Topic Modeling

Topic Modeling

```
library(tidyverse)
library(lexicon)
library(factoextra)
library(tidytext)
library(tidygraph)
library(ldatuning)
library(topicmodels)
library(wordcloud)
library(RColorBrewer)
set.seed(2024)
```

Reading in the data:

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

Unnesting tokens using tidytext:

```
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)`
```

Removing common first names using the 'lexicon package'

```
data("freq_first_names")
first_names <- tolower(freq_first_names$Name)
plot_word_counts <- plot_word_counts %>% filter(!(word %in% first_names))
```

Casting our word counts to a document term matrix

```
plots_dtm <- plot_word_counts %>% cast_dtm(Movie.Name, word, n)
```

Before LDA a look at the dimensions of our matrix:

```
#Distinct words
dim(plot_word_counts %>% distinct(word))[1]
```

```
[1] 13394
```

```
dim(movies)
```

```
[1] 1077    2
```

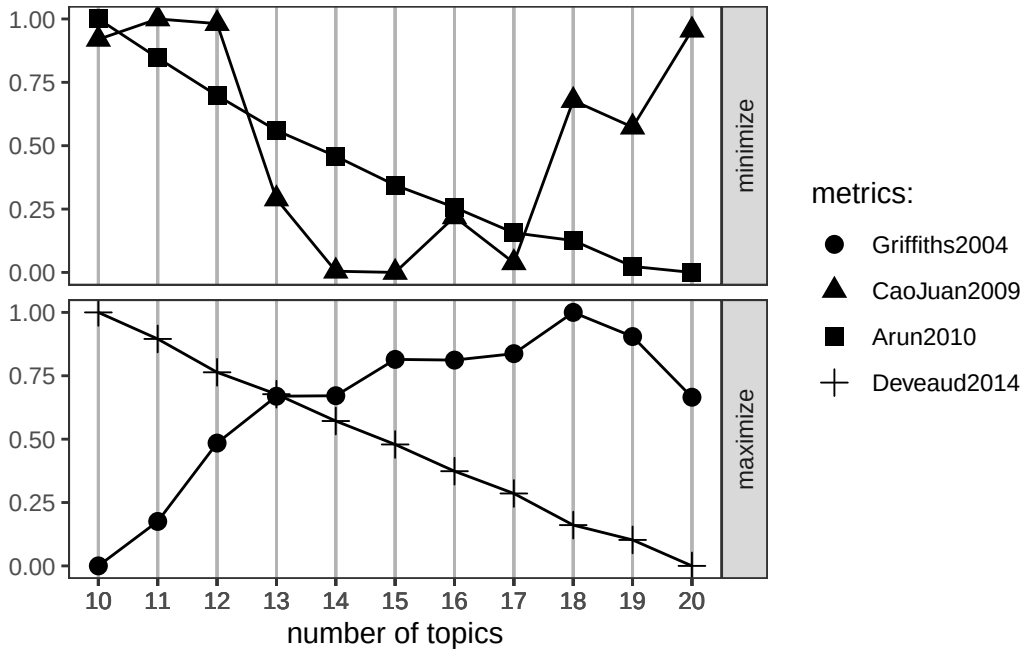
Find the best number of topics for LDA

```
optimal.topics <- FindTopicsNumber(plots_dtm, topics = seq(10, 20, by = 1), metrics = c("Gri
```

```
fit models... done.
calculate metrics:
  Griffiths2004... done.
  CaoJuan2009... done.
  Arun2010... done.
  Deveaud2014... done.
```

```
FindTopicsNumber_plot(optimal.topics)
```

```
Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
of ggplot2 3.3.4.
i The deprecated feature was likely used in the ldatuning package.
  Please report the issue at <https://github.com/nikita-moor/ldatuning/issues>.
```



LDA with 20 topics

```
plots_lda <- LDA(plots_dtm, k = 20, control = list(seed = 2024))
```

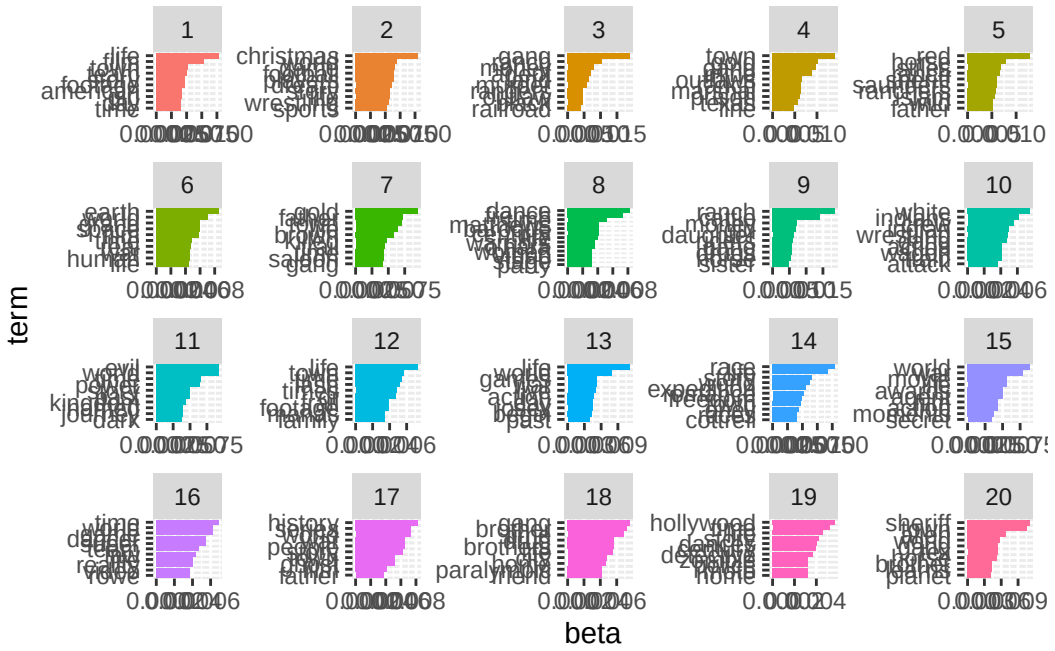
Retrieving gamma and beta:

```
plots_gamma <- tidy(plots_lda, matrix = "gamma")
plots_beta <- tidy(plots_lda, matrix="beta")
```

word-topic probability

```
beta_terms <- plots_beta %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)

beta_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()
```

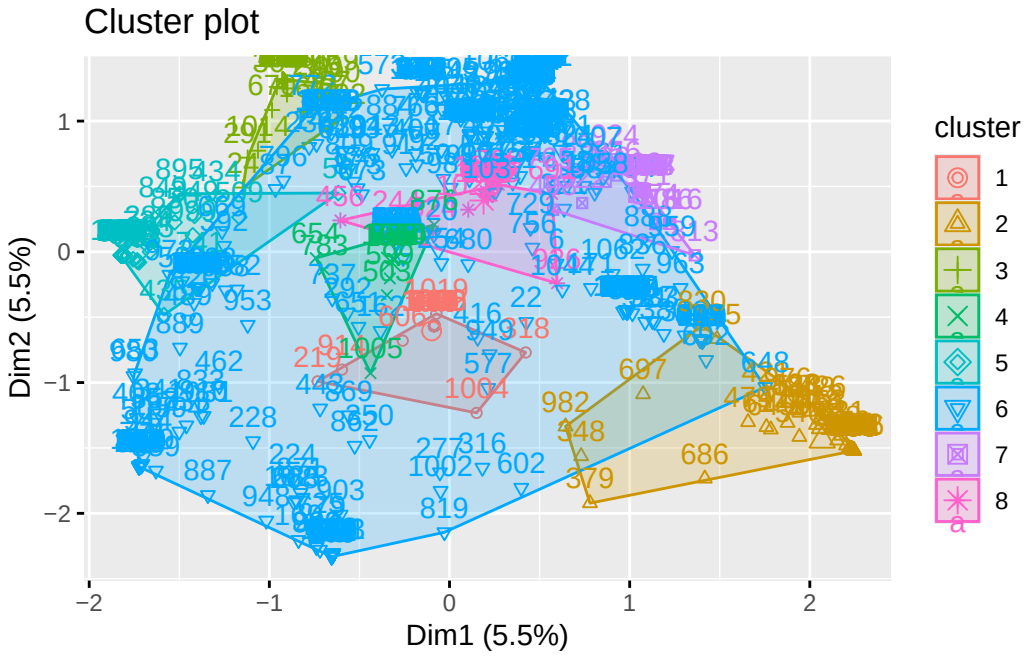


Pivoting the plots_gamma table wider so we can cluster by gammas for each topic

```
plots_gamma_wider <- plots_gamma %>% pivot_wider(
  names_from = topic,
  values_from = gamma
)
```

Clustering. We are considering 8 genres. So lets try 8 clusters

```
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))
```



Let's look at the genres in each cluster: So we'll read in the data with the genres

```
english_movies_with_genres <- read.csv("movie_plots_with_genres.csv")
clusters <- cluster[["cluster"]]
plots_gamma_wider$cluster <- clusters
```

Word cloud Function

```
Cloud <- function(x){
  wordcloud(words = x$Genre, freq = x$n, min.freq = 1,
          max.words = 100, random.order = FALSE,
          colors = brewer.pal(8, "Dark2"))
}
```

All Cluster:

```
cluster_names <- plots_gamma_wider$document
cluster <- english_movies_with_genres %>% filter(Movie.Name %in% cluster_names)
cluster_counts <- cluster %>% group_by(Genre) %>% summarize(n = n())
Cloud(cluster_counts)
```

Cluster 1:

```
plots_clusters1 <- plots_gamma_wider %>% filter(cluster == 1)
cluster_1_names <- plots_clusters1$document
cluster_1 <- english_movies_with_genres %>% filter(Movie.Name %in% cluster_1_names)
cluster_1_counts <- cluster_1 %>% group_by(Genre) %>% summarize(n = n())
Cloud(cluster_1_counts)
```

Cluster 2:

```
plots_clusters2 <- plots_gamma_wider %>% filter(cluster == 2)
cluster_2_names <- plots_clusters2$document
cluster_2 <- english_movies_with_genres %>% filter(Movie.Name %in% cluster_2_names)
cluster_2_counts <- cluster_2 %>% group_by(Genre) %>% summarize(n = n())
Cloud(cluster_2_counts)
```


fantasy
action
history
western
romance
sci−fi
sport