615 Movie

Xiaohan Shi

2024-11-04

#Step 1: data perparing

Read the data

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

Check word frequency first:

```
library(dplyr)
```

```
##
## 载入程序包：'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(janeaustenr)
library(tidytext)
data("stop_words")
movie_words <- movies |> unnest_tokens(word, Plot)
movie_counts <- movie_words %>%
  anti_join(stop_words) %>%
  count(Movie.Name, word, sort = TRUE)
```

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

Weeding out the names, reorganize the data:

```
library(lexicon)
data("freq_first_names")
firstname <- tolower(freq_first_names$Name)
movie_counts <- movie_counts |> filter(!(word %in% firstname))
```

Casting the words counts to a matrix

```
counts_matrix<-movie_counts |> cast_dtm(Movie.Name,word,n)
```

```
#show some info of the text data matrix:
example <- head(counts_matrix, n=6)
print(example)
```

```
## <<DocumentTermMatrix (documents: 6, terms: 13394)>>
## Non-/sparse entries: 638/79726
## Sparsity           : 99%
## Maximal term length: 17
## Weighting          : term frequency (tf)
```

The dimensions of matrix:

```
# View the dimensions of movie counts
dim(movie_counts)
```

```
## [1] 44143     3
```

```
# view the dimensions of movies
dim(movies)
```

```
## [1] 1077    4
```

#Step 2: Use LDA for topic modeling

LDA 30 topics:

```
library(factoextra)
```

```
## 载入需要的程序包：ggplot2
```

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

```
library(topicmodels)
lda<-LDA(counts_matrix, k=30, control= list(seed=1066))
plots_gamma <- tidy(lda, matrix = "gamma")
```

Reorganize data for cluster

```
#Pivoting the plots_gamma table wider in order to cluster by gammas for each topics.
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────── tidy
verse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
## ✓ readr     2.1.5
## ── Conflicts ────────────────────────────────────────────────────────────
──── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
me errors
```

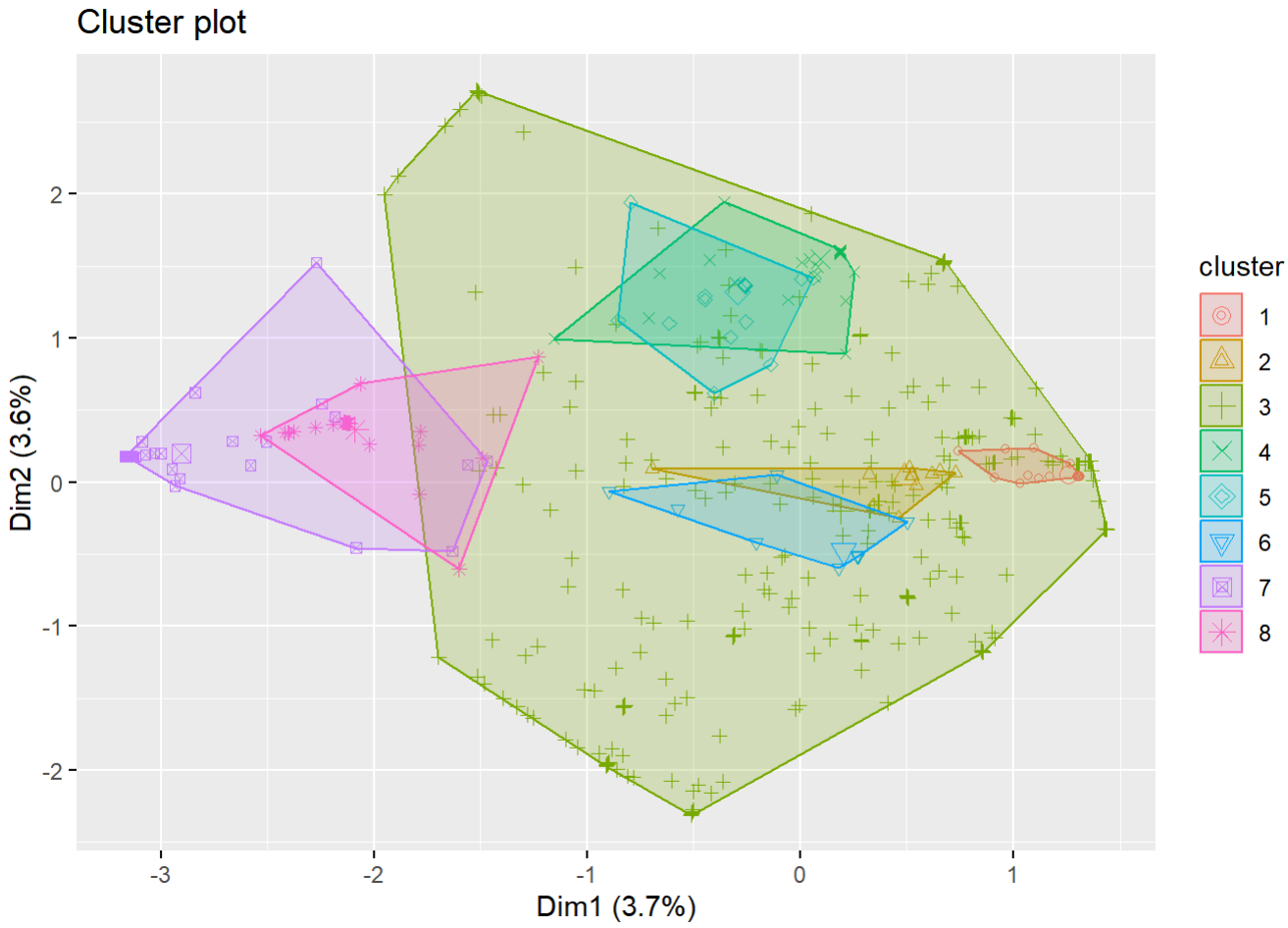
```
plots_gamma_wider<-plots_gamma |> pivot_wider(
  names_from = topic, values_from = gamma
)

#drop NA values
plots_gamma_wider <- plots_gamma_wider |> drop_na()
```

Cluster analysis

```
#Perform K-means clustering
cluster<-kmeans(plots_gamma_wider |> select(-document), centers = 8, nstart = 25)

#visualization
fviz_cluster(cluster, data=plots_gamma_wider |>
               select(-document), geom="point")
```



Summary:

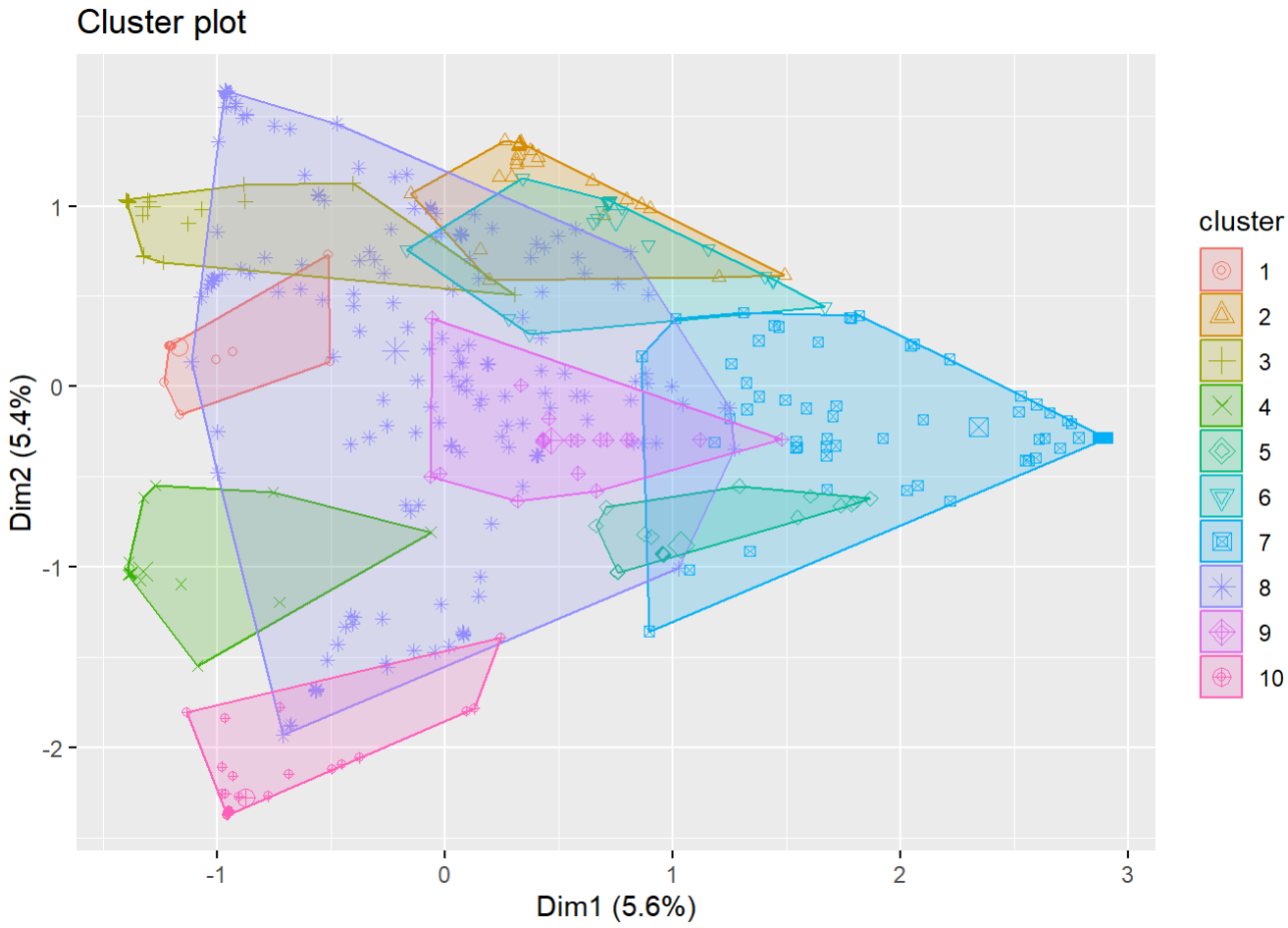
Cluster 4(green) takes up most of the space in the graph, indicating that the themes of the movie are more diverse within that cluster. Other clusters, for example, cluster 2 and 7, take up less space, which indictes that the movies in these clusters shows high similarity.

The boundaries of different clusters overlap, especially in the central region, suggesting that some movies may span multiple themes.

###Different topics and clusters

```
# example: 20 topics & 10 custers
lda_20 <- LDA(counts_matrix, k = 20, control = list(seed = 1066))
plots_gamma_20 <- tidy(lda_20, matrix = "gamma")
plots_gamma_wider_20 <- plots_gamma_20 %>%
  pivot_wider(names_from = topic, values_from = gamma) %>%
  drop_na()

cluster_10 <- kmeans(plots_gamma_wider_20 |> select(-document), centers = 10, nstart = 25)
fviz_cluster(cluster_10, data = plots_gamma_wider_20 |> select(-document), geom = "point")
```



In this new figure, Orange cluster 2 is the most widely distributed and cover multiple regions in the graph, which may indicate that the movie topics are rich in variety.

Comparing to the first figure, Dim1 and Dim2(5.6% and 5.4%) are slightly more higher than figure 1. This means that two-dimensional projections are slightly more interpretive of the data.