Topic Modelling

Jordan Stout

2024-11-8

```r
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 ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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 become error
```

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

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

```r
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
set.seed(100)

knitr::opts_chunk$set(echo = TRUE)
```

```r
movies <- read.csv("movie_plots_with_genres.csv")
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)`
```

```r
data("freq_first_names")
first_names <- tolower(freq_first_names$Name)

plot_word_counts <- plot_word_counts |>
  filter(!(word %in% first_names))

dtm <- plot_word_counts |>
  cast_dtm(Movie.Name, word, n)

lda <- LDA(dtm, k = 20, control = list(seed=100))

top_terms <- terms(lda, 10)
```

Extract greeks

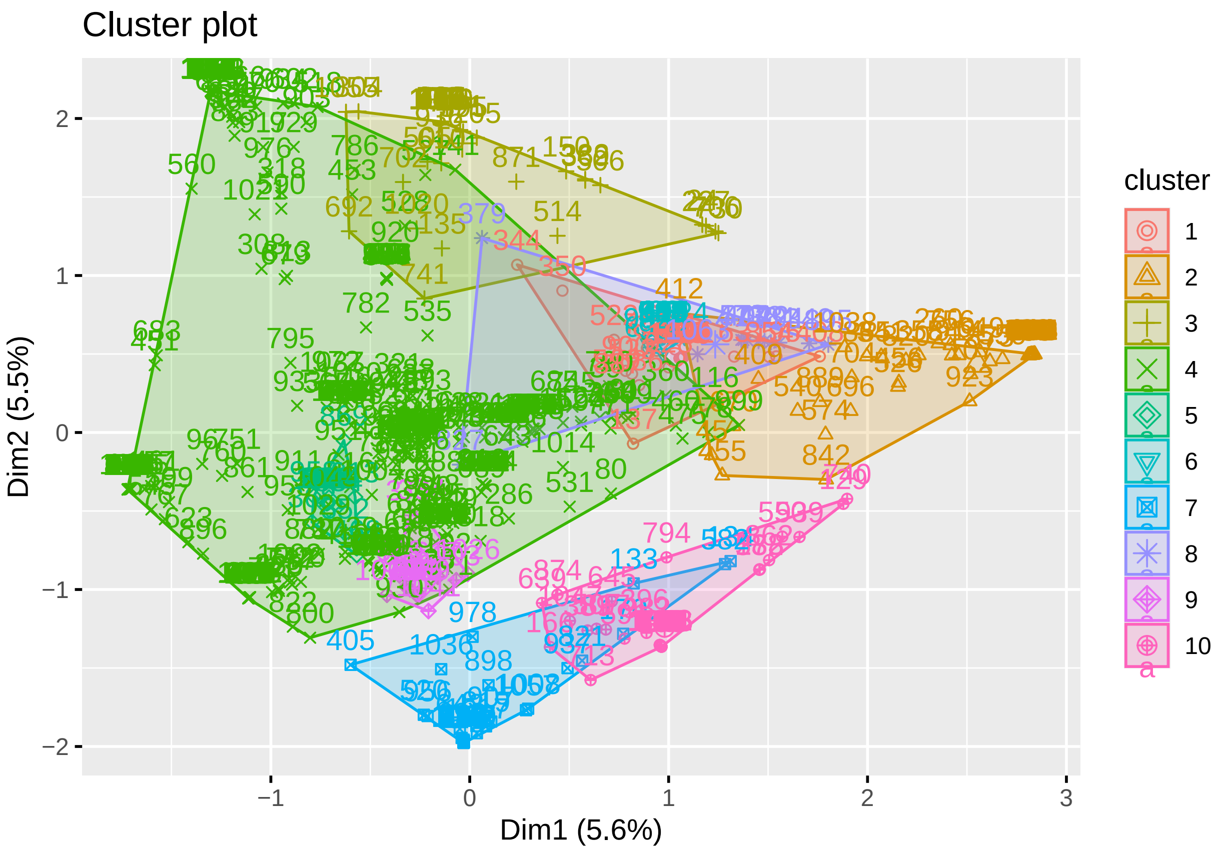
```r
betas <- tidy(lda, matrix = "beta")

gamma_df <- tidy(lda, matrix = "gamma")

gamma_df <- gamma_df |>
  pivot_wider(names_from = topic, values_from = gamma)

cluster <- kmeans(gamma_df |>
  select(-document),10)
```

Take highest gamma for each movie

```r
top_movies_by_topic <- gamma_df |>
  pivot_longer(cols = `1`:`20`, names_to = "topic", values_to = "gamma") |>
  group_by(document) |>
  slice_max(gamma, n = 1) |>
  ungroup() |>
  select(document, topic, gamma)
```

But do get a more detailed look, we need to cluster the movies into 10 clusters by topic

```r
cluster <- kmeans(gamma_df |>  select(-document),10)
fviz_cluster(cluster, data = gamma_df |> select(-document))
```




```
movies <- movies |>
  distinct(Movie.Name, .keep_all = TRUE)

clusters <- cluster[["cluster"]]
cluster$cluster <- clusters
movies$cluster <- clusters
```

Create clusters

```
gamma_df <- gamma_df |>
  left_join(movies |> select(Movie.Name, cluster), by = c("document" = "Movie.Name"))

cluster_1 <- gamma_df |> filter(cluster == 1)

cluster_2 <- gamma_df |> filter(cluster == 2)

cluster_3 <- gamma_df |> filter(cluster == 3)

cluster_4 <- gamma_df |> filter(cluster == 4)

cluster_5 <- gamma_df |> filter(cluster == 5)

cluster_6 <- gamma_df |> filter(cluster == 6)

cluster_7 <- gamma_df |> filter(cluster == 7)
```

```
cluster_8 <- gamma_df |> filter(cluster == 8)

cluster_9 <- gamma_df |> filter(cluster == 9)

cluster_10 <- gamma_df |> filter(cluster == 10)
```

Find which topic is most associated with each cluster

```
col_avg <- function(df) {
  selected_columns <- df |>
    select(`1`:`20`)

  column_averages <- colMeans(selected_columns, na.rm = TRUE)

  return(column_averages)
}

averages_cluster_1 <- col_avg(cluster_1)
print(averages_cluster_1)
```

```
##            1            2            3            4            5            6
## 0.0222333661 0.0527353347 0.0144461738 0.0402533374 0.0213391882 0.0924116503
##            7            8            9           10           11           12
## 0.0554159975 0.1244689656 0.0046185883 0.0385350630 0.0966434180 0.0332871895
##           13           14           15           16           17           18
## 0.0596147530 0.0002856475 0.1095327048 0.0367526515 0.0302752900 0.0575822571
##           19           20
## 0.0765102692 0.0330581547
```

We can see that these probailites are pretty small, however a few of them stick out, particuarily topics 4 and 14. This indicates that cluster 1 is most assiciates with topics 4 and 14.

Make word clouds

```
wordcloud <- function(topic_number) {
  top_words <- betas |>
    filter(topic == topic_number) |>
    top_n(30, beta) |>
    arrange(desc(beta))

  wordcloud::wordcloud(words = top_words$term,
            freq = top_words$beta,
            min.freq = 0,
            scale = c(3, 0.5),
            random.order = FALSE,
            colors = brewer.pal(8, "Dark2"))
}

avg_6 <- col_avg(cluster_6)
print(avg_6)
```

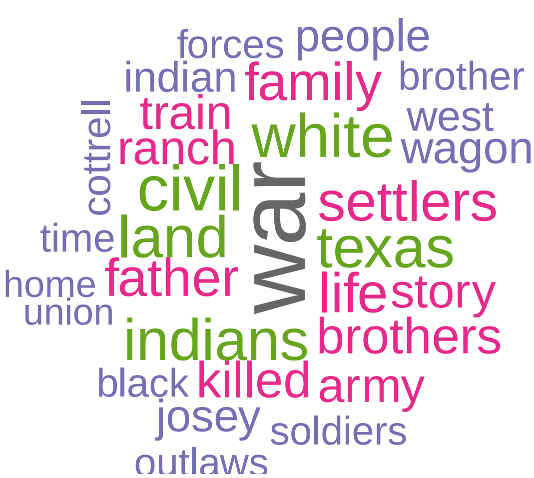
```
##           1           2           3           4           5           6
```

```
## 0.038518157 0.052704407 0.049074038 0.002578612 0.076848261 0.026983826
##           7           8           9          10          11          12
## 0.051280390 0.045101622 0.009226373 0.036245027 0.051364319 0.040780356
##          13          14          15          16          17          18
## 0.025769782 0.094288860 0.033810315 0.025751320 0.092990747 0.057026639
##          19          20
## 0.122000641 0.067656309
```

Lets make some fun word clouds

```r
wordcloud(1)
```


```r
wordcloud(2)
```




```r
wordcloud(3)
```

Jon helped me with the word cloud code