Module 5 Homework

## Module 5 Homework - Topic Modeling - LDA

### Load the libraries

library(gutenbergr)  
library(topicmodels)  
library(tidyverse)  
library(tidytext)

### Download and prep the data

pride\_raw <- gutenberg\_download(1342)

## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest

## Using mirror http://aleph.gutenberg.org

#We can create a new column chapter that keeps track of which of the   
#chapters each line of text comes from. We can detect in that text column  
#when it says "Chapter".  
pride<-pride\_raw %>%  
 mutate(chapter=ifelse(str\_detect(text,"Chapter"),text,NA))%>%  
 #fill down the new column "chapter" until it gets to a new value   
 fill(chapter)%>%  
 #remove the NA that comes at the end of the last chapter.  
 filter(chapter != "NA") %>%  
 #set the factor levels by the order you find them in the text  
 mutate(chapter = factor(chapter, levels = unique(chapter)))  
#Now the chapters are in order   
#Transform the text data set to the tidy text dataframe, filter out the   
#stop words, and count the most common words.  
tidy\_pride <- pride %>%  
#create a new column to track which line every word is coming from  
 mutate(line = row\_number()) %>%  
#unnest the text column into the word column   
 unnest\_tokens(word, text) %>%  
#filter out the stop words   
anti\_join(stop\_words)

## Joining, by = "word"

#count the words and identify the most common words   
tidy\_pride%>%  
 count(word,sort = TRUE)

## # A tibble: 6,029 x 2  
## word n  
## <chr> <int>  
## 1 elizabeth 597  
## 2 darcy 373  
## 3 bennet 294  
## 4 miss 283  
## 5 jane 263  
## 6 bingley 257  
## 7 time 203  
## 8 lady 183  
## 9 sister 179  
## 10 wickham 162  
## # ... with 6,019 more rows

#notice the number of times we have specific names repeted in our dataset.  
Names <- c("elizabeth", "bennet", "miss", "lady", "dear", "darcy", "jane",   
"bingley", "catherine", "wickham", "kitty", "lydia", "collins", "gardiner",   
"charlotte", "lucas", "lizzy", "colonel", "forster", "hurst", "sir",   
"william", "eliza", "darcy's", "reynaolds", "fitzwilliam")  
#remove the outlier words from the dataset  
tidy\_pride <- pride %>%  
 mutate(line = row\_number()) %>%  
 unnest\_tokens(word, text) %>%  
 anti\_join(stop\_words)%>%  
 filter(!word %in% Names)

## Joining, by = "word"

tidy\_pride

## # A tibble: 33,779 x 4  
## gutenberg\_id chapter line word   
## <int> <fct> <int> <chr>   
## 1 1342 " Chapter 1" 1 chapter  
## 2 1342 " Chapter 1" 1 1   
## 3 1342 " Chapter 2" 3 chapter  
## 4 1342 " Chapter 2" 3 2   
## 5 1342 " Chapter 3" 5 chapter  
## 6 1342 " Chapter 3" 5 3   
## 7 1342 " Chapter 4" 7 chapter  
## 8 1342 " Chapter 4" 7 4   
## 9 1342 " Chapter 5" 9 chapter  
## 10 1342 " Chapter 5" 9 5   
## # ... with 33,769 more rows

## Implement Topic Modeling (LDA)

### 3A Cast the tidied dataset to a dtm:

library(topicmodels)  
tidy\_pride $chapter <-gsub("\\s+","", tidy\_pride$chapter)  
pride\_dtm <-tidy\_pride %>%  
 count(chapter, word) %>%  
 cast\_dtm(chapter, word, n)

topic\_model<-LDA(pride\_dtm, k=5, control = list(seed = 1234))  
topic\_model

## A LDA\_VEM topic model with 5 topics.

### 3B - Display top 15 words of each topic

terms(topic\_model, k=15)

## Topic 1 Topic 2 Topic 3 Topic 4 Topic 5   
## [1,] "time" "ladies" "time" "letter" "time"   
## [2,] "mother" "time" "sister" "family" "sister"   
## [3,] "sister" "sister" "father" "time" "letter"   
## [4,] "day" "day" "aunt" "sister" "family"   
## [5,] "sisters" "friend" "mother" "chapter" "hope"   
## [6,] "family" "chapter" "family" "manner" "feelings"  
## [7,] "happy" "hear" "married" "feelings" "father"   
## [8,] "chapter" "party" "day" "friend" "told"   
## [9,] "replied" "house" "hope" "love" "chapter"   
## [10,] "hope" "cried" "house" "opinion" "day"   
## [11,] "happiness" "evening" "uncle" "day" "manner"   
## [12,] "netherfield" "family" "heard" "agreeable" "friend"   
## [13,] "meryton" "pleasure" "happy" "read" "pleasure"  
## [14,] "bingley’s" "manner" "replied" "received" "honour"   
## [15,] "daughter" "father" "cried" "house" "morning"

### 3C - Interpret the results

I chose k=5 for the number of topics.

Topic 1 - The noticeable words are “sister”, “mother”, “daughter”, “happy”, “hope”. Sounds like the ladies are having a good time.

Topic 2: “party”, “House”, “evening”, “cried”, “pleasure”. Sounds like a typical Saturday night.

Topic 3: “Father” , “marriage”, “married”, “cried”, “uncle”, “hope”. After all of that partying in Topic 2, sounds like wedding bells.

Topic 4: “Letter”, “Love”, “opinion”, “agreeable”, “feelings”, “friend”. The women are sitting around, feeling good about each other.

Topic 5: “pleasure”, “hope”, “honour”, “feelings”, “friend”. This is the upbeat theme. Everything must be beautiful in the morning.

## 4. Word-topic probabilities (The beta matrix):

Let’s start with the tidy verb to calculate Beta. Beta is the probability assignments of words to topics. In other words, Beta is the probablity that a word contributes to a topic.

### 4A Calcluate Beta

#install.packages("reshape2")  
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.0.5

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

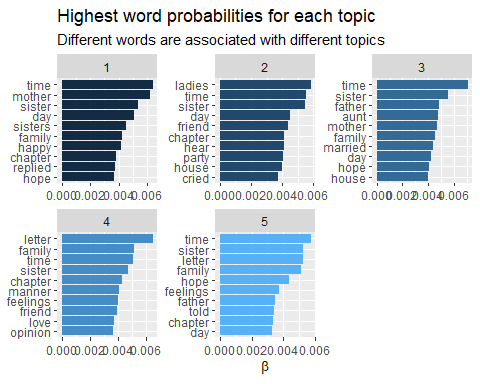
topics <- tidy(topic\_model, matrix = "beta")  
topics

## # A tibble: 30,025 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 \_have\_ 1.55e- 4  
## 2 2 \_have\_ 3.09e-186  
## 3 3 \_have\_ 1.16e-179  
## 4 4 \_have\_ 1.52e- 4  
## 5 5 \_have\_ 1.58e- 4  
## 6 1 \_her\_ 1.95e- 3  
## 7 2 \_her\_ 5.70e- 4  
## 8 3 \_her\_ 3.32e- 6  
## 9 4 \_her\_ 7.93e- 4  
## 10 5 \_her\_ 4.25e- 4  
## # ... with 30,015 more rows

### 4B Calculate and visualize the 10 terms that has the highest probability of belonging to each topic

topics%>%  
 #let's group\_by each topic  
 group\_by(topic)%>%  
 #take the top 10 words in each topic  
 top\_n(10)%>%  
 ungroup%>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(term, beta, fill = topic)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free") +  
 coord\_flip() +  
 labs(x = NULL, y = expression(beta),  
 title = "Highest word probabilities for each topic",  
 subtitle = "Different words are associated with different topics")+  
 scale\_x\_reordered()

## Selecting by beta



### 4C Interpret the results

Topic 1 looks like a bunch of women Just hanging out and chilling. Walking around the house. Good stuff.

Topic 2: There are evening visits that all the women are interested in.

Topic 3: Father and Marriage are involved. Dad gets a shout-out, probably to pay for the wedding, which is probably why “cried” is listed. “letter” is for the invitation.

Topic 4: I don’t see a lot of excitement and mayhem in this topic.

Topic 5: There’s a letter with hope and feelings. Father has been told about it.

## 5. Document-topic probability (The gamma matrix):

LDA also models each document as a mixture of topics. With the matrix = “gamma” argument in tidy(), you can examine the per-document-per-topic probabilities, called γ (“gamma”). gamma is the probablity that a document contributes to a topic.

### 5A Calculate gamma

td\_gamma <- tidy(topic\_model, matrix = "gamma", document\_names = rownames(pride\_dtm))  
td\_gamma

## # A tibble: 305 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 Chapter1 1 1.00   
## 2 Chapter10 1 0.0000296  
## 3 Chapter11 1 0.0000406  
## 4 Chapter12 1 0.797   
## 5 Chapter13 1 0.0000399  
## 6 Chapter14 1 0.0000521  
## 7 Chapter15 1 0.0000351  
## 8 Chapter16 1 0.0000192  
## 9 Chapter17 1 0.0000521  
## 10 Chapter18 1 1.00   
## # ... with 295 more rows

Each of these values is an estimated proportion of words from that document that are generated from that topic.For example, the model estimates that about 99% of the words in ADVENTURE V. were generated from topic 1.

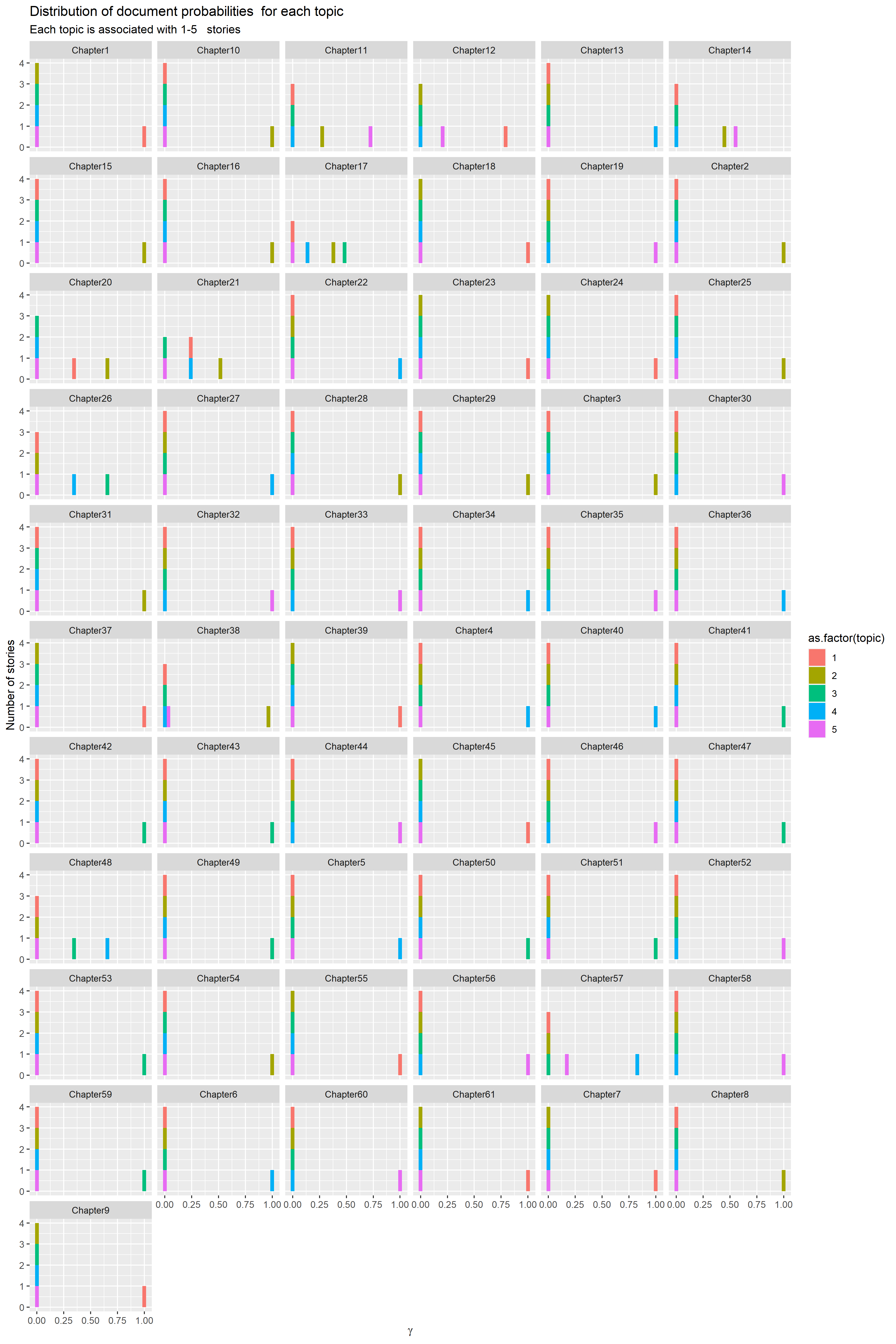
We can plot the results. The y-axis is plotted on a log scale; γ runs from 0 to 1.

### 5B Visualize the results

p1 <- ggplot(td\_gamma, aes(gamma, fill = as.factor(topic))) +  
 #a histogram of gamma  
 #we don't need to see the legend  
 geom\_histogram() +  
 #show the graphs in three columns  
 facet\_wrap(~ document, ncol = 6) +  
 labs(title = "Distribution of document probabilities for each topic",  
 subtitle = "Each topic is associated with 1-5 stories",  
 y = "Number of stories", x = expression(gamma))  
  
ggsave(p1, filename = "chaptertopics.png", height = 18, width = 12)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

knitr::include\_graphics("chaptertopics.png")



### 5C Interpret the results

Chapters 11,12,14,17,20,21,26,38,48, and 57 are veritable roller coasters, careening from topic to topic. Most of the rest of the chapters seem to fit well into one of the defined topics.

## 6 Answer the following questions: (20 points)

### a. How do topic models work? How should we translate topics to concepts

Topic modeling analyzes the text and breaks it up into logical “bags” or topics. The algorithms do the best they can to try a categorize the text based on what it found. This is just a tool that a domain expert can use to interpret the results. Without knowledge of the domain, the findings can be easily misinterpreted.

### b. What do the beta and gamma matrices represent?

Beta matrices show the tentative relationships between words and topics. The value Beta is the probability that a word is a contributor to the topic chosen.

Gamma matrices show the relationships between a chapter/document and a topic. The value Gamma is the probability that a document is a contributor to the topic chosen.

Both beta and Gamma rely on the density of the words or chapters to determine the relationship to a given topic.

### c. What are some of the limitation of topic modeling? How can we overcome the limitations of topic models?

This interpretation is more of an art than a science. Just because a word shows up a lot doesn’t mean that the word is important. One must look at the uniqueness of the word and the context it is used in to determine the significance. If a text is uniform, topic modeling may not be able to find discernable topics. The algorithms are non-deterministic, which can be a double-edged sword. With no pre-conceived notions of what the topics can be, the algorithms are free to discover relationships that might not be obvious, which is why we’re using computers to do the analysis in the first place. On the other hand, sometimes there’s just nothing there. Just because we’re doing fancy analysis doesn’t mean earth-shattering results. To paraphrase, “Boring in, boring out”.