STAT325: LDA Analysis of Frankenstein

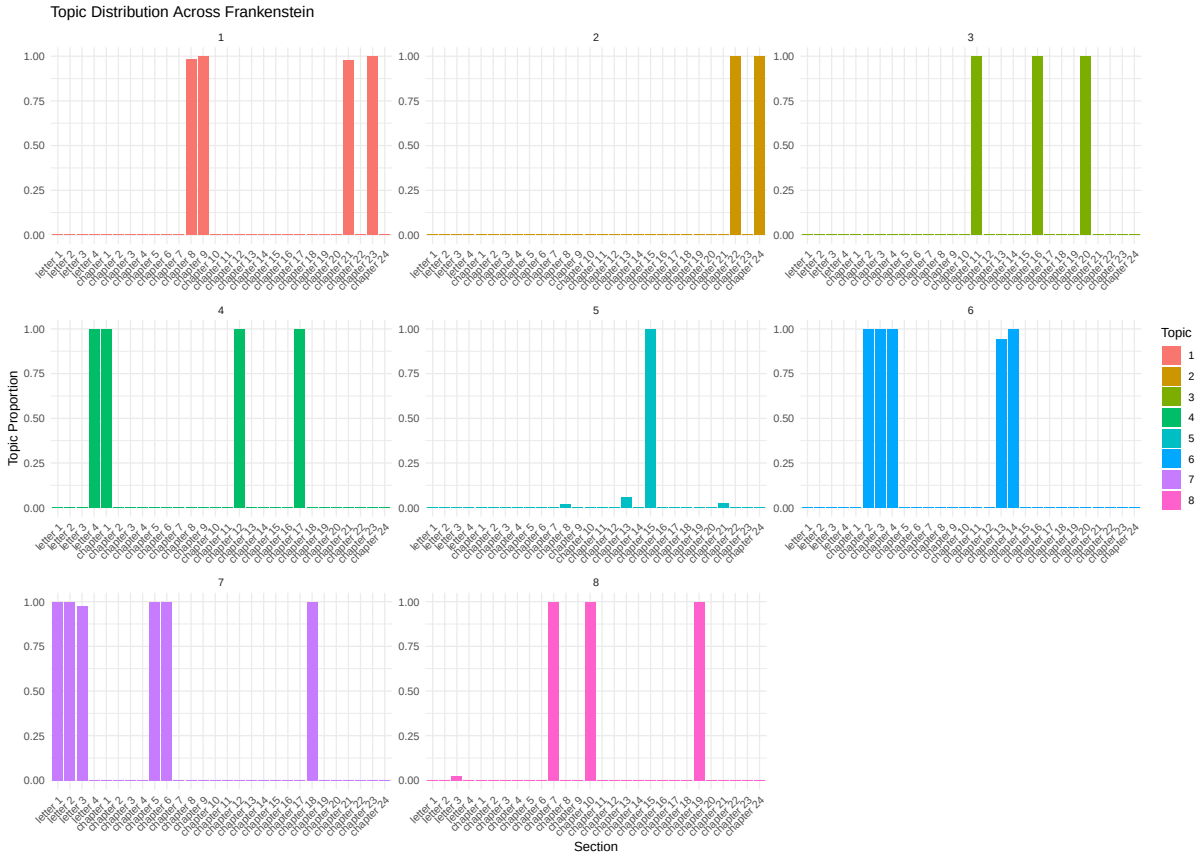
Justin Papagelis (jpapagelis24@amherst.edu)

2024-03-04

```
library(Frankenstein)
library(dplyr)
library(tidyr)
library(tidytext)
library(topicmodels)
library(ggplot2)



other_stop_words <- tibble(  # get rid of common numbers
     word = paste(c(0:24, "Chapter")))


section_names <- c("letter 1", "letter 2", "letter 3", "letter 4", "chapter 1",
              "chapter 2", "chapter 3", "chapter 4", "chapter 5", "chapter 6",
              "chapter 7", "chapter 8", "chapter 9", "chapter 10", "chapter 11",
              "chapter 12", "chapter 13", "chapter 14", "chapter 15",
              "chapter 16", "chapter 17", "chapter 18", "chapter 19",
              "chapter 20", "chapter 21", "chapter 22", "chapter 23",
              "chapter 24")

# Clean and tokenize novel
Frankenstein_LDA <- Frankenstein |>
  filter(section != 0) |>
  mutate(section_label = paste(section_type,section)) |>
  select(-section,-section_type) |>
  unnest_tokens(word, text) |>
  anti_join(stop_words) |>
  anti_join(other_stop_words) |>
  mutate(section_label = factor(section_label, levels = section_names))
```

```r
# Create the document-word matrix
dtm <- Frankenstein_LDA |>
  count(section_label, word) |>
  cast_dtm(section_label, word, n)

# Create LDA model using value of k and seed
lda_model <- LDA(dtm, k = 8, control = list(seed = 1))
```

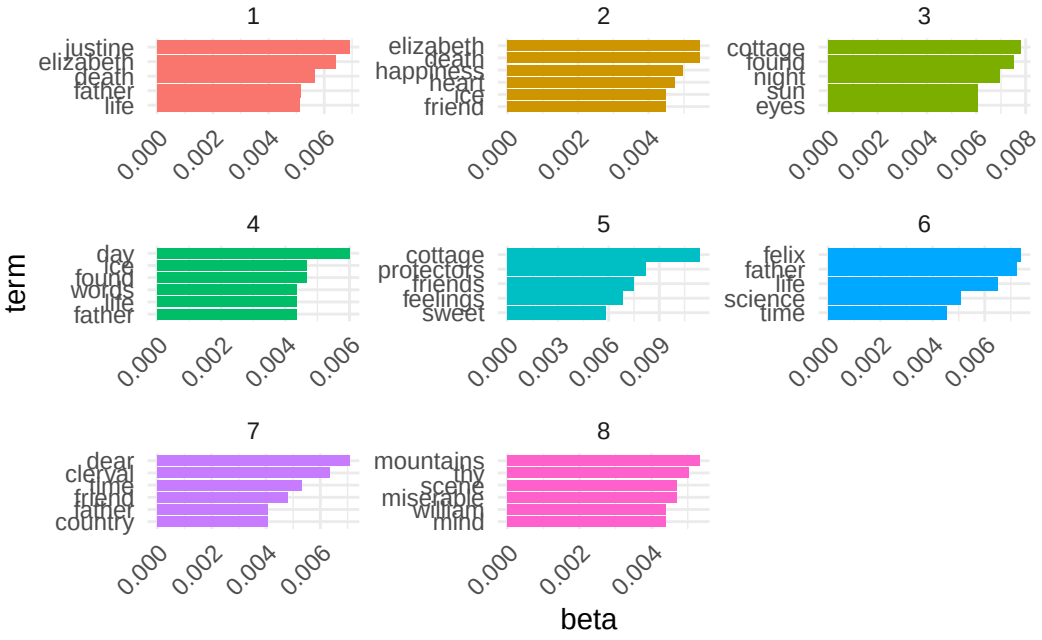
```r
topics_gamma <- tidy(lda_model, matrix = "gamma") |>
  mutate(document = factor(document, levels = section_names))


ggplot(topics_gamma, aes(x = document, y = gamma, fill = factor(topic))) +
  geom_bar(stat ="identity") +
  facet_wrap(~ topic, scales = "free") +
  theme_minimal() +
  labs(x = "Section", y = "Topic Proportion", fill = "Topic",
       title = "Topic Distribution Across Frankenstein") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
topics_beta <- tidy(lda_model, matrix = "beta")
topics_beta
```

```
# A tibble: 52,512 x 3
   topic term       beta
   <int> <chr>     <dbl>
 1     1 11th  3.93e-247
 2     2 11th  2.37e-  4
 3     3 11th  3.31e-264
 4     4 11th  2.00e-263
 5     5 11th  5.27e-261
 6     6 11th  7.20e-267
 7     7 11th  2.53e-  4
 8     8 11th  3.50e-264
 9     1 _to   6.85e-257
10     2 _to   2.37e-  4
# i 52,502 more rows
```

```r
top_terms <- topics_beta |>
  group_by(topic) |>
  slice_max(beta, n = 5) |>
  ungroup() |>
  arrange(topic, -beta)

# Top term models
top_terms |>
  mutate(term = reorder_within(term, beta, topic)) |>
  ggplot(aes(beta, term, fill = factor(topic))) +
  theme_minimal() +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

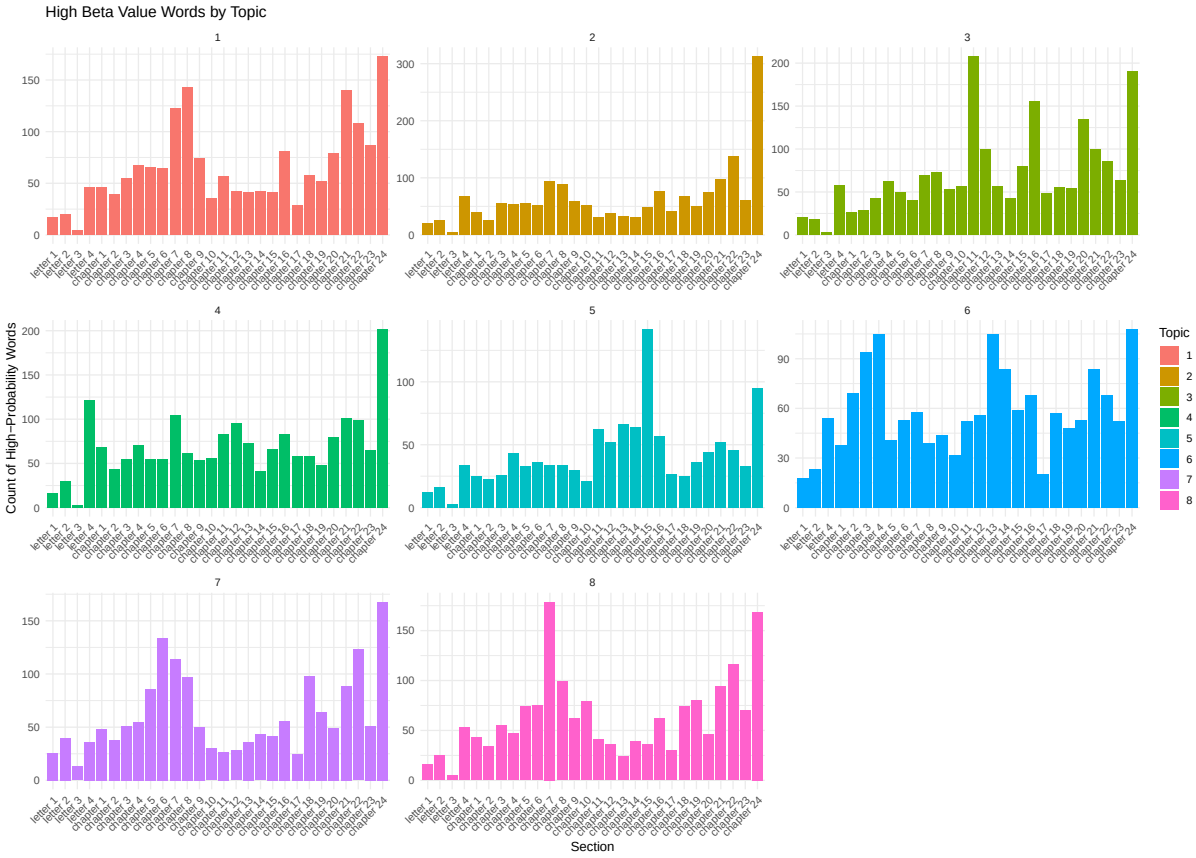



```r
# Extract the top n high beta value words
high_prob_words <- tidy(lda_model, matrix = "beta") |>
  group_by(topic) |>
  top_n(30, beta) |>
  ungroup()
```

```r
frankenstein_terms <- Frankenstein_LDA |>
  rename(term = word)

count_words <- frankenstein_terms |>
  semi_join(high_prob_words, by = "term") |>
  count(section_label, term)

final_count <- count_words |>
  left_join(high_prob_words, by = "term") |>
  select(section_label, term, topic, n) |>
  group_by(section_label, topic) |>
  summarize(total_words = sum(n), by = "section_label")
```

```
`summarise()` has grouped output by 'section_label'. You can override using the
`.groups` argument.
```

```r
ggplot(final_count, aes(x = section_label, y = total_words, fill = factor(topic))) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  facet_wrap(~ topic, scales = "free") +
  labs(x = "Section", y = "Count of High-Probability Words", fill = "Topic",
       title = "High Beta Value Words by Topic") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


High Beta Value Words by Topic
Count of High−Probability Words
Section
Topic
1
2
3
4
5
6
7
8