Topic modeling

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

```
# Load Data
library(readr)
library(dplyr)
```

```
   'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tm)
```

```
     NLP
```

```r
library(tidytext)

library(textclean)
library(SnowballC)
library(Matrix)

library(topicmodels)

library(ldatuning)
library(ggplot2)
```

```
   'ggplot2'

The following object is masked from 'package:NLP':

    annotate
```

```r
library(wordcloud)
```

```
     RColorBrewer
```

```r
# Load the data
movie_data <- read_csv("C:/Users/Owner/Downloads/movie_plots_with_genres.csv")
```

```
Rows: 1077 Columns: 4

-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): Movie Name, Genre, Plot
dbl (1): row

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(movie_data)
```

```
# A tibble: 6 x 4
    row `Movie Name`                             Genre   Plot
  <dbl> <chr>                                    <chr>   <chr>
1    31 "Pioneers of the West"                   western "Pioneers of the West  ~
2    87 "The Infiltrators"                       action  "The Infiltrators  :  A~
3   146 "\"Graviton: The Ghost Particle\""       sci-fi  "\"Graviton: The Ghost ~
4   197 "Moses: Fallen. In the City of Angels." action  "Moses: Fallen. In the ~
5   314 "The Slave Trade"                        history "The Slave Trade  :  Be~
6   448 "The 303rd"                              history "The 303rd  :  Ret. Col~
```

```r
corpus <- VCorpus(VectorSource(movie_data$Plot))  # Replace 'plot' with actual column name

# Preprocess the text (e.g., remove punctuation, stop words, etc.)
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)

# Create Document Term Matrix
dtm <- DocumentTermMatrix(corpus, control = list(bounds = list(global = c(2, Inf))))

# Convert DTM to a matrix if it has non-zero dimensions
if (ncol(dtm) > 0) {
  dtm_matrix <- as.matrix(dtm)
} else {
  stop("The DTM is empty after preprocessing. Try adjusting preprocessing steps.")
}

# Check if the DTM has content
print(dim(dtm_matrix))
```

```
[1] 1077 6359
```

scree plots for k

Purpose:
A scree plot helps to determine the optimal number of topics k in LDA. It visualizes how well different values of k explain the data, allowing you to choose an appropriate number of topics based on metrics like coherence or perplexity.

Interpretation:

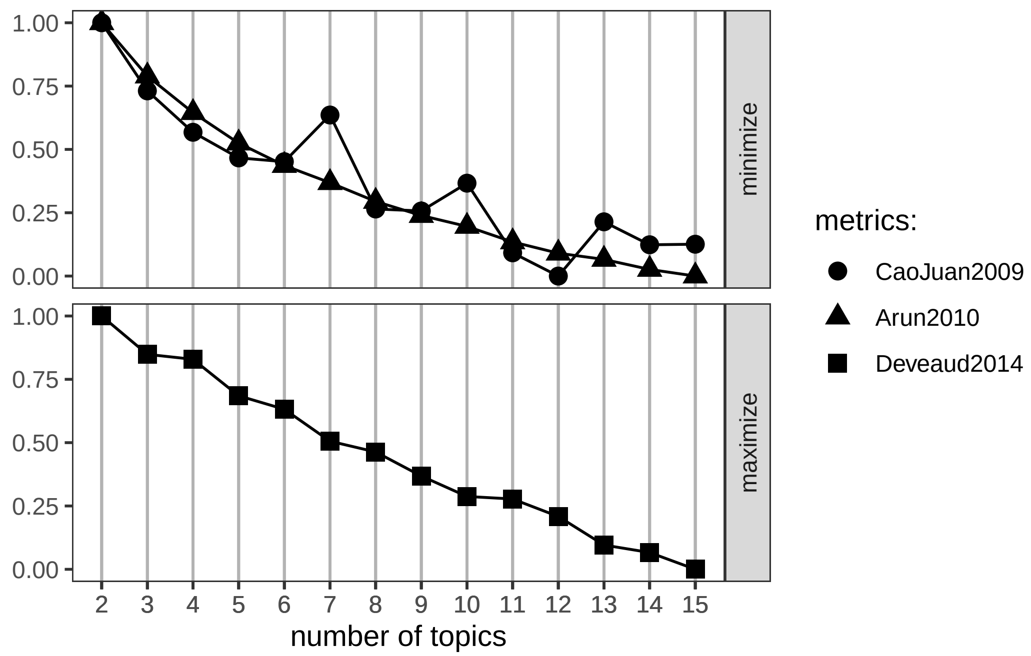
- The "elbow point" or a plateau on the plot often indicates the best value for k, where increasing k further provides diminishing returns

```r
library(Matrix)
dtm_matrix <- as.matrix(dtm)

# Now convert the matrix to a sparse format
dtm_sparse <- Matrix(dtm_matrix, sparse = TRUE)

results <- FindTopicsNumber(
  dtm_sparse,
  topics = seq(from = 2, to = 15, by = 1),
  metrics = c("CaoJuan2009", "Arun2010", "Deveaud2014"),  # Avoid Griffiths2004 if it's probl
  method = "Gibbs",
  control = list(seed = 1234),
  verbose = 5
)
```

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

```r
# Plot the results
FindTopicsNumber_plot(results)
```

```
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>.
```



cluster plot

Purpose:
A cluster plot shows how documents (in this case, movies) group together based on their topic distributions. Each point represents a document, and colors indicate different clusters. After dimensionality reduction (using techniques like PCA or t-SNE), the plot displays relationships between documents in a 2D space.

Interpretation:

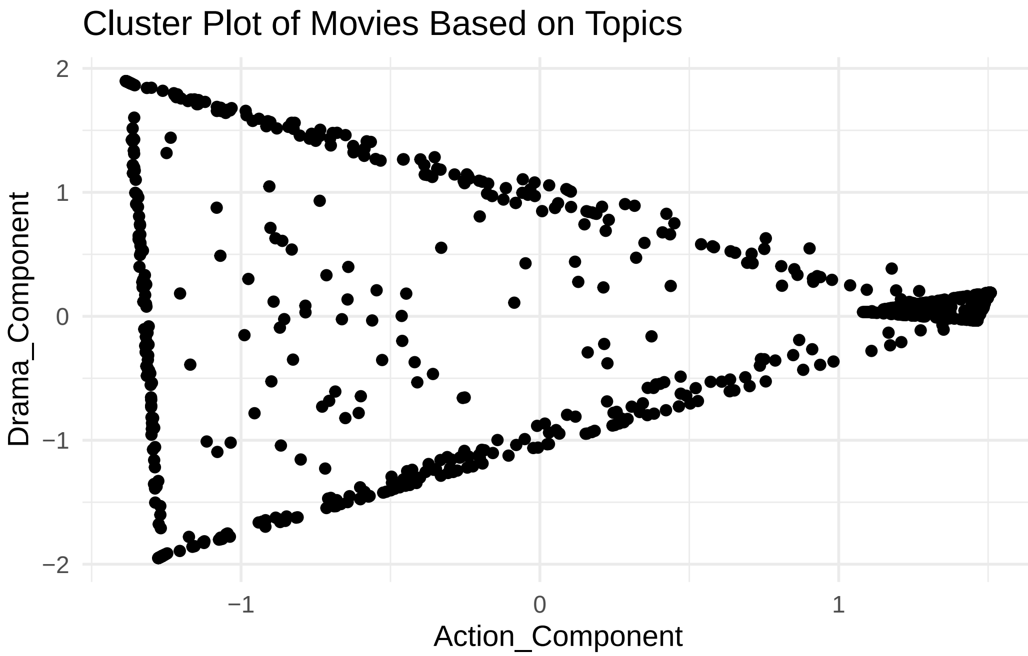
- Documents in the same cluster are close to each other, indicating they have similar topic distributions.
- Points that are far from each other represent documents with distinct topic profiles.

```
k <- 5  # Replace with the optimal number from previous step

# Fit LDA model
lda_model <- LDA(dtm, k = k, control = list(seed = 1234))

# Get the topics and terms
topics <- terms(lda_model, 6)  # 6 terms per topic
print(topics)
```

```
     Topic 1 Topic 2   Topic 3 Topic 4 Topic 5
[1,] "will"  "town"    "world" "world" "bill"
[2,] "life"  "sheriff" "will"  "will"  "gang"
[3,] "man"   "ranch"   "one"   "new"   "john"
[4,] "young" "gang"    "life"  "story" "one"
[5,] "love"  "jim"     "new"   "film"  "jake"
[6,] "one"   "murder"  "time"  "one"   "cattle"
```

```r
# Get gamma matrix
gamma <- posterior(lda_model)$topics

# Perform k-means clustering
set.seed(1234)
kmeans_result <- kmeans(gamma, centers = k)

# Add cluster results to the original data
movie_data$cluster <- as.factor(kmeans_result$cluster)
```

```r
library(ggplot2)
pca <- prcomp(gamma, center = TRUE, scale. = TRUE)

# Extract the first two principal components
gamma_pca <- as.data.frame(pca$x[, 1:2])

# Rename columns based on identified themes
colnames(gamma_pca) <- c("Action_Component", "Drama_Component")  # Replace with appropriate n

# Now `gamma_pca` has descriptive names for the two dimensions
cluster_df <- data.frame(gamma, cluster = kmeans_result$cluster)

ggplot(gamma_pca, aes(x = Action_Component, y = Drama_Component),color= cluster) +
  geom_point() +
  labs(title = "Cluster Plot of Movies Based on Topics") +
  theme_minimal()
```



gamma plots

Purpose:
The gamma plot represents the document-topic distribution. For each document (movie), it shows the proportion (or probability) of each topic being present in that document. This is extracted from the gamma matrix in LDA.

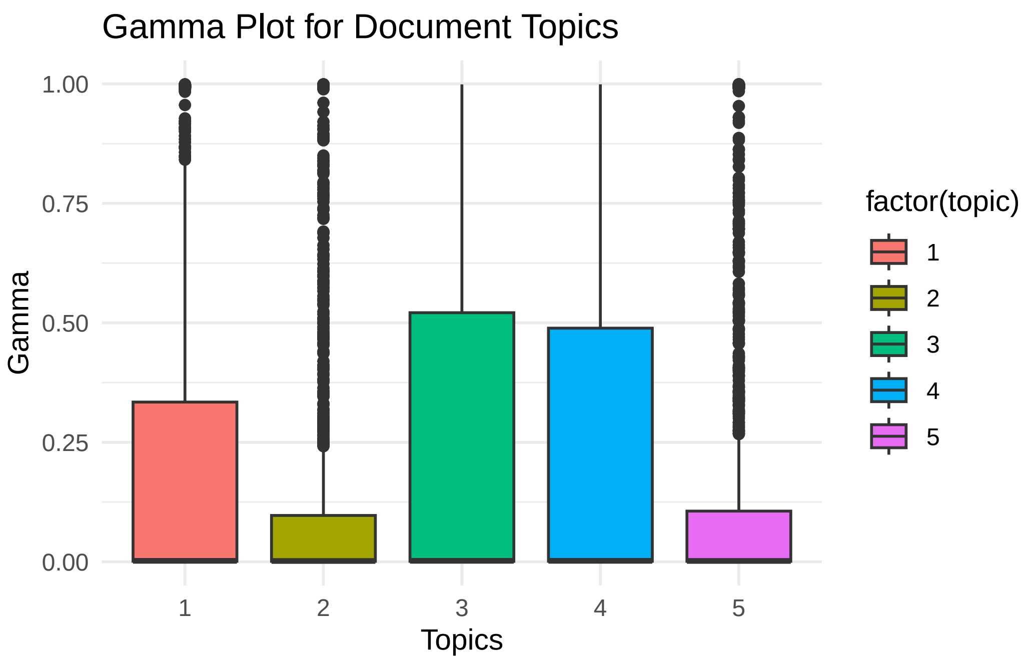
Interpretation:

- A high gamma value for a specific topic indicates that the topic is strongly represented in the document.
- By plotting gamma distributions for documents, you can understand which topics dominate and how evenly distributed topics are across documents.

```
# Convert the gamma matrix (document-topic distribution) to a data frame
gamma_df <- as.data.frame(posterior(lda_model)$topics)

# Add document IDs
gamma_df$document <- 1:nrow(gamma_df)

# Reshape data to long format
library(tidyr)
```

```
    'tidyr'

The following objects are masked from 'package:Matrix':

    expand, pack, unpack
```

```r
gamma_long <- pivot_longer(gamma_df, cols = -document, names_to = "topic", values_to = "gamma

# Plot the gamma values for each topic
ggplot(gamma_long, aes(x = factor(topic), y = gamma, fill = factor(topic))) +
  geom_boxplot() +
  labs(title = "Gamma Plot for Document Topics", x = "Topics", y = "Gamma") +
  theme_minimal()
```



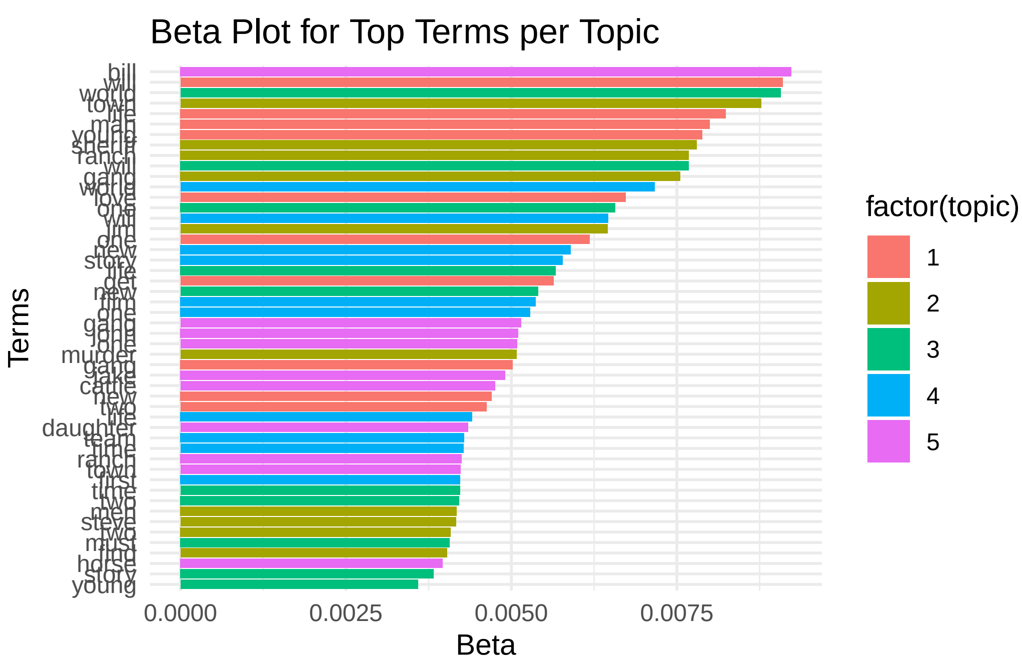
beta plots

Purpose:
The beta plot visualizes the term-topic distribution, showing the probability of each word being associated with a particular topic. This is derived from the beta matrix in LDA.

Interpretation:

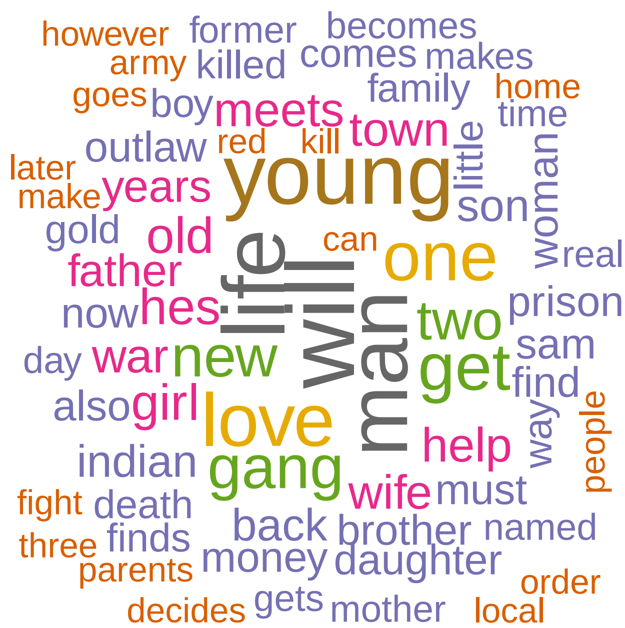
- High beta values for certain terms within a topic indicate that those terms are highly representative of that topic.
- By examining beta values, you can understand the key words that define each topic, which aids in naming and interpreting topics.

```
beta_df <- tidy(lda_model, matrix = "beta")

# Select the top 10 terms per topic based on the beta values
top_terms <- beta_df %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup()

# Plot the beta values for the top terms in each topic
ggplot(top_terms, aes(x = reorder_within(term, beta, topic), y = beta, fill = factor(topic))
  geom_bar(stat = "identity") +
  scale_x_reordered() +
  labs(title = "Beta Plot for Top Terms per Topic", x = "Terms", y = "Beta") +
  coord_flip() +
  theme_minimal()
```



word cloud

```r
# Assuming lda_model is your LDA model object and k is the number of topics
library(tidytext)
topic_words <- tidy(lda_model, matrix = "beta")

# Filter for a specific topic if necessary, e.g., topic 1
topic1_words <- topic_words %>%
  filter(topic == 1) %>%
  arrange(desc(beta)) %>%
  slice_max(order_by = beta, n = 100)  # Top 100 words
library(wordcloud)
library(RColorBrewer)

tryCatch({
  wordcloud(
    words = topic1_words$term,
    freq = topic1_words$beta,
    min.freq = 0.5,
    max.words = 150,
    random.order = FALSE,
    rot.per = 0.2,
    scale = c(3, 0.5),
    colors = brewer.pal(8, "Dark2")
  )
}, warning = function(w) {})
```


however
former
becomes
army
killed
comes
makes
goes
boy
meets
family
home
time
outlaw
red
kill
town
little
later
young
woman
make
years
son
gold
old
can
one
real
father
life
will
man
now
hes
two
prison
day
war
new
get
sam
also
girl
love
find
way
people
indian
gang
help
fight
death
wife
must
three
finds
back
brother
named
parents
money
daughter
order
decides
gets
mother
local

NULL