Topic Modeling

Liwen Yin

Topic Modeling

```
library(topicmodels)
library(tidytext)
library(lexicon)
library(factoextra)
```

```
Loading required package: ggplot2
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggplot2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

##read dataset

```
movies <- read.csv("movie_plots.csv")
movies_gen <- read.csv("movie_plots_with_genres.csv")
```

```
library(dplyr)
library(knitr)
unique_genres <- movies_gen %>%
  select(Genre) %>%
  distinct()
kable(unique_genres, col.names = "Unique Genres")
```

Unique Genres
western
action
sci-fi
history
romance
fantasy
sport
war

There are 8 genres in movie genres dataset. ##Data structure

```
plots_by_word<-movies %>%  unnest_tokens(word, Plot)
plot_word_counts<-plots_by_word%>%
anti_join(stop_words) %>%
count(Movie.Name, word, sort = TRUE)
```

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

```
data("freq_first_names")
first_names <- tolower(freq_first_names$Name)
plot_word_counts <- plot_word_counts %>% filter(!(word %in% first_names))
plots_dtm<-plot_word_counts %>%  cast_dtm(Movie.Name, word, n)
dim(plot_word_counts |> distinct(word))[1]
```

```
[1] 13396
```

```r
dim(movies)
```

```
[1] 1077    2
```

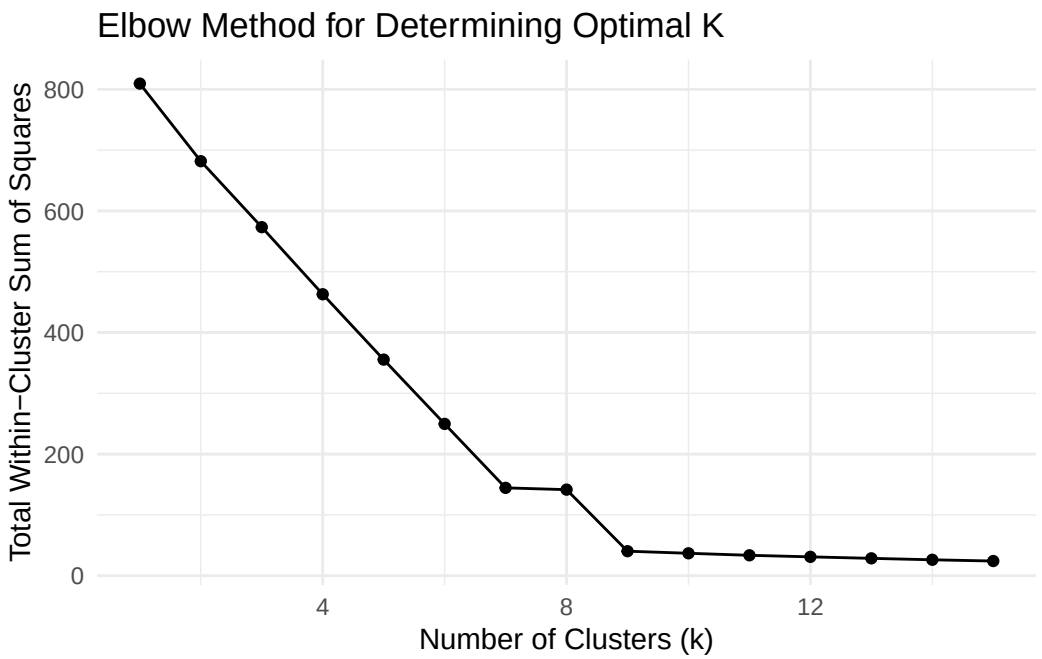
```r
dim(plots_dtm)
```

```
[1]  1063 13396
```

##Elbow Method for Determining Optimal K

```r
library(ggplot2)
library(factoextra)
set.seed(1)
plots_lda <- LDA(plots_dtm, k = 8, control = list(seed = 1))
plots_gamma <- tidy(plots_lda, matrix = "gamma")
plots_gamma_wider<-plots_gamma |> pivot_wider(
names_from = topic,
values_from = gamma)

wss <- numeric()
for (k in 1:15) {
  kmeans_result <- kmeans(plots_gamma_wider %>% select(-document), centers = k, nstart = 25)
  wss[k] <- kmeans_result$tot.withinss
}
elbow_plot <- data.frame(k = 1:15, WSS = wss)
ggplot(elbow_plot, aes(x = k, y = WSS)) +
  geom_line() +
  geom_point() +
  labs(title = "Elbow Method for Determining Optimal K",
       x = "Number of Clusters (k)",
       y = "Total Within-Cluster Sum of Squares") +
  theme_minimal()
```



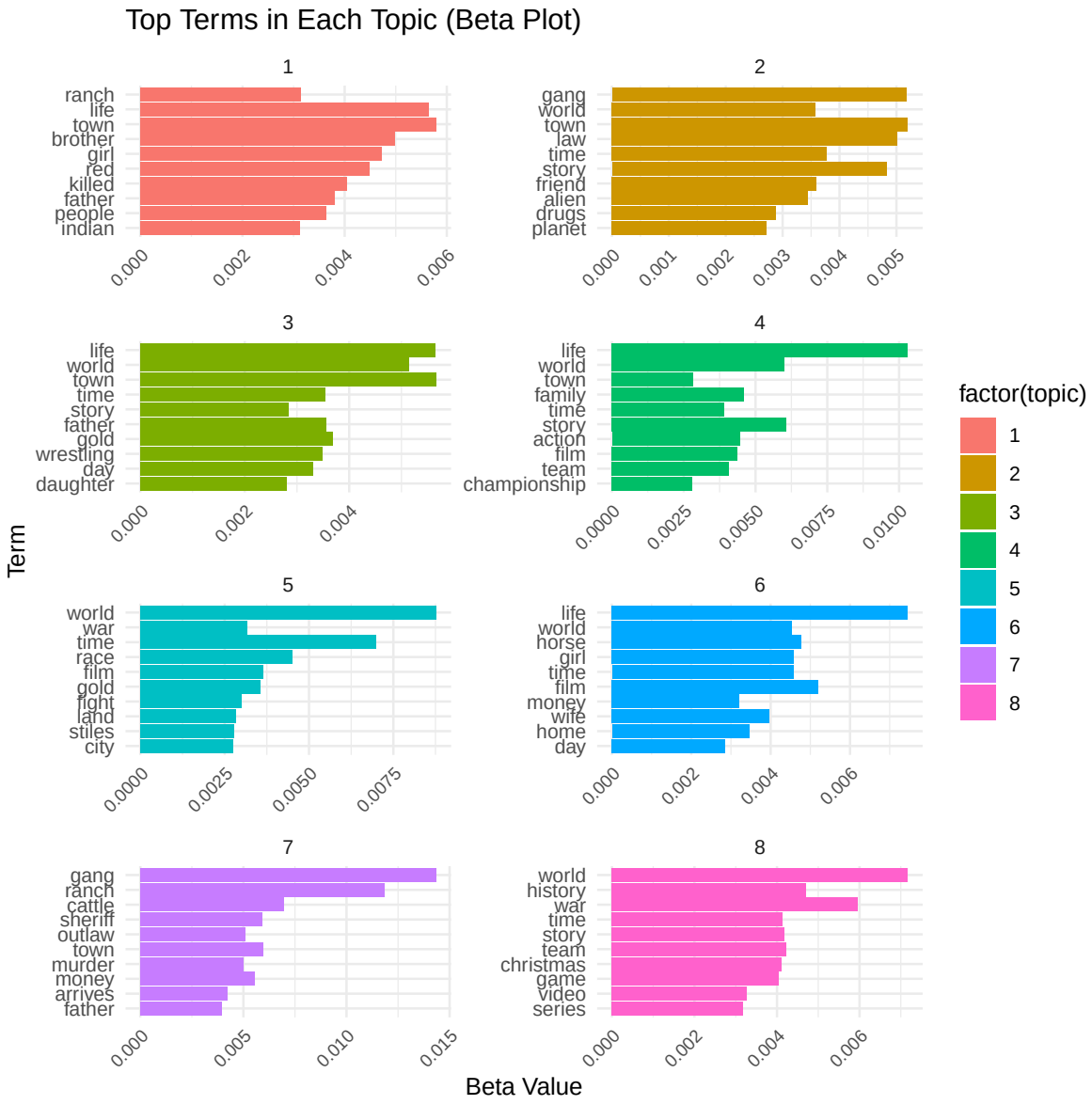
According to the plot above, I would like to choose k=8 to do the following clustering.

```
library(tidytext)
plots_topics <- tidy(plots_lda, matrix = "beta")
top_terms <- plots_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)
print(top_terms)
```

```
# A tibble: 80 x 3
   topic term       beta
   <int> <chr>     <dbl>
 1     1 town    0.00579
 2     1 life    0.00565
 3     1 brother 0.00498
 4     1 girl    0.00472
 5     1 red     0.00448
 6     1 killed  0.00404
 7     1 father  0.00381
 8     1 people  0.00364
 9     1 ranch   0.00314
```

```
10     1 indian  0.00312
# i 70 more rows
```

```
library(tidytext)
library(ggplot2)
plots_topics <- tidy(plots_lda, matrix = "beta")
top_terms <- plots_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)
ggplot(top_terms, aes(x = reorder(term, beta), y = beta, fill = factor(topic))) +
  geom_bar(stat = "identity") +
  facet_wrap(~ topic, scales = "free", ncol = 2) +
  coord_flip() +
  labs(title = "Top Terms in Each Topic (Beta Plot)",
       x = "Term",
       y = "Beta Value") +
  theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8))
```

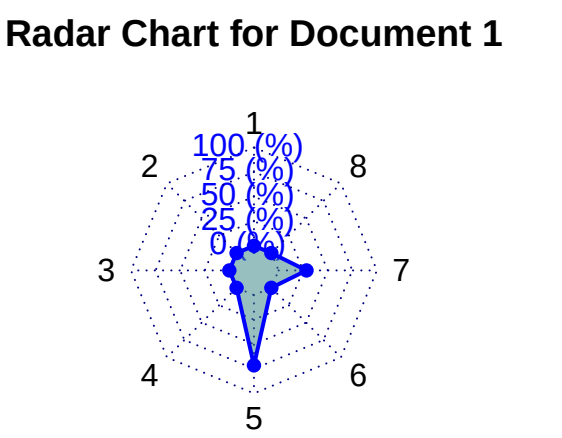



```
ggsave("beta_plot.pdf", height = 10, width = 8)  #
```

Plots above are the beta plots.

```
library(fmsb)
document_index <- 1
radar_data <- plots_gamma_wider[document_index, -1]
radar_data <- rbind(rep(1, ncol(radar_data)), rep(0, ncol(radar_data)), radar_data)
```

```
radarchart(radar_data, axistype = 1, pcol = "blue", pfcol = rgb(0.2, 0.5, 0.5, 0.5), plwd = 2
           title = paste("Radar Chart for Document", document_index))
```

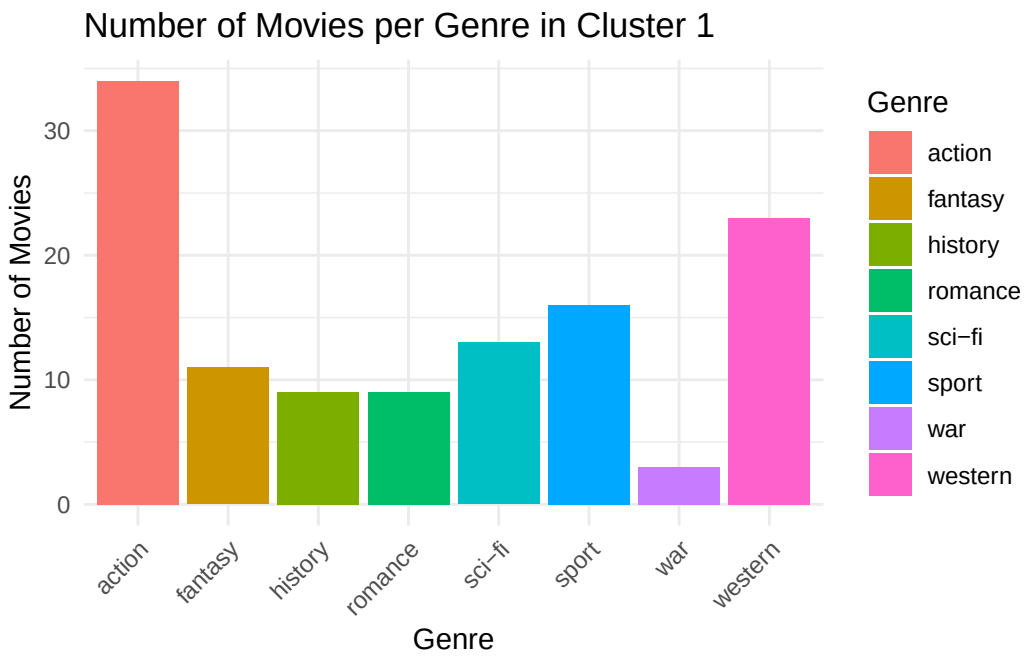


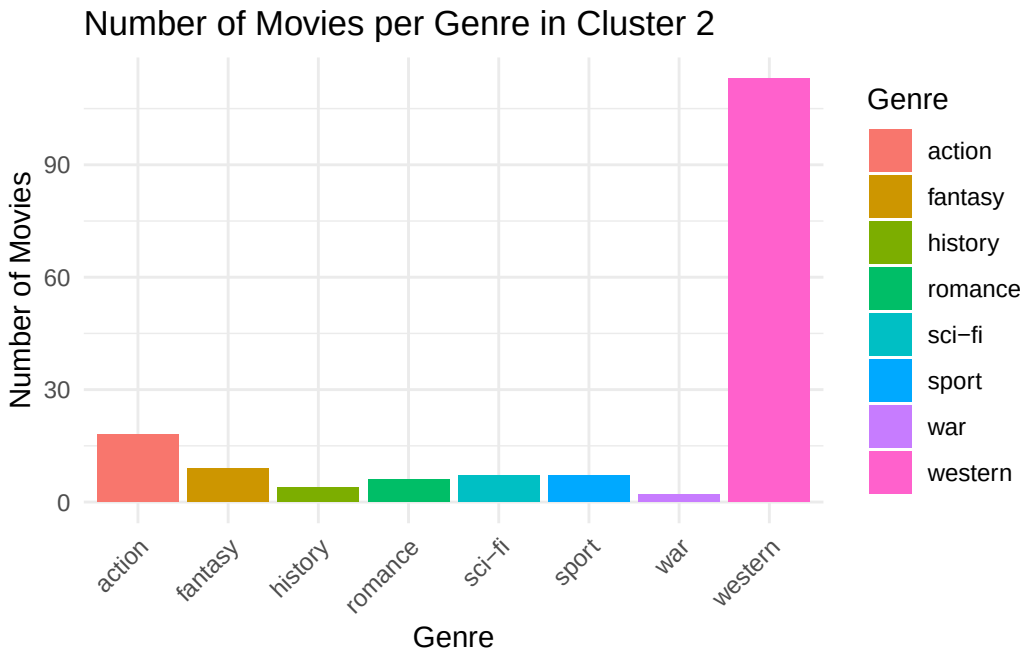
##K-means cluster

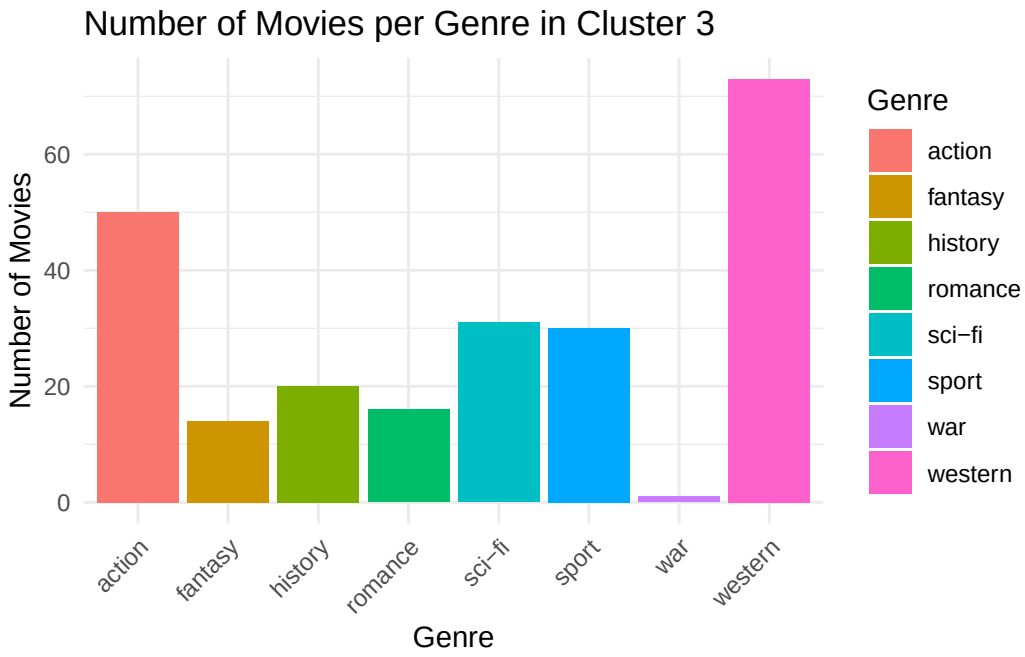
```
plots_gamma_wider_no_na<-plots_gamma_wider %>%  drop_na ()
cluster<-kmeans(plots_gamma_wider %>% select(-document), 8)
fviz_cluster(cluster, data = plots_gamma_wider %>%
select(-document))
```

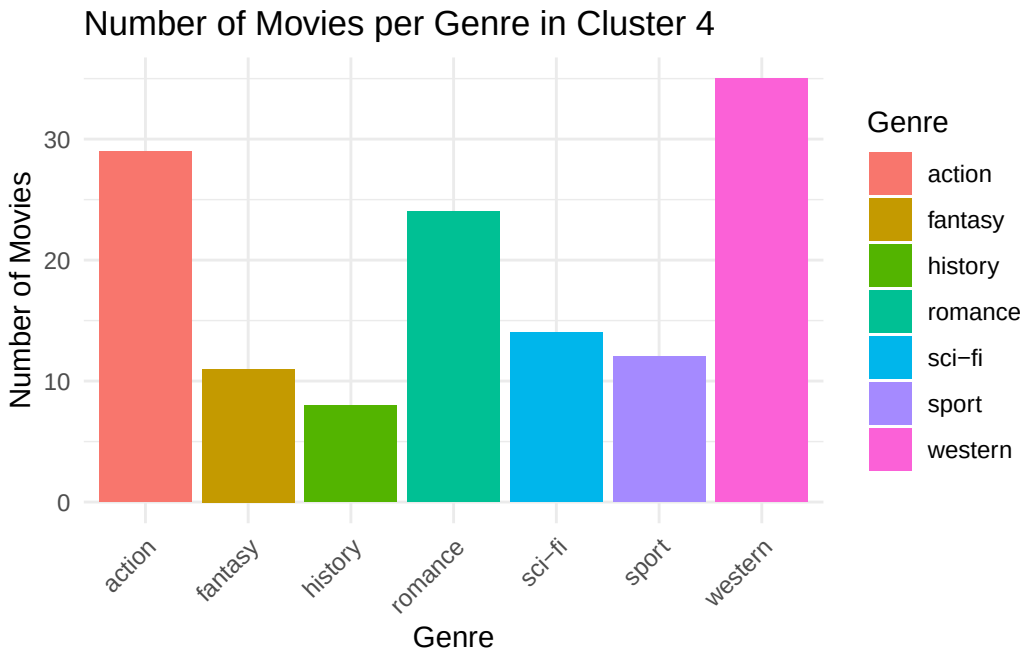
Following plots are the number of movies per genre in different cluster. However, the frequency of some movies have similar type.

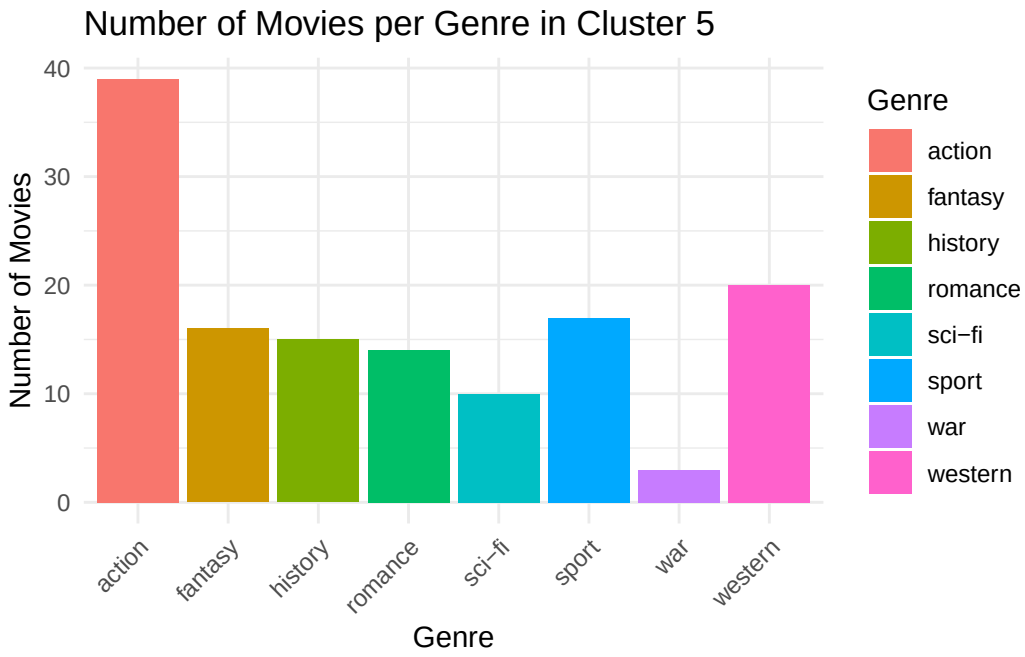
```
clusters <- cluster[["cluster"]]
plots_gamma_wider$cluster <- clusters
for (i in 1:8) {
  plots_cluster <- plots_gamma_wider %>% filter(cluster == i)
  cluster_names <- plots_cluster$document
  cluster_data <- movies_gen %>% filter(Movie.Name %in% cluster_names)
  cluster_counts <- cluster_data %>%
    group_by(Genre) %>%
    summarize(n = n())
  plot <- ggplot(cluster_counts, aes(x = Genre, y = n, fill = Genre)) +
    geom_bar(stat = "identity") +
    labs(title = paste("Number of Movies per Genre in Cluster", i),
         x = "Genre",
         y = "Number of Movies") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
  print(plot)
}
```

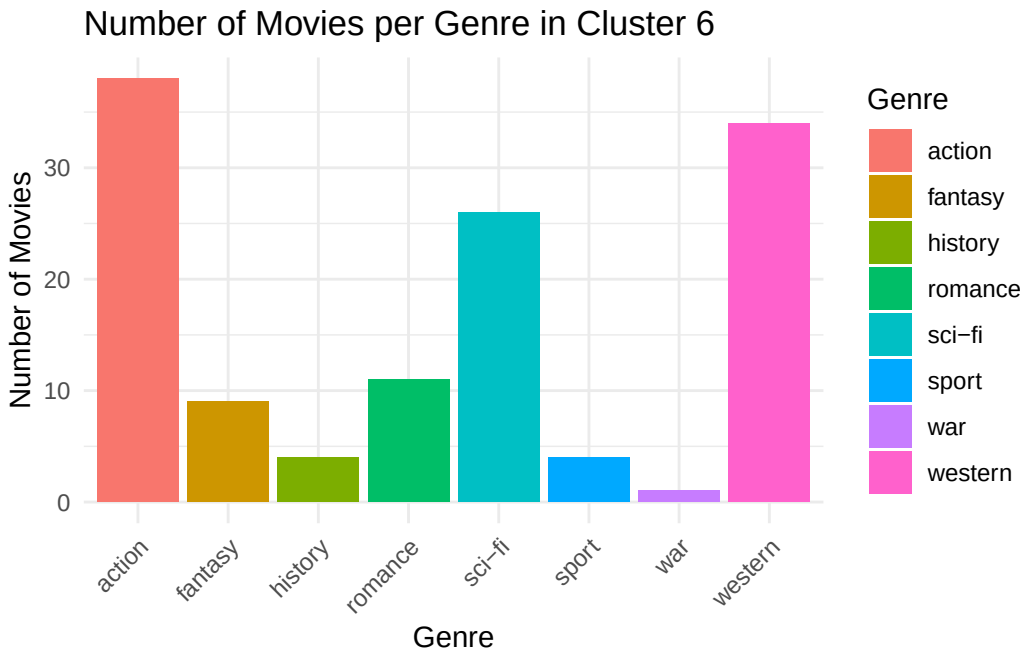

Number of Movies per Genre in Cluster 1
Number of Movies
30
20
10
0
action
fantasy
history
romance
sci−fi
sport
war
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
war
western

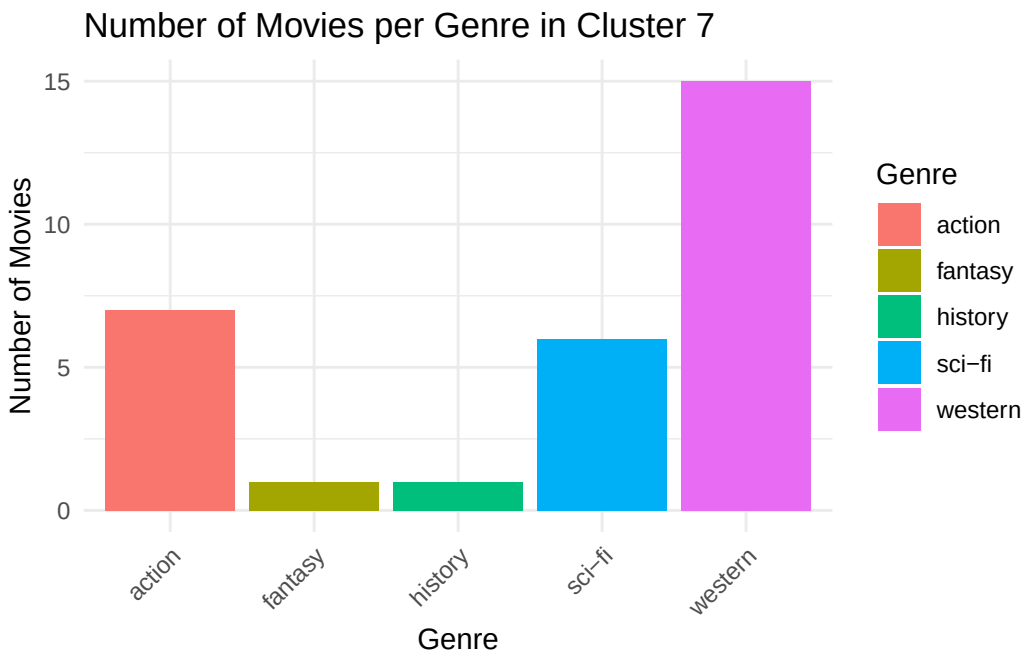

Number of Movies per Genre in Cluster 2
Number of Movies
90
60
30
0
action
fantasy
history
romance
sci−fi
sport
war
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
war
western

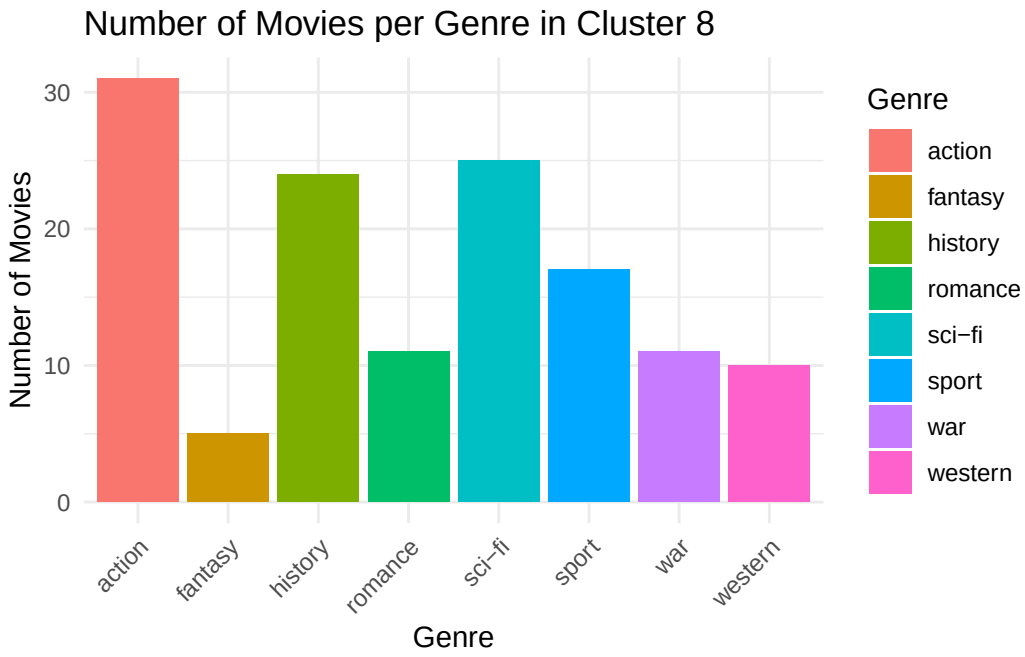

Number of Movies per Genre in Cluster 3
Number of Movies
0
20
40
60
action
fantasy
history
romance
sci−fi
sport
war
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
war
western


Number of Movies per Genre in Cluster 4
Number of Movies
0
10
20
30
action
fantasy
history
romance
sci−fi
sport
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
western


Number of Movies per Genre in Cluster 5
Number of Movies
40
30
20
10
0
action
fantasy
history
romance
sci−fi
sport
war
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
war
western


Number of Movies per Genre in Cluster 6
Number of Movies
30
20
10
0
action
fantasy
history
romance
sci−fi
sport
war
western
Genre
Genre
action
fantasy
history
romance
sci−fi
sport
war
western

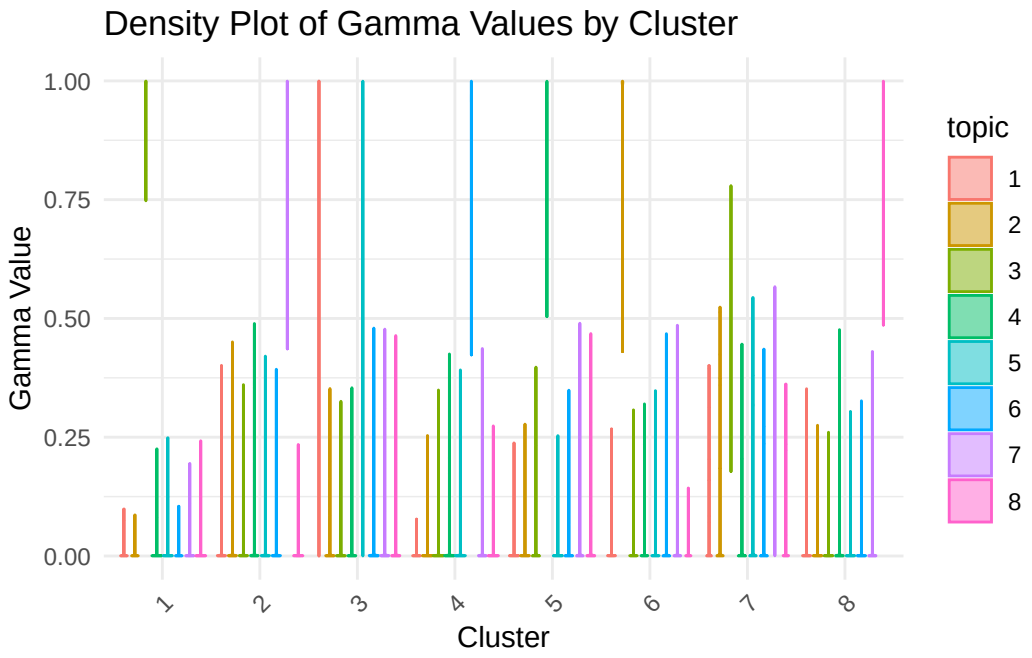





```
clusters <- cluster[["cluster"]]
plots_gamma_wider$cluster <- clusters
plots_gamma_long <- plots_gamma %>%
  left_join(plots_gamma_wider %>% select(document, cluster), by = "document") %>%
```

```
  mutate(topic = as.factor(topic), cluster = as.factor(cluster))

ggplot(plots_gamma_long, aes(x = cluster, y = gamma, color = topic, fill = topic)) +
  geom_violin(alpha = 0.5) +
  labs(title = "Density Plot of Gamma Values by Cluster",
       x = "Cluster",
       y = "Gamma Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

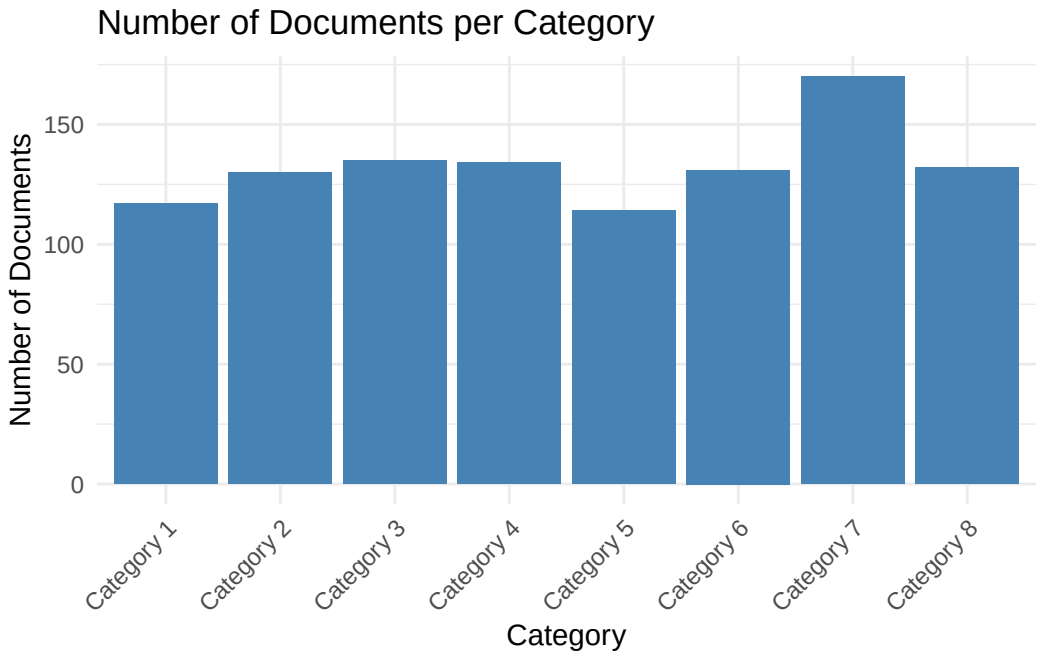



```
library(dplyr)
plots_gamma_classified <- plots_gamma %>%
  group_by(document) %>%
  slice_max(gamma, n = 1) %>%
  ungroup() %>%
  select(document, topic, gamma)
plots_gamma_classified <- plots_gamma_classified %>%
  mutate(topic = as.factor(topic),
         category = paste("Category", topic))
head(plots_gamma_classified)
```

```
# A tibble: 6 x 4
```

```
  document                                          topic gamma category
  <chr>                                             <fct> <dbl> <chr>
1 "\"3x3 the Immersive Fiction\" "                  5     0.998 Category 5
2 "\"All Your Living Needs\" "                      4     0.994 Category 4
3 "\"Beauty and the Beast\" "                       3     0.998 Category 3
4 "\"Because of Meeting You\" "                     1     0.995 Category 1
5 "\"Bees in the Trap\" "                           1     0.830 Category 1
6 "\"Beijing 2008: Games of the XXIX Olympiad\" " 8     0.996 Category 8
```

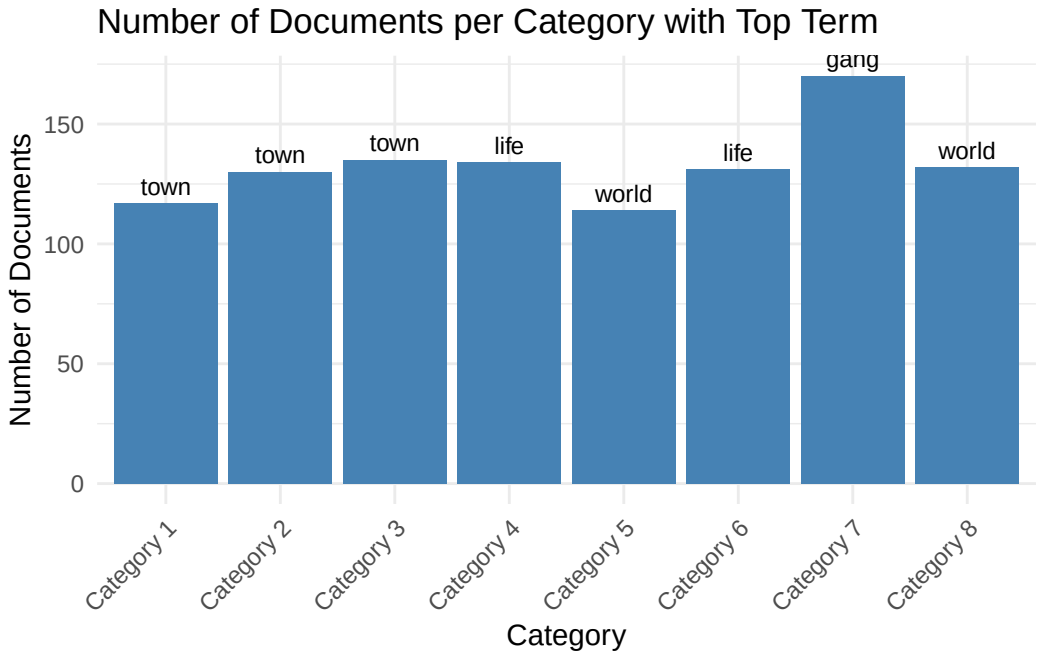
```r
ggplot(plots_gamma_classified, aes(x = category)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Number of Documents per Category",
       x = "Category",
       y = "Number of Documents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
clusters <- cluster[["cluster"]]
plots_gamma_wider$cluster <- clusters
plots_topics <- tidy(plots_lda, matrix = "beta")
library(dplyr)
top_terms_unique <- plots_topics %>%
```

```
  group_by(topic) %>%
  arrange(desc(beta)) %>%
  mutate(rank = row_number()) %>%
  filter(rank <= 10) %>%
  ungroup() %>%
  arrange(topic, rank)
unique_terms_per_topic <- top_terms_unique %>%
  group_by(topic) %>%
  slice(1)
plots_gamma_classified <- plots_gamma_classified %>%
  mutate(topic = as.character(topic))

unique_terms_per_topic <- unique_terms_per_topic %>%
  mutate(topic = as.character(topic))
plots_gamma_classified <- plots_gamma_classified %>%
  left_join(unique_terms_per_topic %>% select(topic, term), by = "topic")
ggplot(plots_gamma_classified, aes(x = category)) +
  geom_bar(fill = "steelblue") +
  geom_text(aes(label = term), stat = "count", vjust = -0.5, size = 3) +
  labs(title = "Number of Documents per Category with Top Term",
       x = "Category",
       y = "Number of Documents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Number of Documents per Category with Top Term
Number of Documents
150
100
50
0
town
town
town
life
world
life
gang
world
Category 1
Category 2
Category 3
Category 4
Category 5
Category 6
Category 7
Category 8
Category