Topic Modeling Assignment

Jin Wen Lin

```
# Load pckages
library(topicmodels)
library(lexicon)
library(factoextra)
```

```
Loading required package: ggplot2

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

```
library(tidytext)
library(ggplot2)
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
```

```
library(tidyr)
```

Exploring and Cleaning Data

Here is the overall structure of the data that we are working on.

```r
# Load data
movie_plots <- read.csv("movie_plots.csv")
glimpse(movie_plots)
```

```
Rows: 1,077
Columns: 2
$ Movie.Name <chr> "Pioneers of the West ", "The Infiltrators ", "\"Graviton: ~
$ Plot       <chr> "Pioneers of the West  :  Caught by the Piutes, pony Expres~
```

First, let’s add a new column named words that include all the words separated from each plot and then remove the stop words such as “and”, and “the” etc. Last but not least, count the number of occurrences of each word.

```r
# split each plot into individual words
plots <- movie_plots %>% unnest_tokens(word, Plot)

# remove the stop words and count the number of occurrences of each word
new_plots <- plots %>% anti_join(stop_words) %>%
  count(Movie.Name, word, sort = TRUE)
```

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

```r
# take a look at the new data
glimpse(new_plots)
```

```
Rows: 47,674
Columns: 3
$ Movie.Name <chr> "Belly of the Beast ", "King of the Pecos ", "A Loving Gent~
$ word       <chr> "jake", "stiles", "meta", "dance", "jim", "kathy", "bill", ~
$ n          <int> 23, 17, 16, 15, 15, 15, 15, 14, 14, 13, 13, 13, 12, 12, 12,~
```

Next, remove the common first names since they might not be meaningful to plot analysis.

```
# list of the common first names
data("freq_first_names")
# convert the common first names into lower cases
lower_names <- tolower(freq_first_names$Name)
# remove the common first names in each plot
new_plots <- new_plots %>% filter(!(word %in% lower_names))
```

LDA (Latent Dirichlet Allocation)

Now, let's construct the DTM (Document-Term Matrix), which calculate the frequency of each word across all the movies.

```
# DTM
plot_dtm <- new_plots %>% cast_dtm(Movie.Name, word, n)
plot_dtm
```

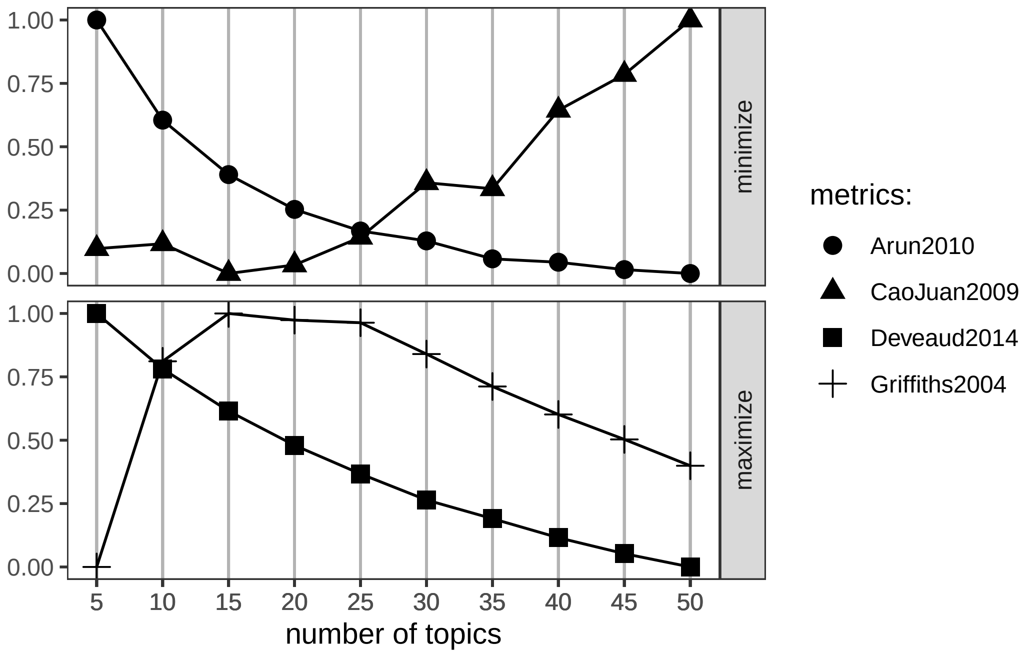
```
<<DocumentTermMatrix (documents: 1063, terms: 13394)>>
Non-/sparse entries: 44143/14193679
Sparsity           : 100%
Maximal term length: 17
Weighting          : term frequency (tf)
```

K Choice

Before creating the topic model of LDA, we are going to first pick an appropriate or optimal k, which represents the number of topics using the package "ldatuning". By going over the documentation of this package, there are many different metrics for determining k. I am going to check all of the suggested metrics to determine the value of k.

```
# load the package
library(ldatuning)

# use the function FindTopicsNumber
k_choice <- FindTopicsNumber(plot_dtm,
                             topics = seq(5, 50, by = 5),
                             metrics = c("Arun2010", "CaoJuan2009",
                                         "Deveaud2014", "Griffiths2004"),
                             method = "Gibbs",
                             control = list(seed = 900)) # set the seed for reproducibility
```

```
# the graph about the k choice
FindTopicsNumber_plot(k_choice)
```



From the above graph, it is easily to see that we want to minimize the metrics "Arun2010" and "CaoJuan2009" and maximize the metrics "Deveaud2014" and "Griffths2004". It is observed that the range of 15 to 25 seems to be the optimal choice for k since "Deveaud2014" and "Griffths2004" attain their maximum range around here while "Arun2010" and "CaoJuan2009" are in the lower range. For the first time, I choose 15 as the k value, the middle of 10 and 20.

LDA Modeling

Here is the LDA modeling with k = 15.

```
# LDA model
lda_plot <- LDA(plot_dtm, k = 15, control = list(seed = 900))
```

Next, we are going to construct the gamma matrix in order to retrieve the probability of each movie falls under each topic.

```r
# Gamma matrix
gamma_matrix <- tidy(lda_plot, matrix = "gamma")
gamma_matrix
```

```
# A tibble: 15,945 x 3
   document                                    topic     gamma
   <chr>                                       <int>     <dbl>
 1 "King of the Pecos "                            1 0.0000605
 2 "French Baroque: Now and Then "                 1 0.0000771
 3 "The Christmas Ornament "                       1 0.0000625
 4 "Hunted by Night "                              1 0.0000897
 5 "Islam in the Heart of the People "             1 0.000163
 6 "Fighting Man of the Plains "                   1 0.253
 7 "Riders of the Purple Sage "                    1 0.0000450
 8 "The Outlaw Josey Wales "                       1 0.0000737
 9 "The Prototypes "                               1 0.0000691
10 "Heroes of the Hills "                          1 0.0000868
# i 15,935 more rows
```

```r
# make the gamma matrix wider for clustering
gamma_wider <- gamma_matrix %>% pivot_wider(names_from = topic,
                                            values_from = gamma)
gamma_wider
```

```
# A tibble: 1,063 x 16
   document            `1`     `2`     `3`     `4`     `5`     `6`     `7`     `8`
   <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 "King of the~ 6.05e-5 6.05e-5 6.05e-5 6.05e-5 9.99e-1 6.05e-5 6.05e-5 6.05e-5
 2 "French Baro~ 7.71e-5 7.71e-5 7.71e-5 7.71e-5 7.71e-5 7.71e-5 7.71e-5 7.71e-5
 3 "The Christm~ 6.25e-5 6.25e-5 6.25e-5 6.25e-5 6.25e-5 6.25e-5 6.25e-5 9.99e-1
 4 "Hunted by N~ 8.97e-5 8.97e-5 8.97e-5 9.99e-1 8.97e-5 8.97e-5 8.97e-5 8.97e-5
 5 "Islam in th~ 1.63e-4 1.63e-4 1.63e-4 1.63e-4 1.63e-4 1.63e-4 1.63e-4 1.63e-4
 6 "Fighting Ma~ 2.53e-1 1.18e-4 1.18e-4 1.18e-4 1.18e-4 1.18e-4 1.18e-4 1.18e-4
 7 "Riders of t~ 4.50e-5 4.50e-5 4.50e-5 4.50e-5 4.50e-5 4.50e-5 4.50e-5 4.50e-5
 8 "The Outlaw ~ 7.37e-5 7.37e-5 7.37e-5 7.37e-5 7.37e-5 7.37e-5 7.37e-5 9.99e-1
 9 "The Prototy~ 6.91e-5 6.91e-5 6.91e-5 6.91e-5 6.91e-5 6.91e-5 6.91e-5 9.99e-1
10 "Heroes of t~ 8.68e-5 8.68e-5 8.68e-5 8.68e-5 8.68e-5 8.68e-5 8.68e-5 8.68e-5
# i 1,053 more rows
# i 7 more variables: `9` <dbl>, `10` <dbl>, `11` <dbl>, `12` <dbl>,
#   `13` <dbl>, `14` <dbl>, `15` <dbl>
```

Now, let's import the data with genres for each movie and linking the clusters to it.

```
# load data
genres <- read.csv("movie_plots_with_genres.csv")
unique(genres$Genre)
```

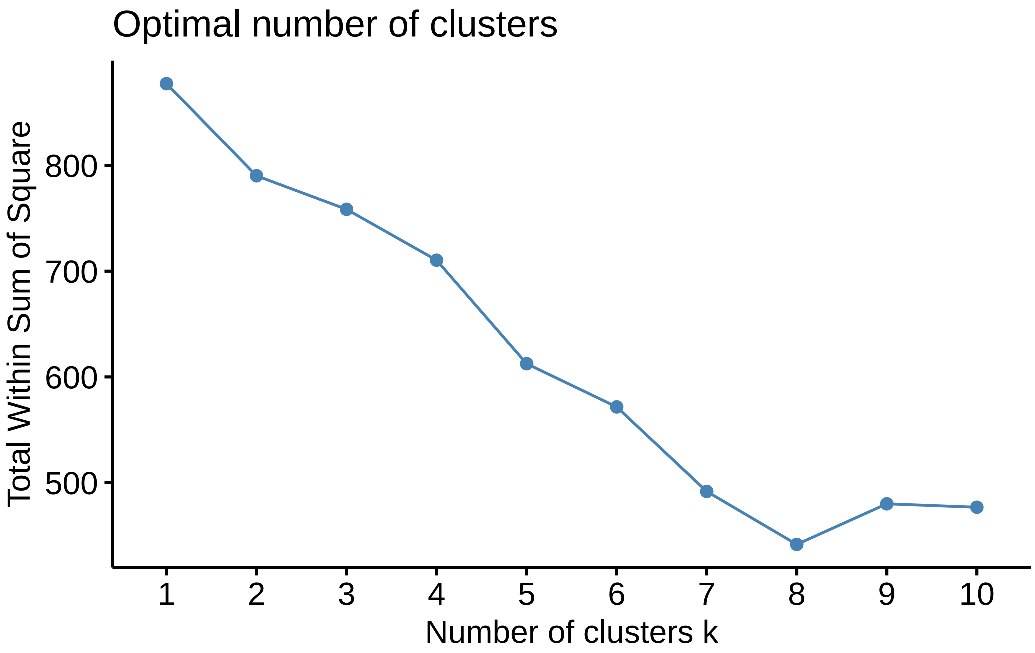
```
[1] "western" "action"  "sci-fi"  "history" "romance" "fantasy" "sport"
[8] "war"
```

Notice there are 8 different types of genres. As a result, we are going to check if the number of clusters here match with the number of generes.

Cluster Plot

Below is the scree plot based on the gamma matrix constructed above.

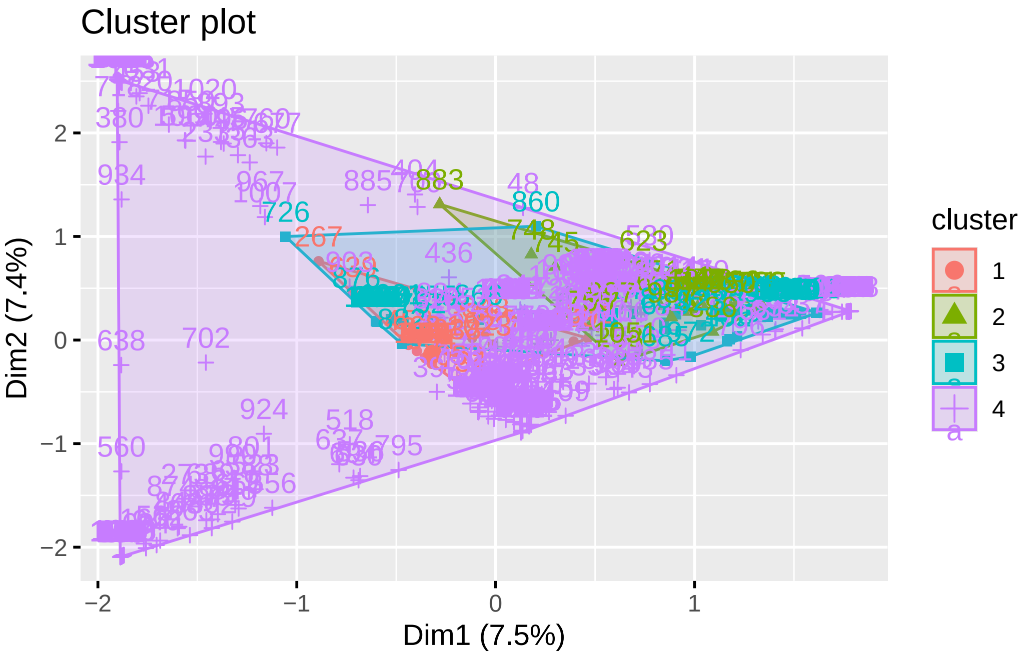
```
# set seed
set.seed(900)
# drop na values from the gamma matrix
gamma_wider_update <- gamma_wider %>% drop_na()
# select the numeric parts of the gamma matrix
new_gamma <- gamma_wider_update %>% select(where(is.numeric))
# the scree plot (WSS)
fviz_nbclust(new_gamma, kmeans, method = "wss")
```



By looking at the scree plot above (Within Sum of Square, elbow method), we can see that the elbow point here is around 4 and 5 clusters, where there seems to be a significantly drop of WSS near 4 and 5. Let's set the number of clusters to be 4 first to see how it goes.

Here is the cluster plot for 4 clusters.

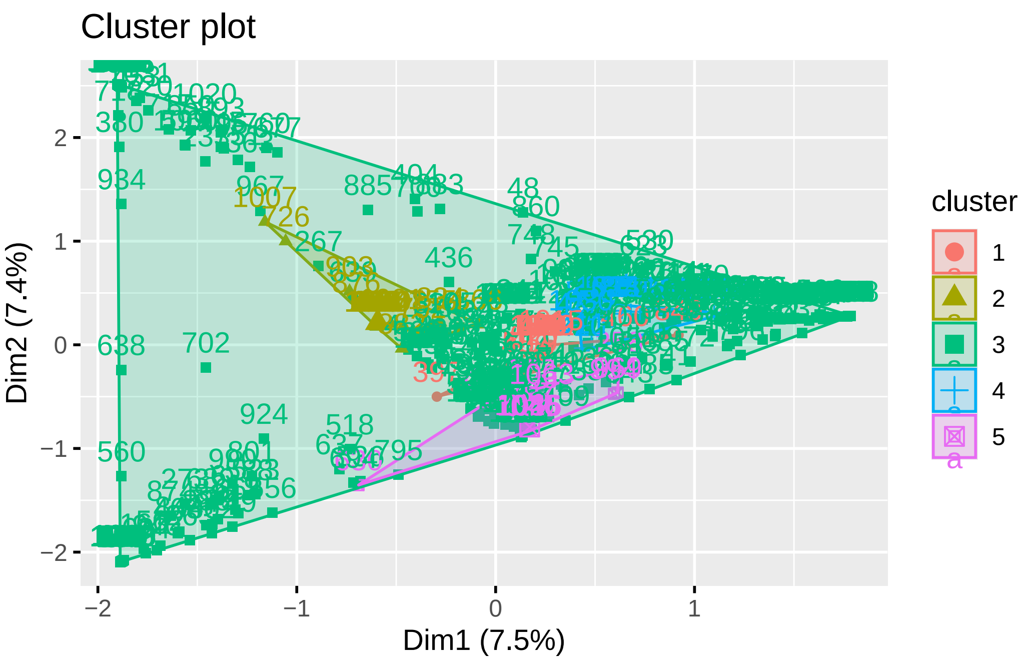
```
# cluster
cluster <- kmeans(gamma_wider_update %>% select(-document), 4)

# cluster plot
fviz_cluster(cluster, data = gamma_wider_update %>% select(-document))
```



From the above cluster plot, we can see that cluster 4 has the largest spread. Notice that other clusters are overlapping with each other. This shows somehow they share similar topics or themes. As a result, this might not distinguish different topics for each movie.

Now, let's set the number of clusters to 5 and see what it looks like.

```r
# cluster
cluster_2 <- kmeans(gamma_wider_update %>% select(-document), 5)

# cluster plot
fviz_cluster(cluster_2, data = gamma_wider_update %>% select(-document))
```



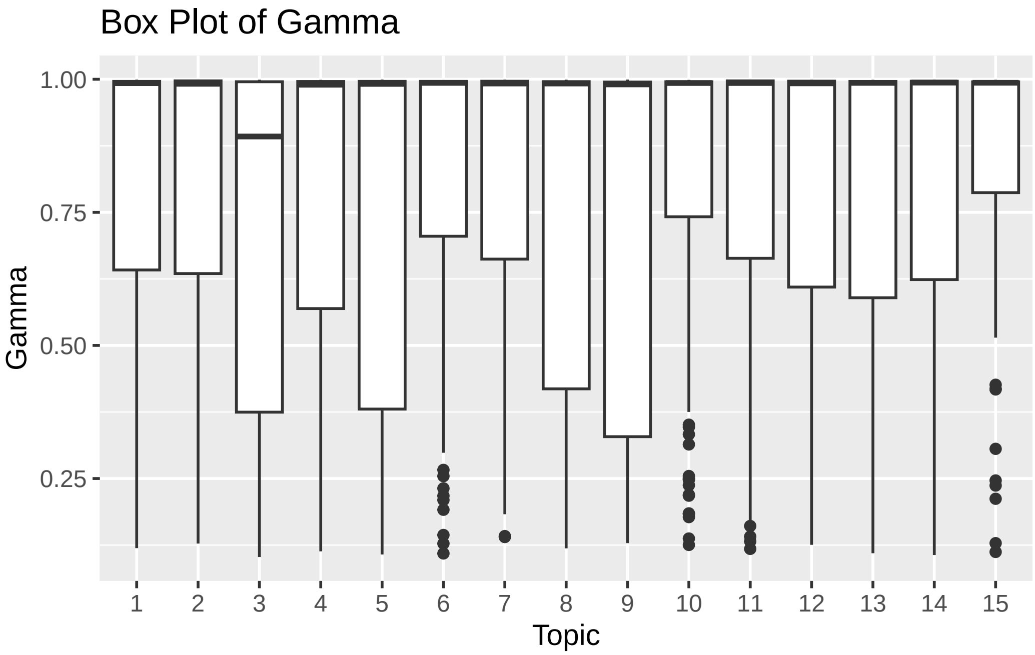
Again, there is a large spread in cluster 3. We can see that clusters 1, 2, 4, and 5 are separated from each other. This may indicates that they are some specific or narrower topics, where as the topic of cluster 3 is more general. Overall, choosing 5 clusters seems to be better.

Gamma Plot

Here are the gamma plots based on the gamma matrix.

Below is the box plot of gamma, which shows the overall distribution of topic probabilities across all the movies. Notice there are many gamma values are very close to 0, hence will have a result of clustering at 0. Hence, I removed all those values and instead of focusing on all of them, I am just going to investigate the gamma values greater than or equal to 1.

```r
# remove the probabilities that are very low (< 0.1)
gamma_for_plot <- gamma_matrix %>% filter(gamma >= 0.1)
# boxplot
ggplot(gamma_for_plot, aes(x = factor(topic), y = gamma)) +
  geom_boxplot() + labs(x = "Topic", y = "Gamma", title = "Box Plot of Gamma")
```

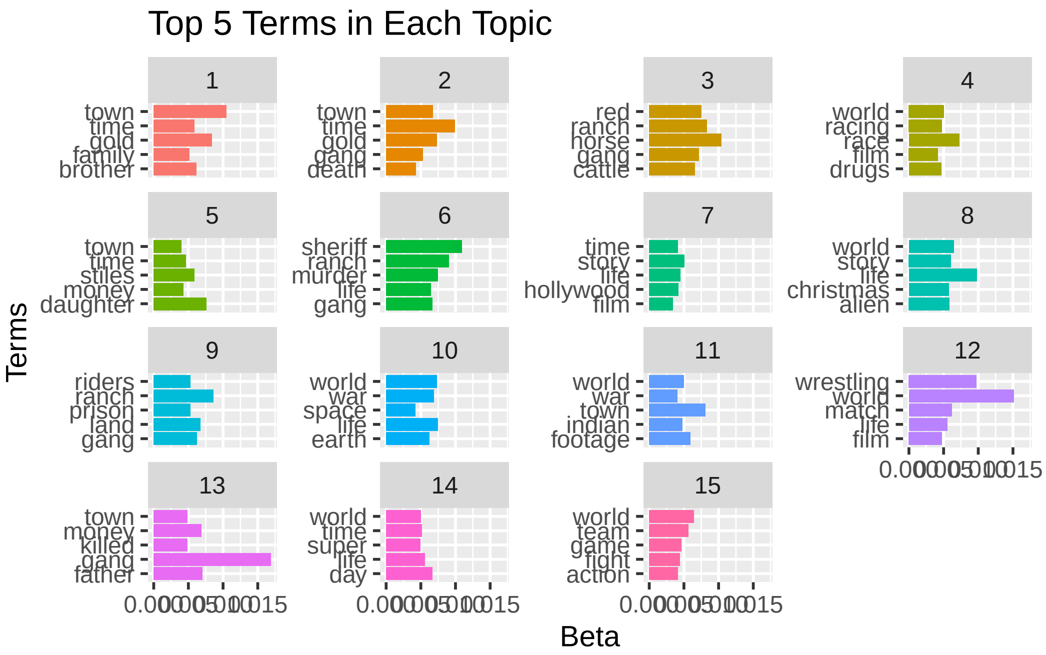


This box plot is showing that most of the topics have the median 1 except topic 3, which has a lower median here. There are also some outliers in topics 6, 7, 10, 11, and 15. Topics 3, 5, 8, and 9 have taller boxes compared to others, which means there might exisit more variability comapred to the shorter boxes.

Beta Plot

Now we are going to look at the beta plot to see the words that best represent the topics.

```
# set seed
set.seed(900)
# extract beta matrix
beta_matrix <- tidy(lda_plot, matrix = "beta")

# choose the top 5 words that appeared the most in each topic and reorder them
top_5 <- beta_matrix %>% group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

# bar plot
ggplot(top_5, aes(x = beta, y = term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y") +
  labs(title = "Top 5 Terms in Each Topic", x = "Beta", y = "Terms")
```



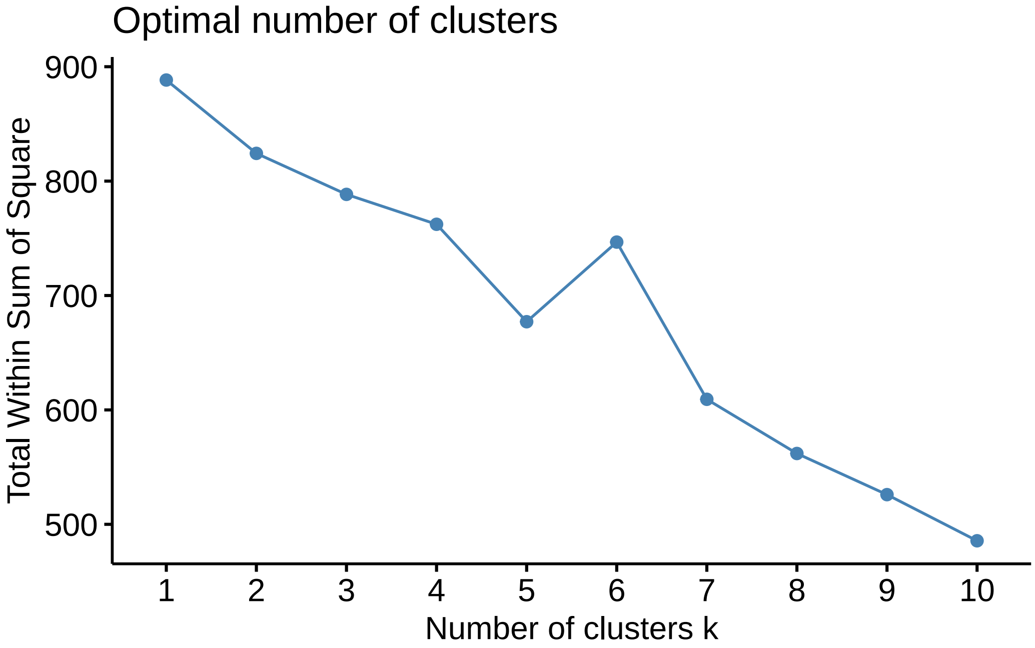
The above graph shows the top 5 words that appeared the most in each topic. We can see that some of the top 1 words appeared in multiple topics. For example the word "time" appeared as top 1 in topics 1, and 2 etc. Hence, there might be some similarities between these topics.

To explore whether there is a better result, we might want to change the number of topics or number of clusters etc. to check if there is any difference.

Change k Value and Number of Clusters

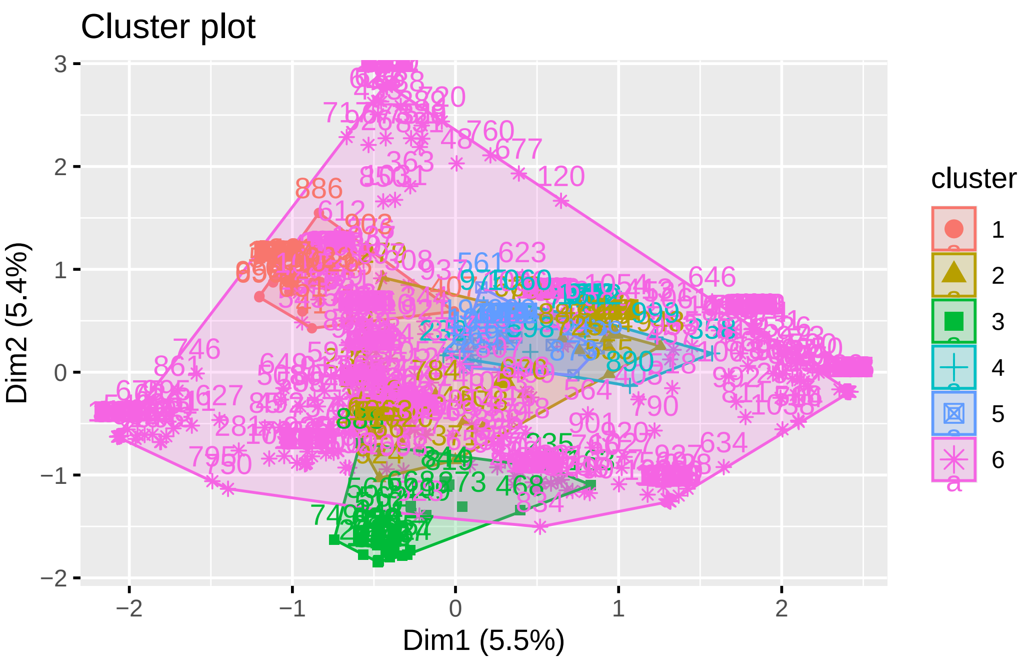
Instead of choosing k = 15, this time change it to 20 first to see what happens.

```
# lda model 2
lda_2 <- lda_plot <- LDA(plot_dtm, k = 20, control = list(seed = 900))

# # Gamma matrix
gamma_2 <- tidy(lda_2, matrix = "gamma")

# make the gamma matrix wider for clustering
gamma_wider_2 <- gamma_2 %>% pivot_wider(names_from = topic,
                                                          values_from = gamma) %>% drop_na()
new_gamma_2 <- gamma_wider_2 %>% select(where(is.numeric))
# the scree plot (WSS)
fviz_nbclust(new_gamma_2, kmeans, method = "wss")
```



It seems like the number of clusters here is 6 since there is a significantly drop of WSS here. Therefore, we are going to set the number of clusters to 6 for this model.

```
set.seed(900)
# cluster
cluster_2 <- kmeans(gamma_wider_2 %>% select(-document), 6)

# cluster plot
fviz_cluster(cluster_2, data = gamma_wider_2 %>% select(-document))
```



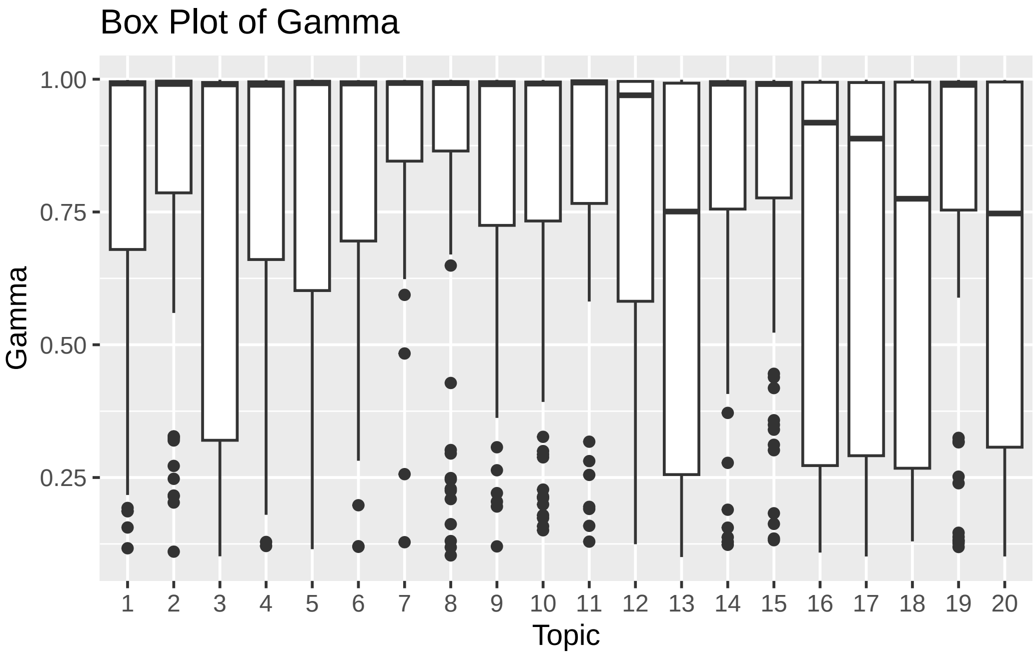
By changing k to 20, we can see that there is a large spread in cluster 6. There are some overlapping clusters such as cluster 1, 2, 4, and 5.

Next, we are going to look at the gamma and beta plots for model 2.

Gamma Plot

Here is the gamma plot for model 2.

```
# remove the probabilities that are very low (< 0.1)
gamma_for_plot_2 <- gamma_2 %>% filter(gamma >= 0.1)
# boxplot
ggplot(gamma_for_plot_2, aes(x = factor(topic), y = gamma)) +
  geom_boxplot() + labs(x = "Topic", y = "Gamma", title = "Box Plot of Gamma")
```

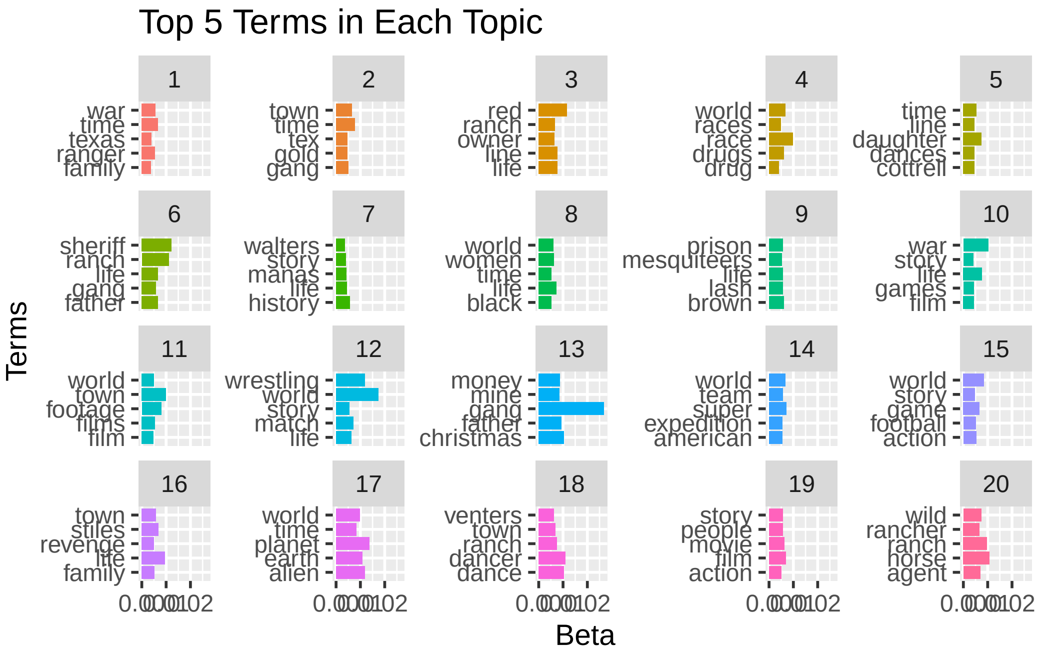


We can see that topics 12, 14, 16, 17, 18, and 20 with the medians not equal to 1. Topics from 1 to 11, 14, 15, and 19 have medians of 1. Topics 3, 13, 16, 17, 18, and 20 have relative longer boxes compared to others, which are showing some variabilites here.

Beta Plot

Here is the beta bar plot for model 2.

```r
# extract beta matrix
beta_2 <- tidy(lda_2, matrix = "beta")

# choose the top 5 words that appeared the most in each topic and reorder them
top_5_ii <- beta_2 %>% group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

# bar plot
ggplot(top_5_ii, aes(x = beta, y = term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y") +
  labs(title = "Top 5 Terms in Each Topic", x = "Beta", y = "Terms")
```



Notice that the words like “war”, “life, and”time” etc. appeared multiple times in each topic. Hence, we can say that these topics do somehow share similarities and therefore do not really distinguish the movie topics.

To conclude, by changing the values like k and the number of clusters, we can see that there still exist some similarities between each topic, and cause the overlapping of multiple clusters shown in the cluster plots. The results are quite similar in model 1 and model 2.

Word Cloud

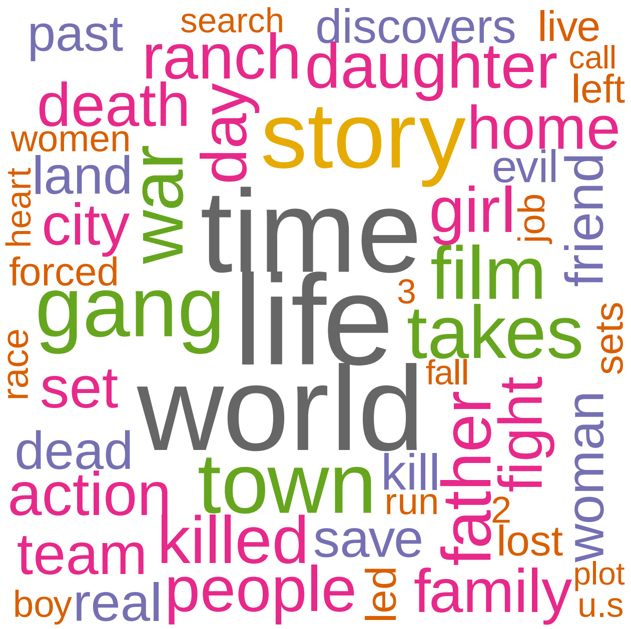
Here is the word cloud using the cleaned plots data (new_plots).

```
# run the package
library(wordcloud)
```

```
Loading required package: RColorBrewer
```

```
# word cloud
set.seed(900)
words_data <- new_plots %>% count(word, sort = TRUE)
# word cloud
```

```r
wordcloud(words = words_data$word, freq = words_data$n, max.words = 200,
          random.order = FALSE, colors=brewer.pal(8, "Dark2"))
```


past
search
discovers
live
ranch
daughter
call
left
death
day
story
home
women
land
war
evil
heart
city
time
girl
job
friend
forced
film
gang
life
takes
race
set
world
fall
sets
dead
kill
father
fight
woman
action
town
run
team
killed
save
lost
boy
real
people
led
family
plot
u.s