project_stat653

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
titles <- c("The Picture of Dorian Gray",
            "Alice's Adventures in Wonderland",
            "Dracula",
            "The Republic")
```

Retrieving the text of these four books using the gutenbergr package

```
library(gutenbergr)

books <- gutenberg_works(title %in% titles) %>%
  gutenberg_download(meta_fields = "title")
```

```
Determining mirror for Project Gutenberg from https://www.gutenberg.org/robot/harvest
```

```
Using mirror http://aleph.gutenberg.org
```

As a pre-processing step, we will break down these books into individual chapters and then use the unnest_tokens() function from the tidytext package to separate the text into individual words. We will also remove commonly occurring stop words from the text. In our analysis, we will treat each chapter of the book as a separate "document".

```
library(stringr)
library(tidytext)
library(tidyr)
library(topicmodels)



# divide into documents, each representing one chapter
by_chapter <- books %>%
  group_by(title) %>%
  mutate(chapter = cumsum(str_detect(
    text, regex("^chapter ", ignore_case = TRUE)
  ))) %>%
  ungroup() %>%
  filter(chapter > 0) %>%
  unite(document, title, chapter)

# split into words
by_chapter_word <- by_chapter %>%
  unnest_tokens(word, text)

# find document-word counts
word_counts <- by_chapter_word %>%
  anti_join(stop_words) %>%
  count(document, word, sort = TRUE)
```

```
Joining with `by = join_by(word)`
```

```
word_counts
```

```
# A tibble: 48,791 x 3
   document                              word       n
   <chr>                                 <chr>  <int>
 1 Alice's Adventures in Wonderland_7 alice     50
 2 Alice's Adventures in Wonderland_9 alice     47
 3 Alice's Adventures in Wonderland_6 alice     43
```

```
 4 The Picture of Dorian Gray_2       dorian    43
 5 The Picture of Dorian Gray_2       lord      40
 6 Alice's Adventures in Wonderland_8 alice     39
 7 Dracula_35                         lucy      37
 8 Dracula_39                         van       37
 9 The Picture of Dorian Gray_2       henry     37
10 The Picture of Dorian Gray_2       gray      36
# i 48,781 more rows
```

LDA on chapters To create a topic model for these four books, we can make use of the LDA() function. Since we have four books, we know that we are looking to create a model with four topics. The LDA() function uses a technique called Latent Dirichlet Allocation (LDA) to identify the underlying topics within a corpus of text. It works by assigning each word in the corpus to a topic and then iteratively refining these assignments until a stable set of topics is identified. By creating a four-topic model, we can identify the key themes and concepts that are present across all four books. This can help us to gain a better understanding of the similarities and differences between the books and provide insights into the underlying themes and ideas that they explore. Overall, the LDA() function is a powerful tool for text analysis and can be used to explore a wide range of textual datasets.

```
chapters_dtm <- word_counts %>%
  cast_dtm(document, word, n)

chapters_dtm
```

```
<<DocumentTermMatrix (documents: 86, terms: 12749)>>
Non-/sparse entries: 48791/1047623
Sparsity           : 96%
Maximal term length: 17
Weighting          : term frequency (tf)
```

```
chapters_lda <- LDA(chapters_dtm, k = 4, control = list(seed = 1234))
chapters_lda
```

```
A LDA_VEM topic model with 4 topics.
```

```
#> A LDA_VEM topic model with 4 topics.
```

Similar to our approach with the Associated Press data, we can analyze the per-topic-per-word probabilities of our topic model for these four books. By examining these probabilities, we can

gain insight into the words that are most strongly associated with each topic. This can help us to understand the key themes and concepts that are present within each topic and how they relate to the overall content of the books.

```
chapter_topics <- tidy(chapters_lda, matrix = "beta")
chapter_topics
```

```
# A tibble: 50,996 x 3
   topic term        beta
   <int> <chr>      <dbl>
 1     1 alice  2.94e-  2
 2     2 alice  1.03e-115
 3     3 alice  2.17e-  7
 4     4 alice  2.19e-145
 5     1 dorian 7.27e- 18
 6     2 dorian 1.33e-  3
 7     3 dorian 1.87e-  2
 8     4 dorian 3.98e-151
 9     1 lord   7.70e-  6
10     2 lord   1.74e-  3
# i 50,986 more rows
```

After examining the per-topic-per-word probabilities of our topic model, we can observe that the format has been transformed to a one-topic-per-term-per-row format. In this format, the model computes the probability of each term being generated from a particular topic, for all possible combinations of terms and topics. By analyzing these probabilities, we can identify which terms are most strongly associated with each topic and gain a deeper understanding of the themes and concepts that underlie each topic. This can help us to interpret and make sense of the output of our topic model, and gain insights into the patterns and relationships that exist within our corpus of text.

```
top_terms <- chapter_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

```
# A tibble: 20 x 3
   topic term       beta
```

```
   <int> <chr>     <dbl>
 1     1 alice   0.0294
 2     1 time    0.00809
 3     1 queen   0.00517
 4     1 don't   0.00502
 5     1 king    0.00472
 6     2 count   0.00623
 7     2 door    0.00585
 8     2 time    0.00545
 9     2 eyes    0.00395
10     2 life    0.00388
11     3 dorian  0.0187
12     3 don't   0.0118
13     3 lord    0.0114
14     3 henry   0.0101
15     3 life    0.00943
16     4 van     0.00859
17     4 time    0.00837
18     4 helsing 0.00801
19     4 night   0.00679
20     4 lucy    0.00587
```

To identify the top 5 terms associated with each topic, we can use the slice_max() function from the dplyr package. This function allows us to slice the data frame to return the rows with the highest values of a specified variable, in our case the per-topic-per-term probabilities. By using slice_max() with a grouping variable for each topic, we can extract the top 5 terms associated with each topic.

```
library(ggplot2)

top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>
```

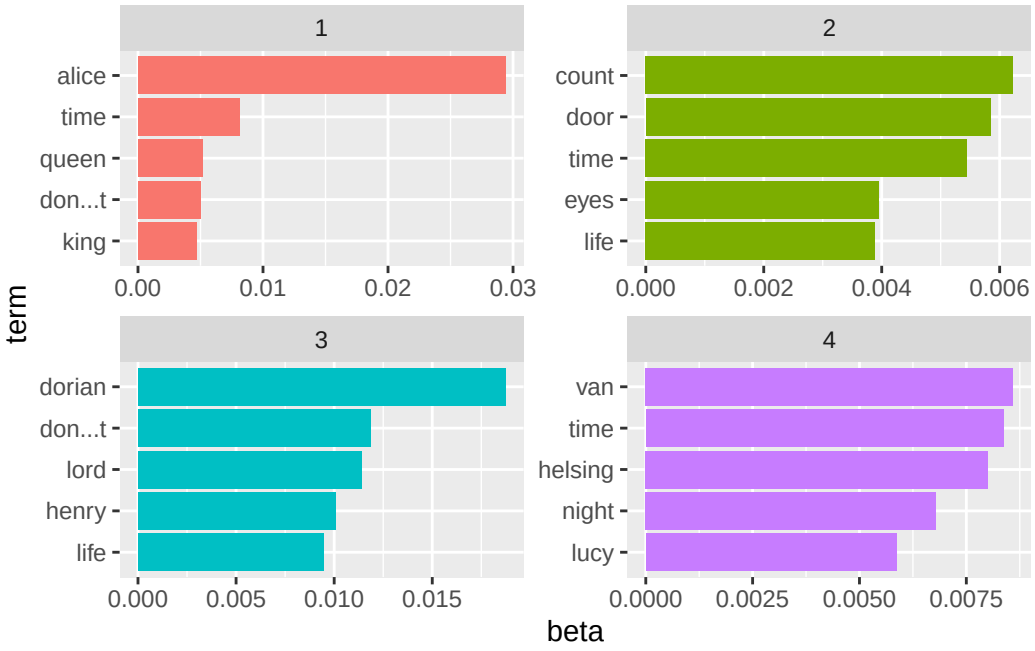
```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for <99>
```




```
chapters_gamma <- tidy(chapters_lda, matrix = "gamma")
chapters_gamma
```

```
# A tibble: 344 x 3
   document                            topic     gamma
   <chr>                               <int>     <dbl>
 1 Alice's Adventures in Wonderland_7      1 1.00
 2 Alice's Adventures in Wonderland_9      1 1.00
```

```
 3 Alice's Adventures in Wonderland_6     1 1.00
 4 The Picture of Dorian Gray_2           1 0.0000148
 5 Alice's Adventures in Wonderland_8     1 1.00
 6 Dracula_35                             1 0.0000132
 7 Dracula_39                             1 0.0000121
 8 Alice's Adventures in Wonderland_5     1 1.00
 9 Dracula_54                             1 0.0499
10 The Picture of Dorian Gray_3           1 0.0000168
# i 334 more rows
```

The function "tidy()" is utilized to extract or modify the matrix containing document-topic distribution, and the resultant "chapters_gamma" variable includes the relevant data for subsequent examination or display.

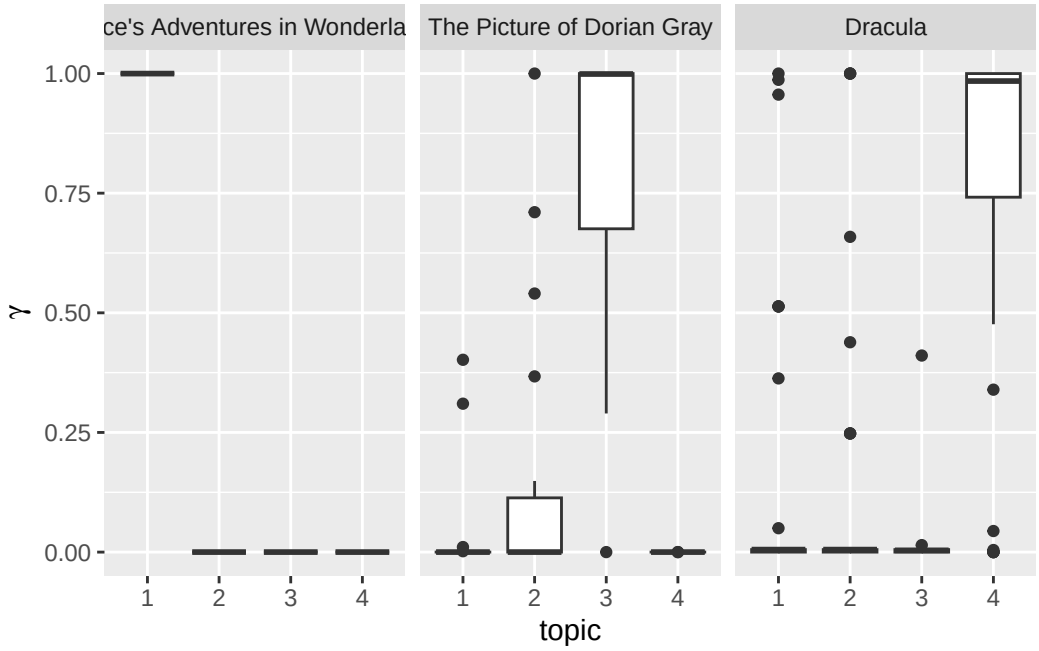
```
chapters_gamma <- chapters_gamma %>%
  separate(document, c("title", "chapter"), sep = "_", convert = TRUE)

chapters_gamma
```

```
# A tibble: 344 x 4
   title                             chapter topic     gamma
   <chr>                               <int> <int>     <dbl>
 1 Alice's Adventures in Wonderland        7     1 1.00
 2 Alice's Adventures in Wonderland        9     1 1.00
 3 Alice's Adventures in Wonderland        6     1 1.00
 4 The Picture of Dorian Gray              2     1 0.0000148
 5 Alice's Adventures in Wonderland        8     1 1.00
 6 Dracula                                35     1 0.0000132
 7 Dracula                                39     1 0.0000121
 8 Alice's Adventures in Wonderland        5     1 1.00
 9 Dracula                                54     1 0.0499
10 The Picture of Dorian Gray              3     1 0.0000168
# i 334 more rows
```

The provided code is capable of dividing a singular column within a data frame into two distinct columns, based on a designated separator. The updated data frame is then saved in the same variable, allowing for enhanced organization and analysis of combined data residing within a single column.

```
chapters_gamma %>%
  mutate(title = reorder(title, gamma * topic)) %>%
```

```
  ggplot(aes(factor(topic), gamma)) +
  geom_boxplot() +
  facet_wrap(~ title) +
  labs(x = "topic", y = expression(gamma))
```



The code is designed to generate a boxplot representation depicting the distribution of topics within each title of the provided data frame. This visualization enables the examination of the comparative predominance of various topics within each document.

```
chapter_classifications <- chapters_gamma %>%
  group_by(title, chapter) %>%
  slice_max(gamma) %>%
  ungroup()

chapter_classifications
```

```
# A tibble: 86 x 4
   title                              chapter topic gamma
   <chr>                                <int> <int> <dbl>
 1 Alice's Adventures in Wonderland         1     1  1.00
 2 Alice's Adventures in Wonderland         2     1  1.00
```

```
 3 Alice's Adventures in Wonderland       3     1  1.00
 4 Alice's Adventures in Wonderland       4     1  1.00
 5 Alice's Adventures in Wonderland       5     1  1.00
 6 Alice's Adventures in Wonderland       6     1  1.00
 7 Alice's Adventures in Wonderland       7     1  1.00
 8 Alice's Adventures in Wonderland       8     1  1.00
 9 Alice's Adventures in Wonderland       9     1  1.00
10 Alice's Adventures in Wonderland      10     1  1.00
# i 76 more rows
```

The given code extracts the primary topic classification for every chapter in all titles featured in the "chapters_gamma" data frame. This information aids in identifying the predominant subjects and themes present in each document.

```
book_topics <- chapter_classifications %>%
  count(title, topic) %>%
  group_by(title) %>%
  slice_max(n, n = 1) %>%
  ungroup() %>%
  transmute(consensus = title, topic)

chapter_classifications %>%
  inner_join(book_topics, by = "topic") %>%
  filter(title != consensus)
```

```
# A tibble: 6 x 5
  title   chapter topic gamma consensus
  <chr>     <int> <int> <dbl> <chr>
1 Dracula       6     1 0.513 Alice's Adventures in Wonderland
2 Dracula       7     1 0.987 Alice's Adventures in Wonderland
3 Dracula       8     1 0.513 Alice's Adventures in Wonderland
4 Dracula       9     1 0.513 Alice's Adventures in Wonderland
5 Dracula      28     1 1.00  Alice's Adventures in Wonderland
6 Dracula      34     1 0.956 Alice's Adventures in Wonderland
```

The provided code is utilized to detect the chapters within the data frame that possess a dissimilar topic classification compared to the general topic consensus for each book. This analysis can assist in identifying sections of potential discord or disparity within the thematic content of each book.

```r
assignments <- augment(chapters_lda, data = chapters_dtm)
assignments
```

```
# A tibble: 48,791 x 4
   document                              term  count .topic
   <chr>                                 <chr> <dbl>  <dbl>
 1 Alice's Adventures in Wonderland_7    alice    50      1
 2 Alice's Adventures in Wonderland_9    alice    47      1
 3 Alice's Adventures in Wonderland_6    alice    43      1
 4 Alice's Adventures in Wonderland_8    alice    39      1
 5 Alice's Adventures in Wonderland_5    alice    35      1
 6 Alice's Adventures in Wonderland_10   alice    30      1
 7 Alice's Adventures in Wonderland_4    alice    30      1
 8 The Picture of Dorian Gray_15         alice     1      1
 9 Alice's Adventures in Wonderland_1    alice    27      1
10 Alice's Adventures in Wonderland_11   alice    16      1
# i 48,781 more rows
```

The given code is used to calculate the topic assignments for all documents contained in the LDA model, utilizing the document-term matrix. This process facilitates the examination of topic distribution throughout the corpus, thereby enabling the identification of patterns pertaining to the thematic content of the documents.

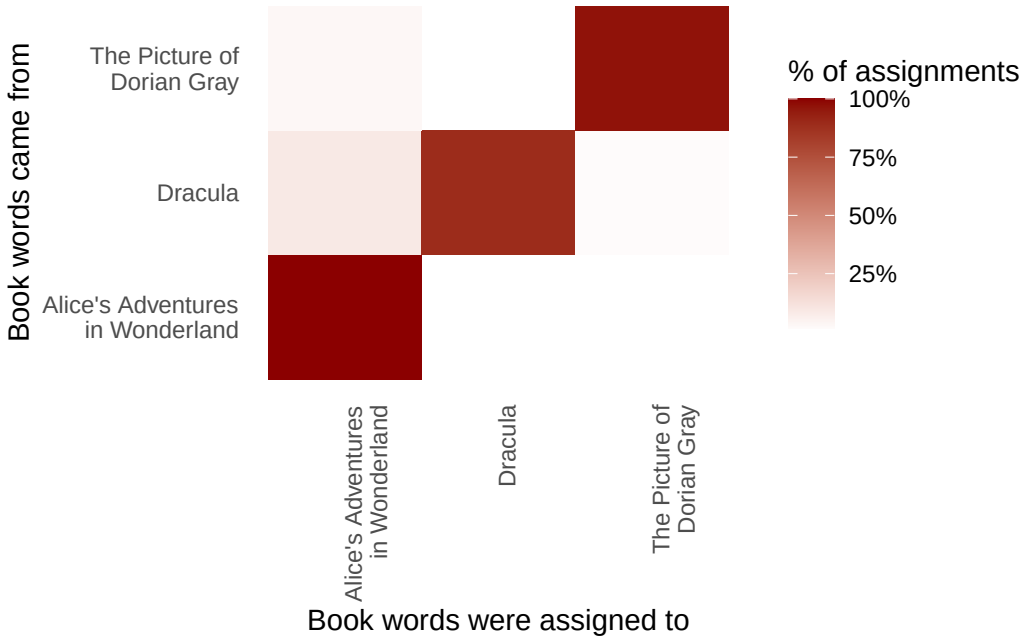
```r
assignments <- assignments %>%
  separate(document, c("title", "chapter"),
           sep = "_", convert = TRUE) %>%
  inner_join(book_topics, by = c(".topic" = "topic"))

assignments
```

```
# A tibble: 41,585 x 6
   title                              chapter term  count .topic consensus
   <chr>                                <int> <chr> <dbl>  <dbl> <chr>
 1 Alice's Adventures in Wonderland         7 alice    50      1 Alice's Adventur~
 2 Alice's Adventures in Wonderland         9 alice    47      1 Alice's Adventur~
 3 Alice's Adventures in Wonderland         6 alice    43      1 Alice's Adventur~
 4 Alice's Adventures in Wonderland         8 alice    39      1 Alice's Adventur~
 5 Alice's Adventures in Wonderland         5 alice    35      1 Alice's Adventur~
 6 Alice's Adventures in Wonderland        10 alice    30      1 Alice's Adventur~
 7 Alice's Adventures in Wonderland         4 alice    30      1 Alice's Adventur~
```

```
 8 The Picture of Dorian Gray            15 alice     1      1 Alice's Adventur~
 9 Alice's Adventures in Wonderland       1 alice    27      1 Alice's Adventur~
10 Alice's Adventures in Wonderland      11 alice    16      1 Alice's Adventur~
# i 41,575 more rows
```

The presented code performs the separation of the title and chapter data from the "document" column of the assignments data frame, assigning them as distinct columns. It subsequently incorporates a new "consensus" column in the "assignments" data frame, based on the prevalent topic for each book. This feature allows for the exploration of topic distribution throughout the corpus, thereby aiding in the identification of trends in the thematic content of the books and chapters.

```
library(scales)

assignments %>%
  count(title, consensus, wt = count) %>%
  mutate(across(c(title, consensus), ~str_wrap(., 20))) %>%
  group_by(title) %>%
  mutate(percent = n / sum(n)) %>%
  ggplot(aes(consensus, title, fill = percent)) +
  geom_tile() +
  scale_fill_gradient2(high = "darkred", label = percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        panel.grid = element_blank()) +
  labs(x = "Book words were assigned to",
       y = "Book words came from",
       fill = "% of assignments")
```



The given code produces a heatmap representation of the topic assignments associated with each book present in the corpus. This visualization enables the observation of topic distribution throughout the books, allowing for the detection of patterns within the thematic content of the corpus.

```
wrong_words <- assignments %>%
  filter(title != consensus)

wrong_words
```

```
# A tibble: 3,359 x 6
   title                      chapter term  count .topic consensus
   <chr>                        <int> <chr> <dbl>  <dbl> <chr>
 1 The Picture of Dorian Gray      15 alice     1      1 Alice's Adventures in ~
 2 Dracula                         47 lord      8      3 The Picture of Dorian ~
 3 Dracula                         28 snow      3      1 Alice's Adventures in ~
 4 Dracula                         34 snow      1      1 Alice's Adventures in ~
 5 The Picture of Dorian Gray       5 sleep     1      1 Alice's Adventures in ~
 6 Dracula                         28 sleep     4      1 Alice's Adventures in ~
 7 Dracula                         34 sleep     6      1 Alice's Adventures in ~
 8 Dracula                         34 dear      1      1 Alice's Adventures in ~
 9 Dracula                         28 dead      1      1 Alice's Adventures in ~
```

```
10 Dracula                          34 dead      8      1 Alice's Adventures in ~
# i 3,349 more rows
```

```r
wrong_words %>%
  count(title, consensus, term, wt = count) %>%
  ungroup() %>%
  arrange(desc(n))
```

```
# A tibble: 3,000 x 4
   title   consensus                          term       n
   <chr>   <chr>                              <chr>  <dbl>
 1 Dracula Alice's Adventures in Wonderland time      29
 2 Dracula Alice's Adventures in Wonderland driver    25
 3 Dracula Alice's Adventures in Wonderland night     24
 4 Dracula Alice's Adventures in Wonderland sea       22
 5 Dracula Alice's Adventures in Wonderland found     19
 6 Dracula Alice's Adventures in Wonderland horses    19
 7 Dracula Alice's Adventures in Wonderland mate      19
 8 Dracula Alice's Adventures in Wonderland wind      18
 9 Dracula Alice's Adventures in Wonderland watch     17
10 Dracula Alice's Adventures in Wonderland white     17
# i 2,990 more rows
```