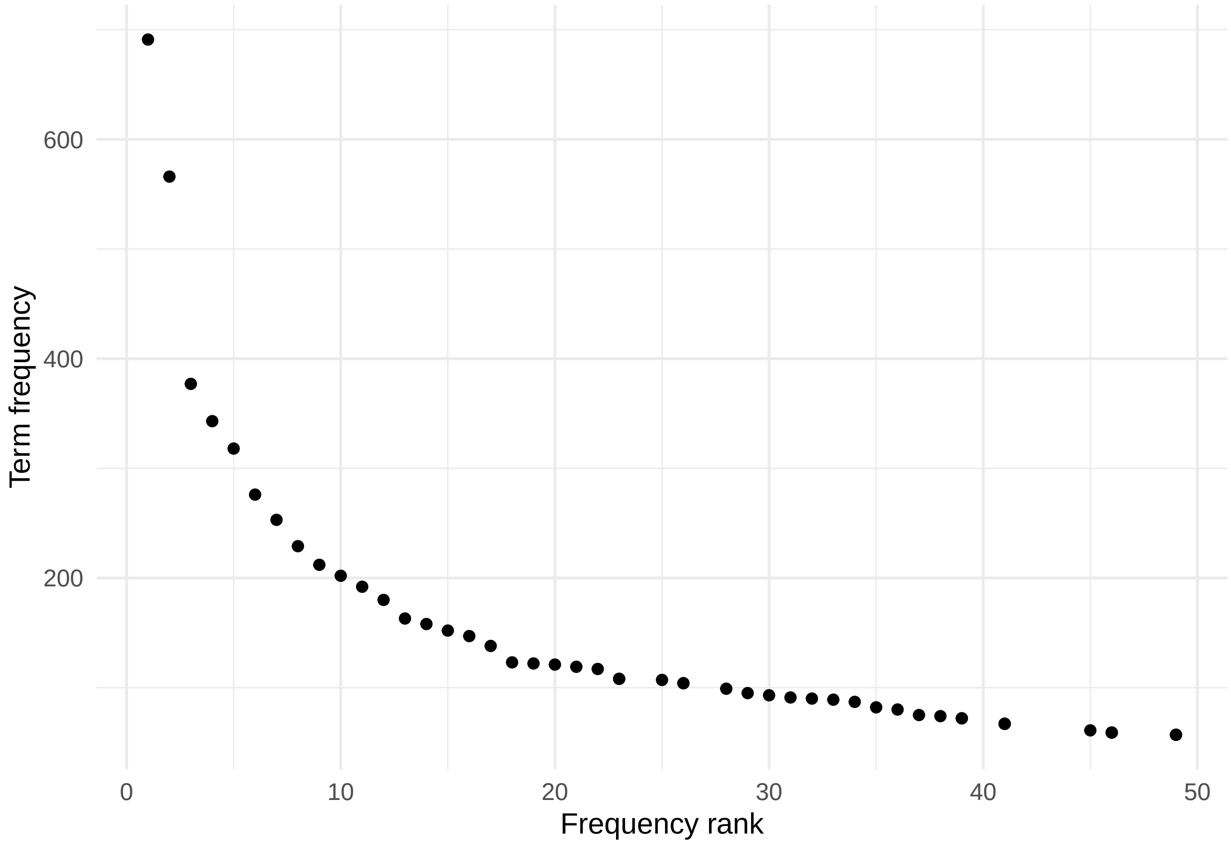
```
Warning: package 'quanteda.textstats' was built under R version 4.1.2
```

```
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
```

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))
```

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

Accessing Word Data

Frequencies of "he" and "she" - these are matrixes, not numerics

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

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

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
```

```
sorted_macbeth_freqs_t[1]
```

```
the
691
```

```
sorted_macbeth_freqs_t["the"]
```

```
the
691
```

Term frequency ratios

```
sorted_macbeth_freqs_t["him"] / sorted_macbeth_freqs_t["her"]
```

```
     him
2.116279
```

```
sorted_macbeth_freqs_t["he"] / sorted_macbeth_freqs_t["she"]
```

```
      he
6.157895
```

```
ntoken(macbeth_dfm)
```

```
text1
18190
```

```
sum(sorted_macbeth_freqs_t)
```

```
[1] 18190
```

2. Make frequency plots

Recycling

```
sorted_macbeth_rel_freqs_t <- sorted_macbeth_freqs_t / sum(sorted_macbeth_freqs_t) * 100
sorted_macbeth_rel_freqs_t["the"]
```

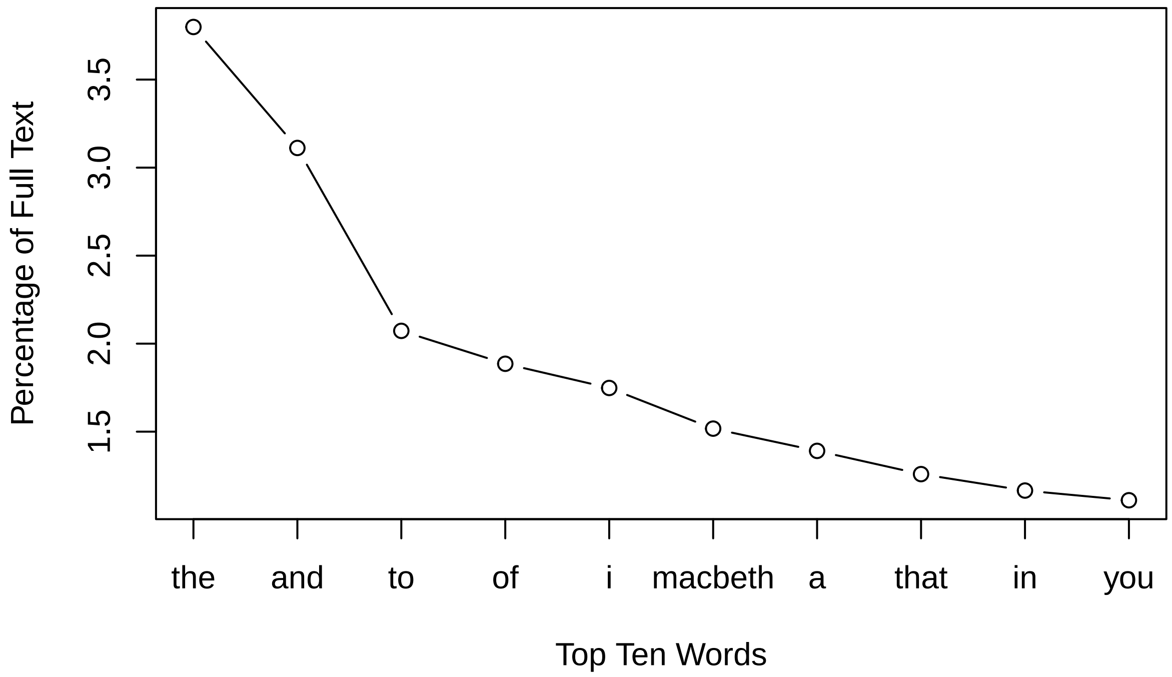
```
     the
3.798791
```

By weighting the dfm directly

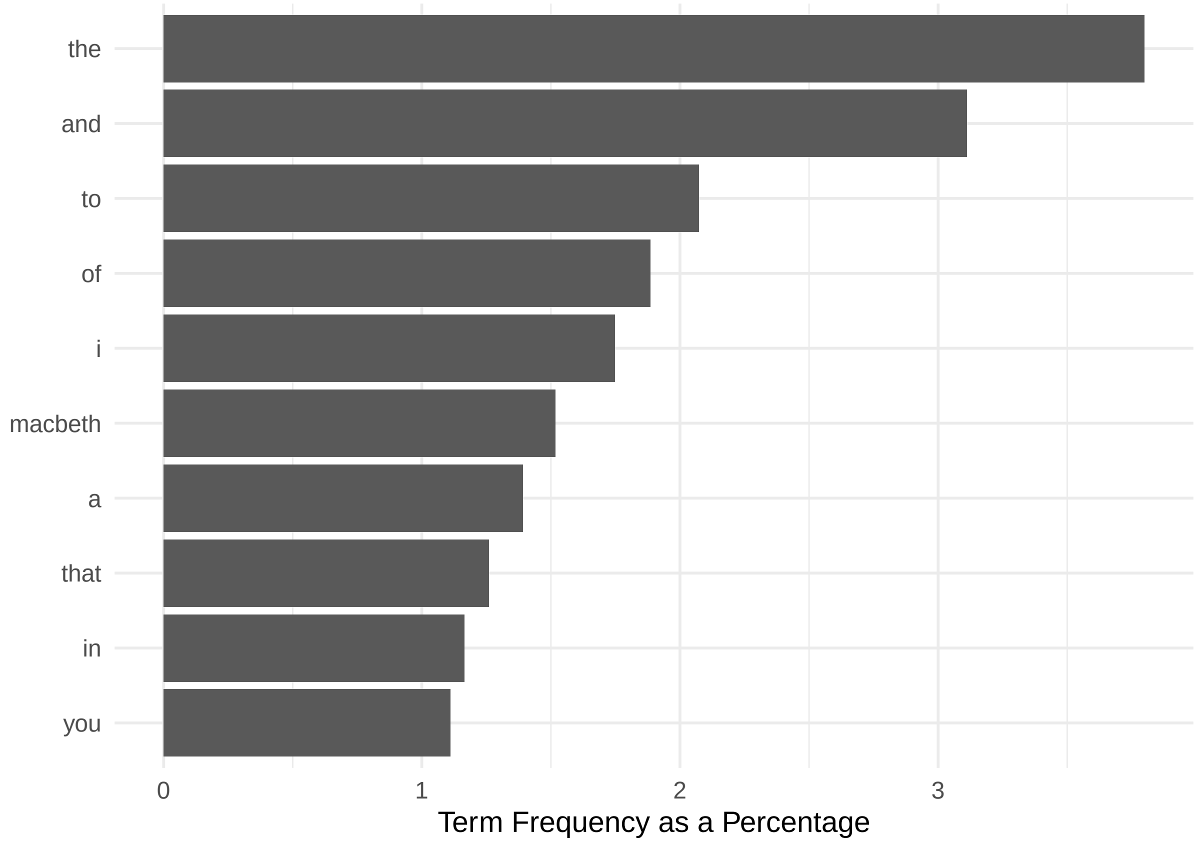
```
macbeth_dfm_pct <- dfm_weight(macbeth_dfm, scheme = "prop") * 100

dfm_select(macbeth_dfm_pct, pattern = "the")
```

```
Document-feature matrix of: 1 document, 1 feature (0.00% sparse) and 0 docvars.
       features
docs         the
  text1 3.798791
```

```
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")
```



4. Make a token distribution analysis.

Dispersion plots

Using words from tokenized corpus for dispersion

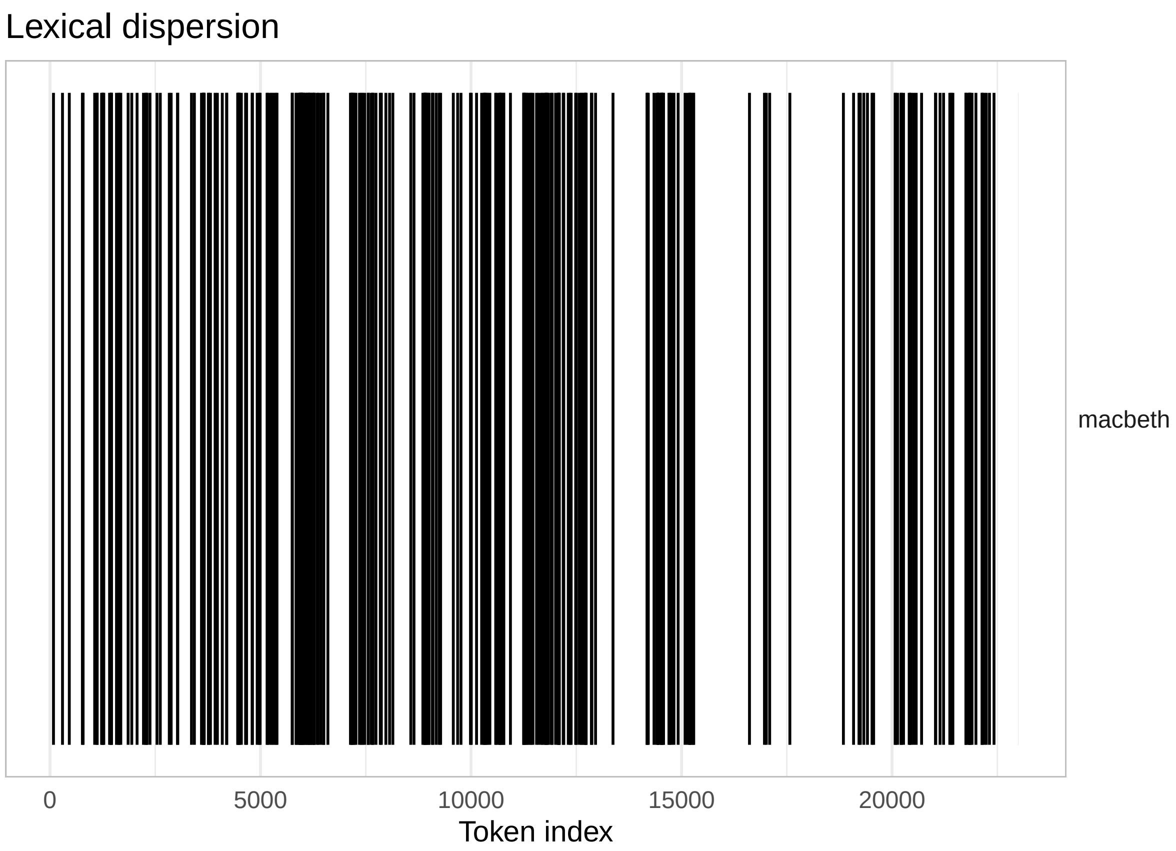
```
library("quanteda.textplots")
```

```
Warning: package 'quanteda.textplots' was built under R version 4.1.2
```

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

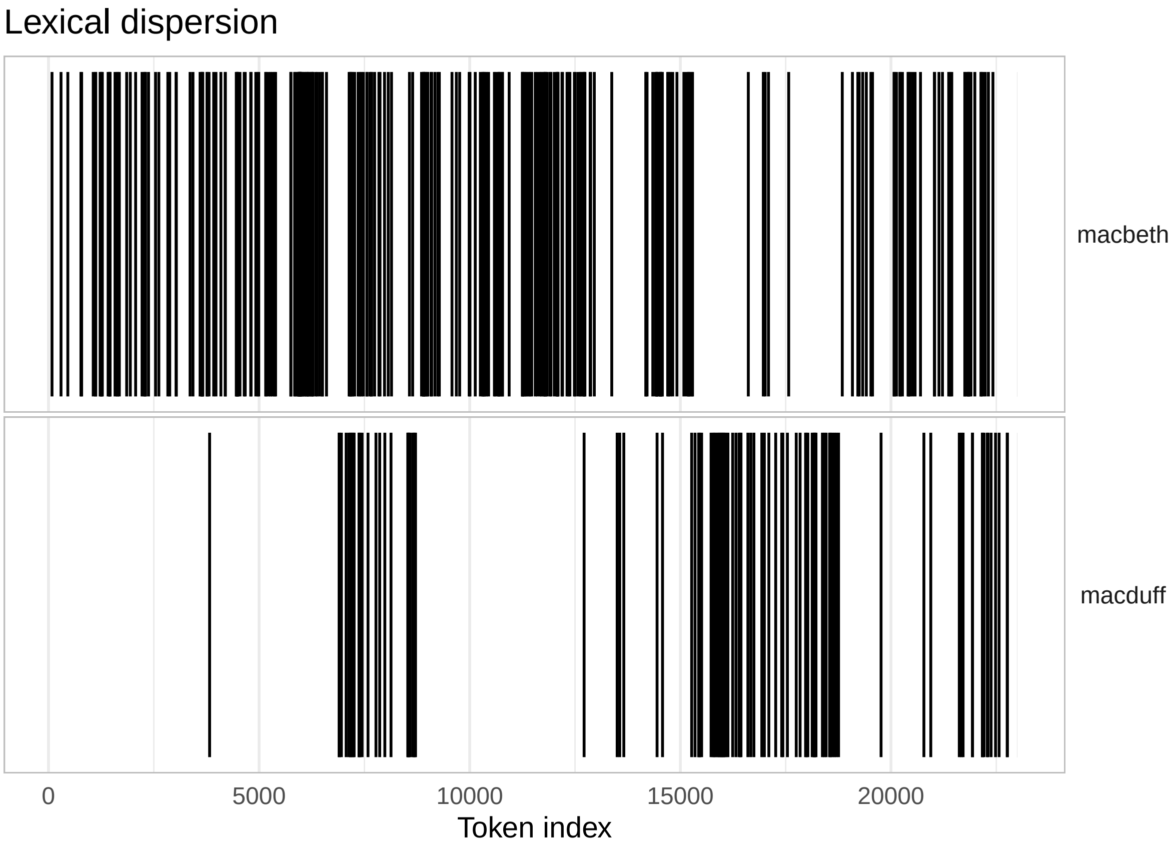
```
Warning: 'kwic.character()' is deprecated. Use 'tokens()' first.
```

```
Warning: Use of `x$ntokens` is discouraged. Use `ntokens` instead.
```



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

```
Warning: 'kwic.character()' is deprecated. Use 'tokens()' first.

Warning: 'kwic.character()' is deprecated. Use 'tokens()' first.

Warning: Use of `x$ntokens` is discouraged. Use `ntokens` instead.
```



5. Identify chapter breaks.

Searching with regular expression

Identify the chapter break locations

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

```
Warning: 'kwic.character()' is deprecated. Use 'tokens()' first.
```

```
head(chap_positions_v)
```

```
[1]    1  128  796 2391 3015 3797
```

```
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
```

Identifying chapter breaks

```r
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
```

```r
docvars(chapters_corp, "pattern") <- stringi::stri_trim_right(docvars(chapters_corp, "pattern"))
summary(chapters_corp, n = 3)
```

```
Corpus consisting of 28 documents, showing 3 documents:

    Text Types Tokens Sentences                       pattern
 text1.1    67    120        25       SCENE I. An open Place.
 text1.2   361    660        52 SCENE II. A Camp near Forres.
 text1.3   591   1589       145           SCENE III. A heath.
```

```r
docnames(chapters_corp) <- docvars(chapters_corp, "pattern")
```

Barplots of Macbeth and Macduff

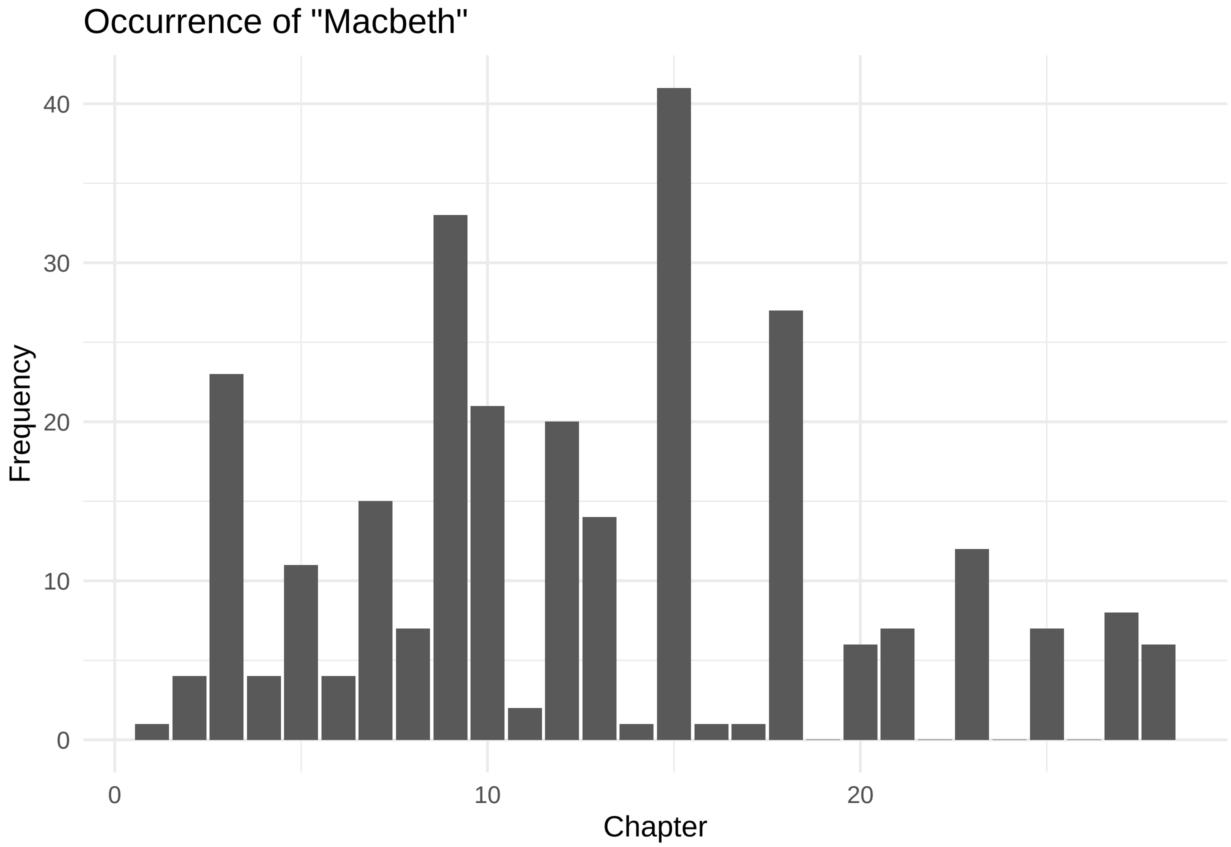
Create a dfm

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

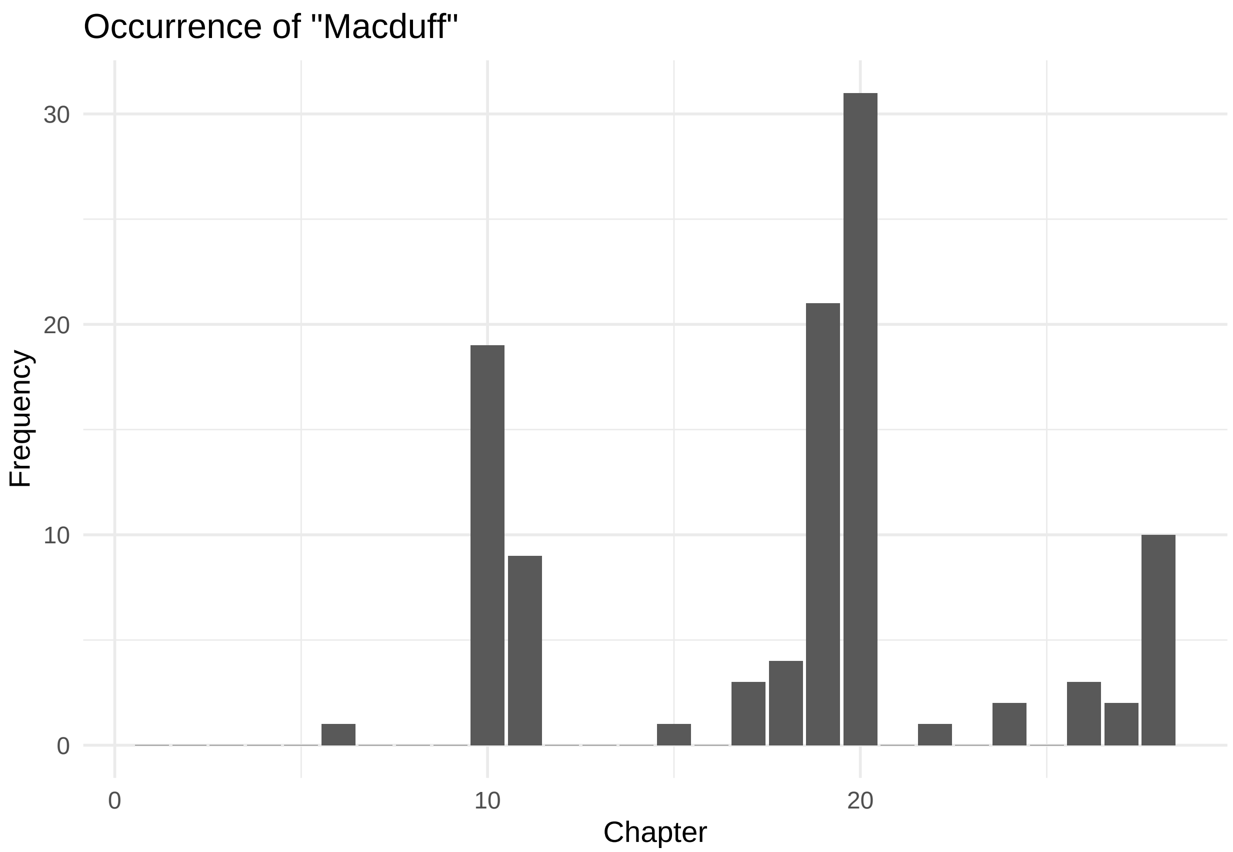
```
Warning: 'dfm.corpus()' is deprecated. Use 'tokens()' first.
```

Extract row with count for “whale”/“ahab” in each chapter and convert to data frame for plotting

```
macbeth_macduff_df <- chap_dfm %>%
  dfm_keep(pattern = c("macbeth", "macduff")) %>%
  convert(to = "data.frame")

macbeth_macduff_df$chapter <- 1:nrow(macbeth_macduff_df)

ggplot(data = macbeth_macduff_df, aes(x = chapter, y = macbeth)) +
  geom_bar(stat = "identity") +
  labs(x = "Chapter",
       y = "Frequency",
       title = 'Occurrence of "Macbeth"')
```



```
ggplot(data = macbeth_macduff_df, aes(x = chapter, y = macduff)) +
  geom_bar(stat = "identity") +
  labs(x = "Chapter",
       y = "Frequency",
       title = 'Occurrence of "Macduff"')
```

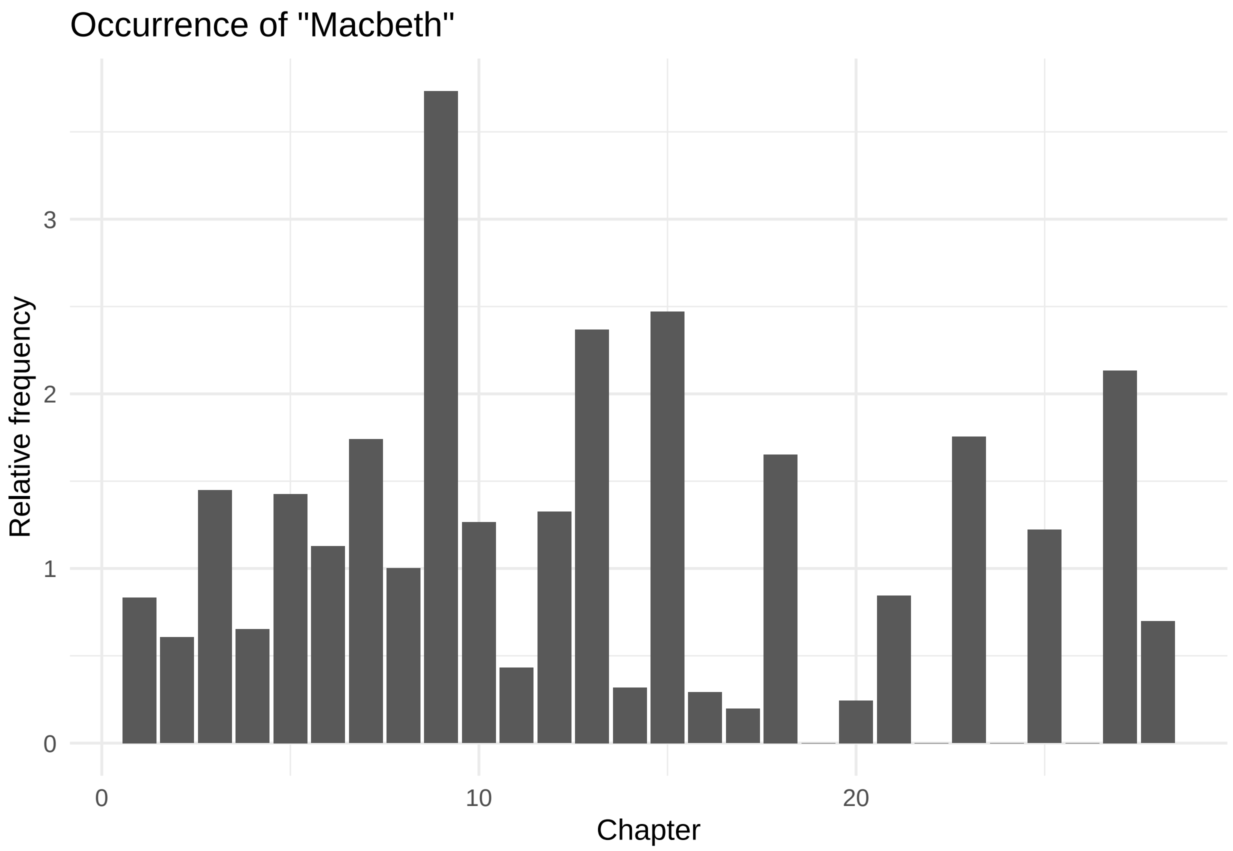


```r
rel_dfm <- dfm_weight(chap_dfm, scheme = "prop") * 100
head(rel_dfm)
```

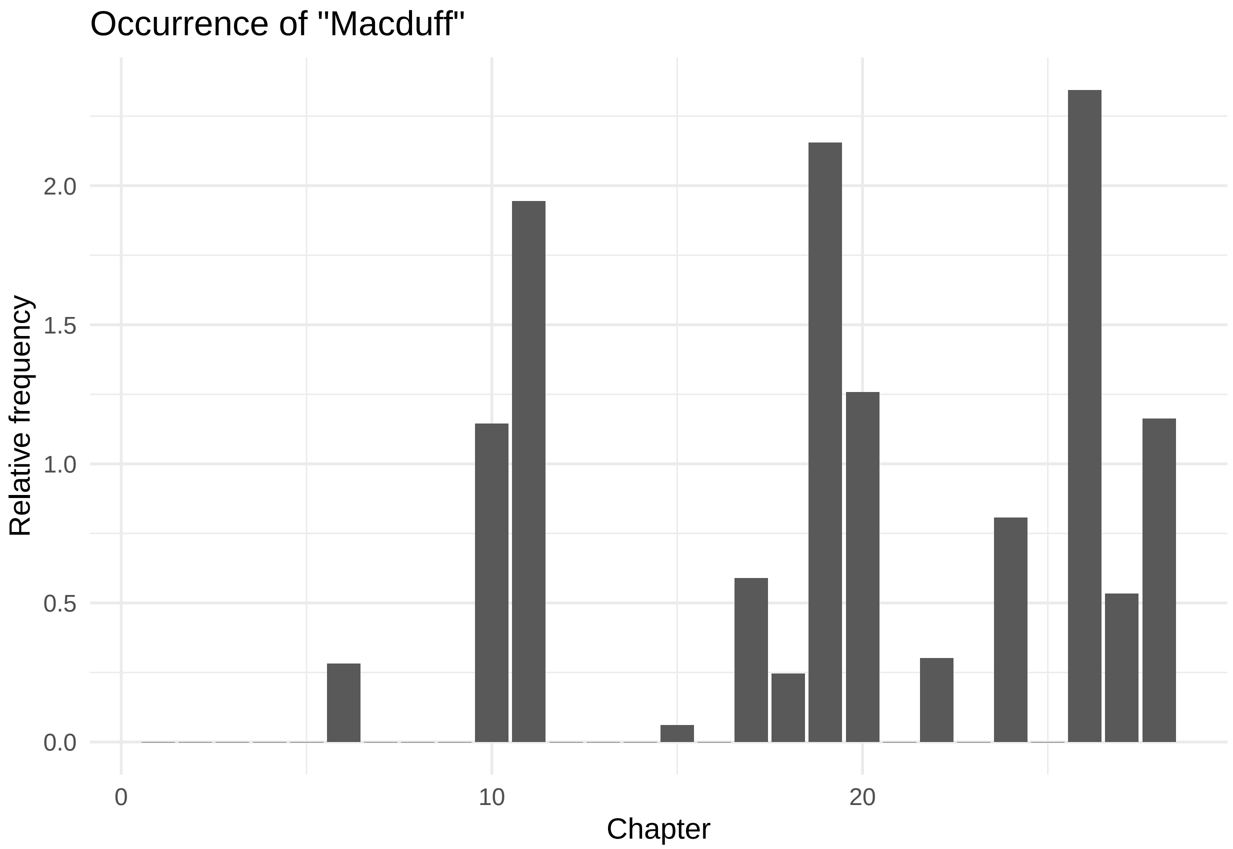
```
Document-feature matrix of: 6 documents, 3,500 features (91.86% sparse) and 1 docvar.
                                                features
docs                                                thunder      and lightning
  SCENE I. An open Place.                         1.66666667 3.333333  1.666667
  SCENE II. A Camp near Forres.                   0          1.818182  0
  SCENE III. A heath.                             0.06293266 2.769037  0
  SCENE IV. Forres. A Room in the Palace.         0          2.610114  0
  SCENE V. Inverness. A Room in Macbethâs Castle. 0          2.464332  0
  SCENE VI. The same. Before the Castle.          0          3.943662  0
                                                features
docs                                                       .     enter
  SCENE I. An open Place.                         16.666667 0.8333333
  SCENE II. A Camp near Forres.                    6.363636 0.3030303
  SCENE III. A heath.                              7.614852 0.1887980
  SCENE IV. Forres. A Room in the Palace.          6.035889 0.3262643
  SCENE V. Inverness. A Room in Macbethâs Castle.  5.577173 0.3891051
  SCENE VI. The same. Before the Castle.           6.760563 0.5633803
                                                features
docs                                                   three    witches
  SCENE I. An open Place.                         1.66666667 0.83333333
  SCENE II. A Camp near Forres.                   0          0
  SCENE III. A heath.                             0.06293266 0.06293266
  SCENE IV. Forres. A Room in the Palace.          0          0
  SCENE V. Inverness. A Room in Macbethâs Castle. 0          0
  SCENE VI. The same. Before the Castle.           0          0
                                                 features
docs                                                 first    witch      when
  SCENE I. An open Place.                          2.500000 7.500000 2.5000000
  SCENE II. A Camp near Forres.                    0        0        0
  SCENE III. A heath.                              0.566394 1.384519 0.1258653
  SCENE IV. Forres. A Room in the Palace.          0        0        0.1631321
  SCENE V. Inverness. A Room in Macbethâs Castle. 0        0        0.2594034
  SCENE VI. The same. Before the Castle.           0        0        0
[ reached max_nfeat ... 3,490 more features ]
```

Subset dfm and convert to data.frame object

```
rel_chap_freq <- rel_dfm %>%
  dfm_keep(pattern = c("macbeth", "macduff")) %>%
  convert(to = "data.frame")

rel_chap_freq$chapter <- 1:nrow(rel_chap_freq)
ggplot(data = rel_chap_freq, aes(x = chapter, y = macbeth)) +
  geom_bar(stat = "identity") +
  labs(x = "Chapter", y = "Relative frequency",
       title = 'Occurrence of "Macbeth"')
```



```r
ggplot(data = rel_chap_freq, aes(x = chapter, y = macduff)) +
  geom_bar(stat = "identity") +
  labs(x = "Chapter", y = "Relative frequency",
       title = 'Occurrence of "Macduff"')
```



6. Only if you have some knowledge about the novel: Make a correlation analysis between words related with love or positive feelings and some particular characters or people of the novel.

Correlation Analysis

```r
dfm_weight(chap_dfm, scheme = "prop") %>%
  textstat_simil(selection = c("macbeth", "macduff"), method = "correlation", margin = "features") %>%
  as.matrix() %>%
  head(2)
```

```
Warning: 'selection' is deprecated. Use 'y' instead.
```

```
            macbeth    macduff
thunder -0.06668006 -0.1554300
and     -0.34230943  0.2220466
```

Testing Correlation with Randomization+

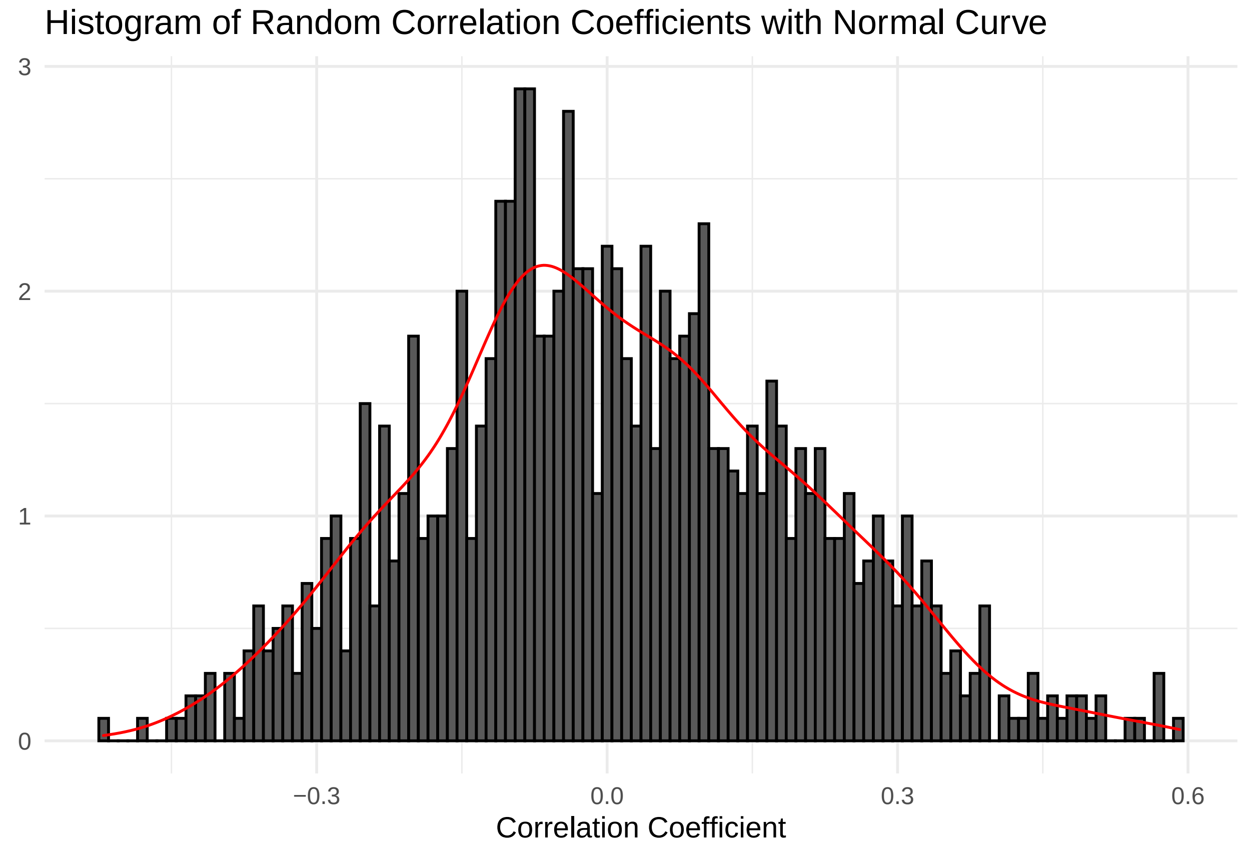
```
cor_data_df <- dfm_weight(chap_dfm, scheme = "prop") %>%
  dfm_keep(pattern = c("macbeth", "macduff")) %>%
  convert(to = "data.frame")
```

Sample 1000 replicates and create data frame

```
n <- 1000
samples <- data.frame(
  cor_sample = replicate(n, cor(sample(cor_data_df$macbeth), cor_data_df$macduff)),
  id_sample = 1:n
)
```

Plot distribution of resampled correlations

```
ggplot(data = samples, aes(x = cor_sample, y = ..density..)) +
  geom_histogram(colour = "black", binwidth = 0.01) +
  geom_density(colour = "red") +
  labs(x = "Correlation Coefficient", y = NULL,
       title = "Histogram of Random Correlation Coefficients with Normal Curve")
```



7. Show some measures of lexical variety.

Mean word frequency

Length of the book in chapters

```
ndoc(chapters_corp)
```

```
[1] 28
```

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."
```

For 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 Macbethâs Castle.
                                            771
         SCENE VI. The same. Before the Castle.
                                            355
```

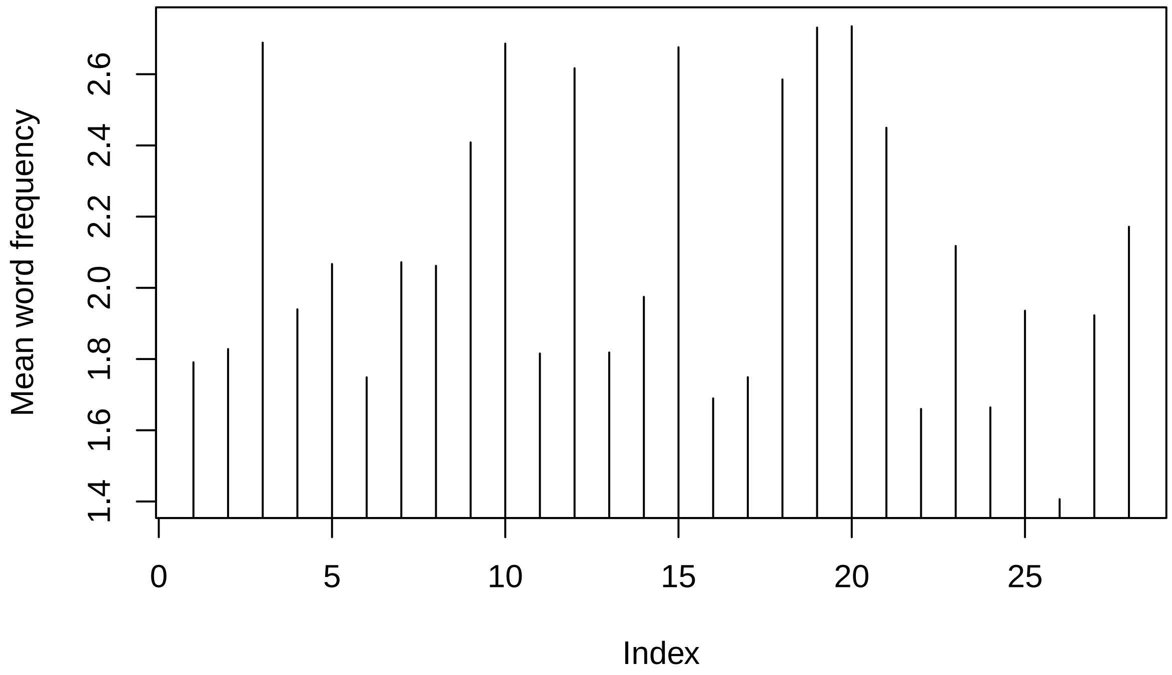
Average

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

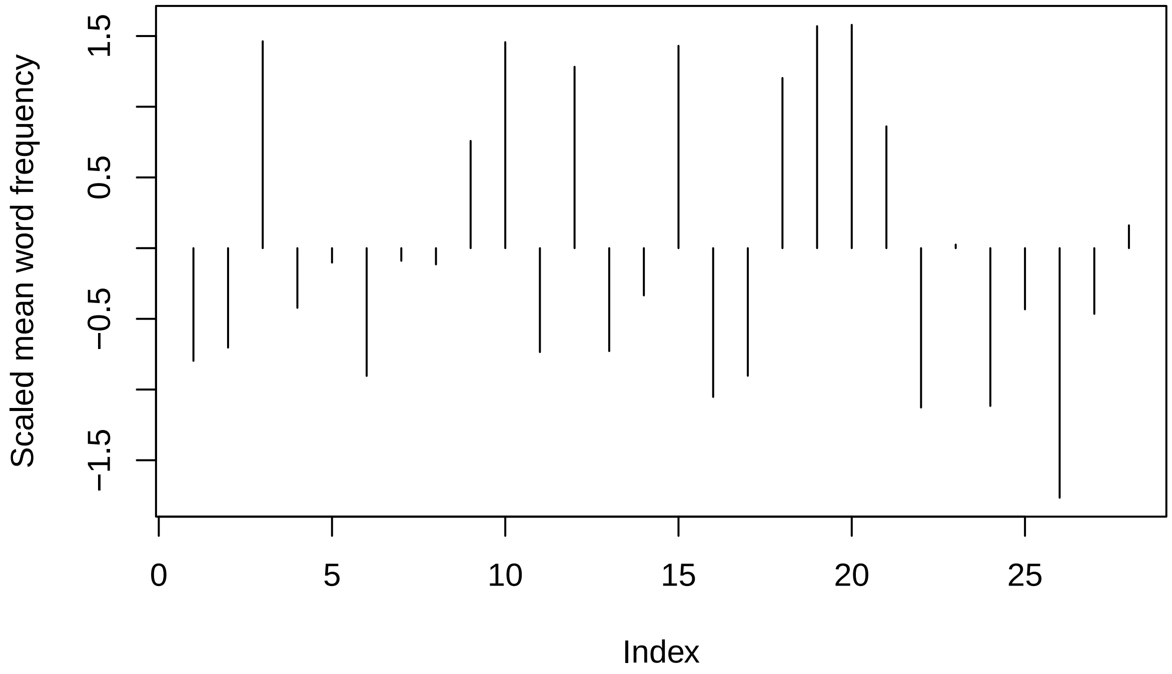
```
                        SCENE I. An open Place.
                                       1.791045
                  SCENE II. A Camp near Forres.
                                       1.828255
                            SCENE III. A heath.
                                       2.688663
        SCENE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

Extracting 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")
```



Ranking 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
```

Calculating the TTR

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

```
Warning: 'dfm.corpus()' is deprecated. Use 'tokens()' first.
```

```
                                                 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.

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
```

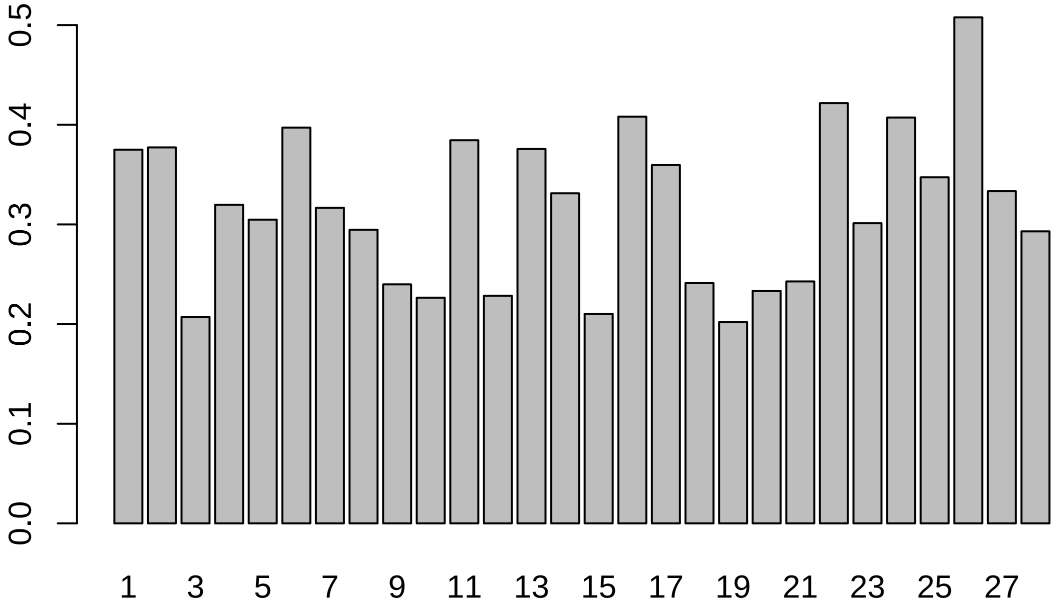
As a proportion

```
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
```

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


0.0
0.1
0.2
0.3
0.4
0.5
1
3
5
7
9
11
13
15
17
19
21
23
25
27