Topic Model Simulation

### Team Members :

#### Anmol More : 11915043

#### Dharani Kiran Kavuri : 11915033

#### Shubhendu Vimal : 11915067

### import libraries in specific order

if (!require(NLP)) {install.packages("NLP")}

## Loading required package: NLP

if (!require(tm)) {install.packages("tm")}

## Loading required package: tm

if (!require(openNLP)) {install.packages("openNLP")}

## Loading required package: openNLP

if (!require(readtext)) {install.packages("readtext")}

## Loading required package: readtext

if (!require(dplyr)) {install.packages("dplyr")}

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

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

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

if (!require(stringr)) {install.packages("stringr")}

## Loading required package: stringr

if (!require(tidytext)) {install.packages("tidytext")}

## Loading required package: tidytext

#order of libraries is important here to avoid masking of functions  
require(NLP)  
require(tm)  
require(openNLP)  
library(readtext)  
library(dplyr)  
library(stringr)  
library(tidytext)

## Data Collection and cleaning

Collect data from wikipedia, through ‘wikipedia’ library in python, code at - Collect Wikipedia Data.ipynb

Ref : <https://stackoverflow.com/questions/18712878/r-break-wiki_content-into-sentences>

Take four different topics from wikipedia files

wiki\_files <- c('Brexit.txt', 'Donald Trump.txt', 'Game of Thrones.txt', 'Bitcoin.txt')  
categories <- c('Brexit', 'Donald Trump', 'Game of Thrones', 'Bitcoin')  
files <- lapply(wiki\_files, readLines)

## Warning in FUN(X[[i]], ...): incomplete final line found on 'Brexit.txt'

## Warning in FUN(X[[i]], ...): incomplete final line found on 'Donald  
## Trump.txt'

## Warning in FUN(X[[i]], ...): incomplete final line found on 'Game of  
## Thrones.txt'

## Warning in FUN(X[[i]], ...): incomplete final line found on 'Bitcoin.txt'

master\_df <- data.frame()  
i <- 1  
for (file in files) {  
 sentences <- get\_sentences\_from\_wiki\_content(file)  
 df <- data.frame(matrix(unlist(sentences), nrow=length(sentences), byrow=T), stringsAsFactors=FALSE)  
 colnames(df) <- 'Content'  
 df['Category'] <- categories[i]  
 master\_df <- rbind(master\_df, df)  
 i <- i+1  
}  
  
#random shuffling of dataframe rows  
df <- master\_df[sample(nrow(master\_df)),]

### Document creation

Number the rows of dataframe, such that each document contains 8 lines

no\_of\_docs <- nrow(df)/8  
df <- df[sample(nrow(df)),]  
doc\_sequences <- rep(1:no\_of\_docs, each=8)  
df$doc\_id <- doc\_sequences

See how random our documents have been assigned across all 216 documents

print(as.data.frame(table(df$Category)))

## Var1 Freq  
## 1 Bitcoin 307  
## 2 Brexit 429  
## 3 Donald Trump 636  
## 4 Game of Thrones 356

### Data cleaning

Remove all stop words, special characters, punctuations etc

Ref :

* <https://stackoverflow.com/questions/48545106/using-tm-package-in-r-to-clean-the-columns-in-dataframe>
* <https://eight2late.wordpress.com/2015/09/29/a-gentle-introduction-to-topic-modeling-using-r/>

if (!require(tm)) {install.packages("tm")}  
library(tm)  
  
wiki\_content <- Corpus(VectorSource(df$Content))  
  
remove\_symbols <- content\_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})  
wiki\_content <- tm\_map(wiki\_content, remove\_symbols, "-")

## Warning in tm\_map.SimpleCorpus(wiki\_content, remove\_symbols, "-"):  
## transformation drops documents

wiki\_content <- tm\_map(wiki\_content, remove\_symbols, "=")

## Warning in tm\_map.SimpleCorpus(wiki\_content, remove\_symbols, "="):  
## transformation drops documents

wiki\_content <- tm\_map(wiki\_content, removeNumbers)

## Warning in tm\_map.SimpleCorpus(wiki\_content, removeNumbers): transformation  
## drops documents

wiki\_content <- tm\_map(wiki\_content, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(wiki\_content, content\_transformer(tolower)):  
## transformation drops documents

wiki\_content <- tm\_map(wiki\_content, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(wiki\_content, removeWords,  
## stopwords("english")): transformation drops documents

wiki\_content <- tm\_map(wiki\_content, removePunctuation)

## Warning in tm\_map.SimpleCorpus(wiki\_content, removePunctuation):  
## transformation drops documents

wiki\_content <- tm\_map(wiki\_content, stripWhitespace)

## Warning in tm\_map.SimpleCorpus(wiki\_content, stripWhitespace):  
## transformation drops documents

custom\_stop\_words <- c("trump")  
wiki\_content <- tm\_map(wiki\_content, removeWords, custom\_stop\_words)

## Warning in tm\_map.SimpleCorpus(wiki\_content, removeWords,  
## custom\_stop\_words): transformation drops documents

clean\_content <- unlist(as.list(wiki\_content))  
df$text\_content <- clean\_content

## Topic Modeling

### Sentence proportion matrix

Create sentence proportion matrix, calculating percentage of each topic in each document

sentence\_prop\_matrix <- data.frame(matrix(ncol = 5, nrow = 0))  
for (i in seq(1:no\_of\_docs)) {  
 df\_current\_doc <- subset(df, df$doc\_id == i)  
 frequency\_count <- as.data.frame(table(df\_current\_doc$Category))  
   
 brexit <- ifelse(length(frequency\_count[frequency\_count$Var1 == "Brexit",]$Freq) > 0,   
 frequency\_count[frequency\_count$Var1 == "Brexit",]$Freq, 0)  
 trump <- ifelse(length(frequency\_count[frequency\_count$Var1 == "Donald Trump",]$Freq) > 0,   
 frequency\_count[frequency\_count$Var1 == "Donald Trump",]$Freq, 0)  
 got <- ifelse(length(frequency\_count[frequency\_count$Var1 == "Game of Thrones",]$Freq) > 0,  
 frequency\_count[frequency\_count$Var1 == "Game of Thrones",]$Freq, 0)  
 bitcoin <- ifelse(length(frequency\_count[frequency\_count$Var1 == "Bitcoin",]$Freq) > 0,  
 frequency\_count[frequency\_count$Var1 == "Bitcoin",]$Freq, 0)  
 sentence\_prop\_matrix <- rbind(sentence\_prop\_matrix,   
 data.frame('Document Id'=i,  
 'Brexit' = brexit/8,   
 'Donald Trump' = trump/8,  
 'Game of Thrones' = got/8,  
 'Bitcoin' = bitcoin/8))  
}  
sentence\_prop\_matrix

## Document.Id Brexit Donald.Trump Game.of.Thrones Bitcoin  
## 1 1 0.750 0.250 0.000 0.000  
## 2 2 0.125 0.375 0.125 0.375  
## 3 3 0.625 0.125 0.125 0.125  
## 4 4 0.125 0.125 0.375 0.375  
## 5 5 0.250 0.500 0.125 0.125  
## 6 6 0.375 0.250 0.375 0.000  
## 7 7 0.125 0.500 0.375 0.000  
## 8 8 0.500 0.125 0.250 0.125  
## 9 9 0.000 0.250 0.375 0.375  
## 10 10 0.125 0.500 0.250 0.125  
## 11 11 0.250 0.375 0.375 0.000  
## 12 12 0.250 0.375 0.125 0.250  
## 13 13 0.375 0.375 0.125 0.125  
## 14 14 0.250 0.375 0.250 0.125  
## 15 15 0.250 0.125 0.250 0.375  
## 16 16 0.000 0.375 0.250 0.375  
## 17 17 0.250 0.375 0.000 0.375  
## 18 18 0.125 0.375 0.125 0.375  
## 19 19 0.250 0.250 0.375 0.125  
## 20 20 0.250 0.375 0.250 0.125  
## 21 21 0.375 0.250 0.125 0.250  
## 22 22 0.375 0.250 0.125 0.250  
## 23 23 0.625 0.125 0.125 0.125  
## 24 24 0.250 0.375 0.125 0.250  
## 25 25 0.000 0.750 0.125 0.125  
## 26 26 0.500 0.250 0.250 0.000  
## 27 27 0.250 0.250 0.500 0.000  
## 28 28 0.125 0.625 0.000 0.250  
## 29 29 0.000 0.250 0.500 0.250  
## 30 30 0.250 0.250 0.250 0.250  
## 31 31 0.125 0.500 0.125 0.250  
## 32 32 0.250 0.375 0.125 0.250  
## 33 33 0.250 0.500 0.125 0.125  
## 34 34 0.250 0.500 0.250 0.000  
## 35 35 0.375 0.625 0.000 0.000  
## 36 36 0.125 0.375 0.375 0.125  
## 37 37 0.000 0.250 0.250 0.500  
## 38 38 0.250 0.500 0.250 0.000  
## 39 39 0.375 0.250 0.000 0.375  
## 40 40 0.375 0.375 0.125 0.125  
## 41 41 0.250 0.500 0.125 0.125  
## 42 42 0.000 0.625 0.125 0.250  
## 43 43 0.375 0.250 0.375 0.000  
## 44 44 0.125 0.625 0.125 0.125  
## 45 45 0.375 0.250 0.125 0.250  
## 46 46 0.125 0.375 0.250 0.250  
## 47 47 0.125 0.500 0.250 0.125  
## 48 48 0.375 0.125 0.375 0.125  
## 49 49 0.500 0.125 0.375 0.000  
## 50 50 0.250 0.250 0.250 0.250  
## 51 51 0.500 0.375 0.125 0.000  
## 52 52 0.250 0.375 0.125 0.250  
## 53 53 0.375 0.500 0.000 0.125  
## 54 54 0.375 0.375 0.000 0.250  
## 55 55 0.250 0.500 0.125 0.125  
## 56 56 0.250 0.500 0.250 0.000  
## 57 57 0.500 0.125 0.250 0.125  
## 58 58 0.375 0.250 0.250 0.125  
## 59 59 0.000 0.250 0.375 0.375  
## 60 60 0.000 0.625 0.250 0.125  
## 61 61 0.500 0.125 0.250 0.125  
## 62 62 0.500 0.250 0.000 0.250  
## 63 63 0.500 0.375 0.000 0.125  
## 64 64 0.125 0.500 0.250 0.125  
## 65 65 0.000 0.125 0.500 0.375  
## 66 66 0.250 0.375 0.125 0.250  
## 67 67 0.375 0.375 0.250 0.000  
## 68 68 0.500 0.000 0.375 0.125  
## 69 69 0.000 0.375 0.375 0.250  
## 70 70 0.500 0.375 0.125 0.000  
## 71 71 0.250 0.125 0.375 0.250  
## 72 72 0.375 0.375 0.250 0.000  
## 73 73 0.375 0.125 0.125 0.375  
## 74 74 0.000 0.625 0.250 0.125  
## 75 75 0.125 0.625 0.125 0.125  
## 76 76 0.125 0.375 0.375 0.125  
## 77 77 0.375 0.500 0.125 0.000  
## 78 78 0.500 0.125 0.375 0.000  
## 79 79 0.375 0.375 0.000 0.250  
## 80 80 0.375 0.375 0.125 0.125  
## 81 81 0.375 0.375 0.125 0.125  
## 82 82 0.125 0.500 0.250 0.125  
## 83 83 0.250 0.125 0.375 0.250  
## 84 84 0.125 0.250 0.250 0.375  
## 85 85 0.125 0.125 0.125 0.625  
## 86 86 0.375 0.250 0.250 0.125  
## 87 87 0.125 0.500 0.125 0.250  
## 88 88 0.000 0.625 0.250 0.125  
## 89 89 0.000 0.250 0.375 0.375  
## 90 90 0.125 0.375 0.375 0.125  
## 91 91 0.250 0.625 0.125 0.000  
## 92 92 0.125 0.500 0.375 0.000  
## 93 93 0.375 0.375 0.000 0.250  
## 94 94 0.000 0.375 0.250 0.375  
## 95 95 0.000 0.125 0.500 0.375  
## 96 96 0.250 0.250 0.125 0.375  
## 97 97 0.250 0.500 0.125 0.125  
## 98 98 0.125 0.125 0.375 0.375  
## 99 99 0.125 0.500 0.250 0.125  
## 100 100 0.250 0.500 0.000 0.250  
## 101 101 0.125 0.375 0.125 0.375  
## 102 102 0.375 0.500 0.000 0.125  
## 103 103 0.375 0.375 0.125 0.125  
## 104 104 0.250 0.500 0.125 0.125  
## 105 105 0.000 0.250 0.125 0.625  
## 106 106 0.125 0.375 0.125 0.375  
## 107 107 0.000 0.125 0.125 0.750  
## 108 108 0.000 0.625 0.250 0.125  
## 109 109 0.125 0.375 0.375 0.125  
## 110 110 0.000 0.375 0.375 0.250  
## 111 111 0.250 0.625 0.125 0.000  
## 112 112 0.375 0.000 0.375 0.250  
## 113 113 0.250 0.500 0.000 0.250  
## 114 114 0.000 0.250 0.375 0.375  
## 115 115 0.250 0.375 0.125 0.250  
## 116 116 0.250 0.500 0.250 0.000  
## 117 117 0.500 0.375 0.125 0.000  
## 118 118 0.250 0.500 0.000 0.250  
## 119 119 0.125 0.250 0.500 0.125  
## 120 120 0.375 0.375 0.125 0.125  
## 121 121 0.250 0.125 0.250 0.375  
## 122 122 0.375 0.125 0.125 0.375  
## 123 123 0.375 0.125 0.250 0.250  
## 124 124 0.250 0.375 0.250 0.125  
## 125 125 0.250 0.250 0.250 0.250  
## 126 126 0.375 0.250 0.125 0.250  
## 127 127 0.000 0.250 0.250 0.500  
## 128 128 0.375 0.500 0.000 0.125  
## 129 129 0.250 0.375 0.250 0.125  
## 130 130 0.125 0.500 0.125 0.250  
## 131 131 0.125 0.625 0.125 0.125  
## 132 132 0.250 0.250 0.375 0.125  
## 133 133 0.125 0.750 0.000 0.125  
## 134 134 0.500 0.375 0.125 0.000  
## 135 135 0.375 0.250 0.250 0.125  
## 136 136 0.375 0.250 0.125 0.250  
## 137 137 0.125 0.250 0.500 0.125  
## 138 138 0.000 0.375 0.625 0.000  
## 139 139 0.500 0.375 0.000 0.125  
## 140 140 0.125 0.625 0.000 0.250  
## 141 141 0.250 0.500 0.250 0.000  
## 142 142 0.250 0.250 0.000 0.500  
## 143 143 0.125 0.500 0.375 0.000  
## 144 144 0.250 0.500 0.125 0.125  
## 145 145 0.250 0.125 0.125 0.500  
## 146 146 0.375 0.375 0.250 0.000  
## 147 147 0.125 0.125 0.375 0.375  
## 148 148 0.250 0.625 0.000 0.125  
## 149 149 0.250 0.375 0.250 0.125  
## 150 150 0.625 0.250 0.125 0.000  
## 151 151 0.250 0.375 0.250 0.125  
## 152 152 0.250 0.375 0.250 0.125  
## 153 153 0.000 0.750 0.250 0.000  
## 154 154 0.125 0.375 0.375 0.125  
## 155 155 0.375 0.375 0.000 0.250  
## 156 156 0.250 0.250 0.250 0.250  
## 157 157 0.375 0.125 0.125 0.375  
## 158 158 0.125 0.625 0.125 0.125  
## 159 159 0.250 0.500 0.125 0.125  
## 160 160 0.250 0.500 0.125 0.125  
## 161 161 0.125 0.500 0.000 0.375  
## 162 162 0.375 0.500 0.125 0.000  
## 163 163 0.375 0.375 0.250 0.000  
## 164 164 0.625 0.125 0.125 0.125  
## 165 165 0.250 0.375 0.125 0.250  
## 166 166 0.125 0.375 0.375 0.125  
## 167 167 0.625 0.125 0.125 0.125  
## 168 168 0.000 0.500 0.250 0.250  
## 169 169 0.125 0.375 0.125 0.375  
## 170 170 0.250 0.375 0.250 0.125  
## 171 171 0.250 0.000 0.250 0.500  
## 172 172 0.125 0.625 0.250 0.000  
## 173 173 0.250 0.500 0.125 0.125  
## 174 174 0.000 0.750 0.250 0.000  
## 175 175 0.125 0.625 0.125 0.125  
## 176 176 0.375 0.250 0.125 0.250  
## 177 177 0.125 0.375 0.375 0.125  
## 178 178 0.125 0.375 0.250 0.250  
## 179 179 0.500 0.250 0.125 0.125  
## 180 180 0.375 0.250 0.250 0.125  
## 181 181 0.375 0.375 0.000 0.250  
## 182 182 0.000 0.375 0.500 0.125  
## 183 183 0.125 0.500 0.125 0.250  
## 184 184 0.375 0.250 0.375 0.000  
## 185 185 0.250 0.250 0.125 0.375  
## 186 186 0.000 0.500 0.375 0.125  
## 187 187 0.375 0.250 0.250 0.125  
## 188 188 0.500 0.125 0.250 0.125  
## 189 189 0.125 0.375 0.125 0.375  
## 190 190 0.250 0.500 0.250 0.000  
## 191 191 0.500 0.250 0.125 0.125  
## 192 192 0.375 0.000 0.375 0.250  
## 193 193 0.500 0.375 0.000 0.125  
## 194 194 0.000 0.500 0.375 0.125  
## 195 195 0.125 0.625 0.125 0.125  
## 196 196 0.500 0.125 0.125 0.250  
## 197 197 0.250 0.375 0.250 0.125  
## 198 198 0.375 0.250 0.250 0.125  
## 199 199 0.000 0.500 0.250 0.250  
## 200 200 0.375 0.250 0.250 0.125  
## 201 201 0.125 0.375 0.500 0.000  
## 202 202 0.250 0.250 0.375 0.125  
## 203 203 0.375 0.375 0.125 0.125  
## 204 204 0.500 0.375 0.125 0.000  
## 205 205 0.250 0.500 0.250 0.000  
## 206 206 0.000 0.750 0.250 0.000  
## 207 207 0.125 0.500 0.250 0.125  
## 208 208 0.250 0.375 0.250 0.125  
## 209 209 0.125 0.875 0.000 0.000  
## 210 210 0.250 0.375 0.125 0.250  
## 211 211 0.250 0.625 0.000 0.125  
## 212 212 0.500 0.125 0.250 0.125  
## 213 213 0.375 0.250 0.250 0.125  
## 214 214 0.000 0.625 0.250 0.125  
## 215 215 0.250 0.625 0.125 0.000  
## 216 216 0.125 0.375 0.125 0.375

### DTM creation

create count of words across each document

df\_by\_words <- df %>%  
 unnest\_tokens(word, text\_content)  
  
df\_word\_count <- df\_by\_words %>%   
 anti\_join(stop\_words) %>%  
 count(doc\_id, word, sort = TRUE) %>%  
 ungroup()

## Joining, by = "word"

documents\_dtm <- df\_word\_count %>%  
 cast\_dtm(doc\_id, word, n)

### Run LDA

library(topicmodels)  
  
system.time({  
 documents\_lda <- LDA(documents\_dtm, k = 4)  
 })

## user system elapsed   
## 1.190 0.003 1.199

documents\_lda

## A LDA\_VEM topic model with 4 topics.

Ref - <https://cran.r-project.org/web/packages/tidytext/vignettes/topic_modeling.html>

Identify topics based on beta parameter to get basic idea

library(tidytext)  
term\_topics <- tidy(documents\_lda, matrix = "beta", control = list(seed = 9876))  
term\_topics

## # A tibble: 26,436 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 outstanding 0.000329   
## 2 2 outstanding 0.00453   
## 3 3 outstanding 0.00308   
## 4 4 outstanding 0.0000430  
## 5 1 brexit 0.00520   
## 6 2 brexit 0.00470   
## 7 3 brexit 0.0104   
## 8 4 brexit 0.00526   
## 9 1 series 0.00608   
## 10 2 series 0.0108   
## # … with 26,426 more rows

str(term\_topics)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 26436 obs. of 3 variables:  
## $ topic: int 1 2 3 4 1 2 3 4 1 2 ...  
## $ term : chr "outstanding" "outstanding" "outstanding" "outstanding" ...  
## $ beta : num 0.000329 0.004528 0.003082 0.000043 0.005196 ...

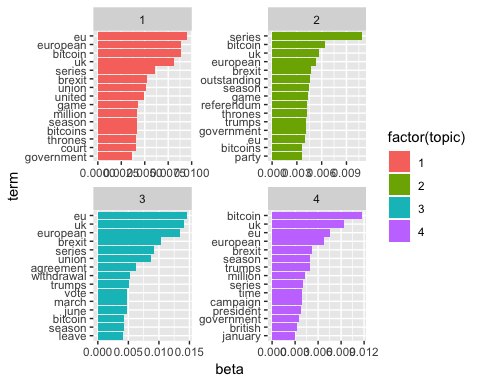
Identify topics based on gamma parameter to get basic idea

top\_terms = term\_topics %>%  
 group\_by(topic) %>%  
 top\_n(15, beta) %>%  
 ungroup() %>%  
 arrange(topic, -beta)  
  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

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

top\_terms %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(term, beta, fill = factor(topic))) +  
 geom\_bar(stat = "identity") +  
 scale\_x\_reordered() +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free") +  
 coord\_flip()



Lot of confusion is there between topic 3 and topic 4

documents\_gamma = tidy(documents\_lda, matrix = "gamma")  
documents\_gamma

## # A tibble: 864 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 11 1 0.0000962  
## 2 139 1 0.0000791  
## 3 105 1 0.000175   
## 4 212 1 0.000121   
## 5 50 1 1.000   
## 6 34 1 0.000142   
## 7 162 1 0.000203   
## 8 167 1 0.000173   
## 9 16 1 0.999   
## 10 18 1 0.000152   
## # … with 854 more rows

Based on above beta and gamma matrix, we see that

* Trump - topic1
* GOT - topic2
* Brexit - topic4
* Bitcoin - topic3

Create a gamma proportion matrix to do a comparision to our original sentence proportion matrix

gamma\_prop\_matrix <- data.frame(matrix(ncol = 5, nrow = 0))  
for (i in seq(1:(nrow(documents\_gamma)/4))) {  
 df\_current\_doc <- subset(documents\_gamma, documents\_gamma$document == i)  
 prop\_topic\_1 <- df\_current\_doc[df\_current\_doc$topic == 1,]$gamma  
 prop\_topic\_2 <- df\_current\_doc[df\_current\_doc$topic == 2,]$gamma  
 prop\_topic\_3 <- df\_current\_doc[df\_current\_doc$topic == 3,]$gamma  
 prop\_topic\_4 <- df\_current\_doc[df\_current\_doc$topic == 4,]$gamma  
   
 gamma\_prop\_matrix <- rbind(gamma\_prop\_matrix,   
 data.frame('Document Id'=i,  
 'Brexit' = prop\_topic\_4,   
 'Donald Trump' = prop\_topic\_1,  
 'Game of Thrones' = prop\_topic\_2,  
 'Bitcoin' = prop\_topic\_3))  
}  
gamma\_prop\_matrix

## Document.Id Brexit Donald.Trump Game.of.Thrones Bitcoin  
## 1 1 9.995032e-01 1.656142e-04 1.656142e-04 1.656142e-04  
## 2 2 2.291250e-04 2.291250e-04 9.993126e-01 2.291250e-04  
## 3 3 2.223920e-04 9.993328e-01 2.223920e-04 2.223920e-04  
## 4 4 2.668229e-04 9.991995e-01 2.668229e-04 2.668229e-04  
## 5 5 9.994995e-01 1.668311e-04 1.668311e-04 1.668311e-04  
## 6 6 2.492478e-04 2.492478e-04 9.992523e-01 2.492478e-04  
## 7 7 3.023563e-04 9.990929e-01 3.023563e-04 3.023563e-04  
## 8 8 9.992761e-01 2.413007e-04 2.413007e-04 2.413007e-04  
## 9 9 9.702614e-01 2.637236e-04 2.921116e-02 2.637236e-04  
## 10 10 9.990419e-01 3.193687e-04 3.193687e-04 3.193687e-04  
## 11 11 9.616716e-05 9.616716e-05 9.997115e-01 9.616716e-05  
## 12 12 2.223920e-04 2.223920e-04 2.223920e-04 9.993328e-01  
## 13 13 1.815016e-04 1.815016e-04 1.815016e-04 9.994555e-01  
## 14 14 2.465412e-04 9.992604e-01 2.465412e-04 2.465412e-04  
## 15 15 2.606955e-04 2.606955e-04 9.992179e-01 2.606955e-04  
## 16 16 1.668311e-04 9.994995e-01 1.668311e-04 1.668311e-04  
## 17 17 9.993456e-01 2.181190e-04 2.181190e-04 2.181190e-04  
## 18 18 9.995431e-01 1.522842e-04 1.522842e-04 1.522842e-04  
## 19 19 1.375254e-04 9.995874e-01 1.375254e-04 1.375254e-04  
## 20 20 3.527353e-01 6.467769e-01 2.438928e-04 2.438928e-04  
## 21 21 9.994030e-01 1.990010e-04 1.990010e-04 1.990010e-04  
## 22 22 2.160434e-04 6.962202e-01 3.033477e-01 2.160434e-04  
## 23 23 2.062314e-04 2.062314e-04 5.285197e-01 4.710679e-01  
## 24 24 1.620676e-04 2.425759e-01 7.571000e-01 1.620676e-04  
## 25 25 1.939023e-04 9.994183e-01 1.939023e-04 1.939023e-04  
## 26 26 2.314608e-04 2.314608e-04 2.314608e-04 9.993056e-01  
## 27 27 1.693194e-04 1.693194e-04 1.693194e-04 9.994920e-01  
## 28 28 2.120087e-04 9.993640e-01 2.120087e-04 2.120087e-04  
## 29 29 9.995172e-01 1.609189e-04 1.609189e-04 1.609189e-04  
## 30 30 1.758776e-04 9.994724e-01 1.758776e-04 1.758776e-04  
## 31 31 1.990010e-04 1.990010e-04 9.994030e-01 1.990010e-04  
## 32 32 2.577361e-04 2.577361e-04 9.992268e-01 2.577361e-04  
## 33 33 9.995032e-01 1.656142e-04 1.656142e-04 1.656142e-04  
## 34 34 1.418206e-04 1.418206e-04 9.995745e-01 1.418206e-04  
## 35 35 1.575686e-04 1.575686e-04 1.575686e-04 9.995273e-01  
## 36 36 2.062314e-04 2.062314e-04 9.993813e-01 2.062314e-04  
## 37 37 9.069248e-01 2.223920e-04 9.263037e-02 2.223920e-04  
## 38 38 1.972719e-04 1.972719e-04 9.994082e-01 1.972719e-04  
## 39 39 2.668229e-04 2.668229e-04 1.007179e-01 8.987484e-01  
## 40 40 2.043750e-04 2.043750e-04 2.043750e-04 9.993869e-01  
## 41 41 7.647430e-01 2.348445e-01 2.062314e-04 2.062314e-04  
## 42 42 2.699960e-04 9.991900e-01 2.699960e-04 2.699960e-04  
## 43 43 1.392119e-04 9.995824e-01 1.392119e-04 1.392119e-04  
## 44 44 9.994232e-01 1.922604e-04 1.922604e-04 1.922604e-04  
## 45 45 1.311692e-04 9.996065e-01 1.311692e-04 1.311692e-04  
## 46 46 2.577361e-04 9.992268e-01 2.577361e-04 2.577361e-04  
## 47 47 2.995782e-01 2.223920e-04 2.223920e-04 6.999770e-01  
## 48 48 1.693194e-04 1.693194e-04 1.693194e-04 9.994920e-01  
## 49 49 2.160434e-04 2.160434e-04 2.160434e-04 9.993519e-01  
## 50 50 1.522842e-04 9.995431e-01 1.522842e-04 1.522842e-04  
## 51 51 1.731943e-04 9.994804e-01 1.731943e-04 1.731943e-04  
## 52 52 1.955726e-04 1.955726e-04 1.955726e-04 9.994133e-01  
## 53 53 6.809397e-01 2.668229e-04 3.185267e-01 2.668229e-04  
## 54 54 9.585372e-01 4.093531e-02 2.637236e-04 2.637236e-04  
## 55 55 2.120087e-04 9.993640e-01 2.120087e-04 2.120087e-04  
## 56 56 2.223920e-04 2.223920e-04 2.223920e-04 9.993328e-01  
## 57 57 2.025517e-04 2.025517e-04 2.025517e-04 9.993923e-01  
## 58 58 1.990010e-04 9.994030e-01 1.990010e-04 1.990010e-04  
## 59 59 2.245919e-04 9.993262e-01 2.245919e-04 2.245919e-04  
## 60 60 1.906460e-04 1.906460e-04 9.994281e-01 1.906460e-04  
## 61 61 2.268358e-04 2.268358e-04 9.993195e-01 2.268358e-04  
## 62 62 2.699960e-04 9.991900e-01 2.699960e-04 2.699960e-04  
## 63 63 2.043750e-04 2.043750e-04 9.993869e-01 2.043750e-04  
## 64 64 1.745256e-04 1.745256e-04 9.994764e-01 1.745256e-04  
## 65 65 3.239252e-04 9.990282e-01 3.239252e-04 3.239252e-04  
## 66 66 2.062314e-04 2.062314e-04 9.993813e-01 2.062314e-04  
## 67 67 1.745256e-04 1.745256e-04 1.745256e-04 9.994764e-01  
## 68 68 1.597864e-04 1.597864e-04 1.597864e-04 9.995206e-01  
## 69 69 1.990010e-04 1.990010e-04 9.994030e-01 1.990010e-04  
## 70 70 9.991703e-01 2.765740e-04 2.765740e-04 2.765740e-04  
## 71 71 2.338448e-04 9.992985e-01 2.338448e-04 2.338448e-04  
## 72 72 1.939023e-04 1.939023e-04 9.994183e-01 1.939023e-04  
## 73 73 2.202348e-04 2.202348e-04 2.202348e-04 9.993393e-01  
## 74 74 2.314608e-04 2.314608e-04 9.993056e-01 2.314608e-04  
## 75 75 9.994641e-01 1.786453e-04 1.786453e-04 1.786453e-04  
## 76 76 1.939023e-04 9.994183e-01 1.939023e-04 1.939023e-04  
## 77 77 1.358793e-04 9.995924e-01 1.358793e-04 1.358793e-04  
## 78 78 2.120087e-04 2.120087e-04 2.120087e-04 9.993640e-01  
## 79 79 1.620676e-04 1.620676e-04 1.620676e-04 9.995138e-01  
## 80 80 2.387631e-04 2.387631e-04 2.387631e-04 9.992837e-01  
## 81 81 2.160434e-04 9.993519e-01 2.160434e-04 2.160434e-04  
## 82 82 2.465412e-04 2.465412e-04 9.992604e-01 2.465412e-04  
## 83 83 2.413007e-04 2.413007e-04 9.992761e-01 2.413007e-04  
## 84 84 3.656609e-04 9.989030e-01 3.656609e-04 3.656609e-04  
## 85 85 6.966894e-02 9.299295e-01 2.007606e-04 2.007606e-04  
## 86 86 7.861931e-01 2.520144e-04 2.520144e-04 2.133029e-01  
## 87 87 2.362784e-04 2.362784e-04 9.992912e-01 2.362784e-04  
## 88 88 1.786453e-04 1.786453e-04 9.994641e-01 1.786453e-04  
## 89 89 2.338448e-04 9.992985e-01 2.338448e-04 2.338448e-04  
## 90 90 1.890584e-04 1.890584e-04 9.994328e-01 1.890584e-04  
## 91 91 2.732454e-04 9.991803e-01 2.732454e-04 2.732454e-04  
## 92 92 9.993328e-01 2.223920e-04 2.223920e-04 2.223920e-04  
## 93 93 2.699960e-04 2.625774e-01 2.699960e-04 7.368826e-01  
## 94 94 9.988851e-01 3.716464e-04 3.716464e-04 3.716464e-04  
## 95 95 1.844507e-04 1.844507e-04 9.994466e-01 1.844507e-04  
## 96 96 9.992837e-01 2.387631e-04 2.387631e-04 2.387631e-04  
## 97 97 2.387631e-04 2.387631e-04 9.992837e-01 2.387631e-04  
## 98 98 2.181190e-04 9.993456e-01 2.181190e-04 2.181190e-04  
## 99 99 1.786453e-04 1.786453e-04 9.994641e-01 1.786453e-04  
## 100 100 3.488078e-04 9.989536e-01 3.488078e-04 3.488078e-04  
## 101 101 9.990807e-01 3.064372e-04 3.064372e-04 3.064372e-04  
## 102 102 1.815016e-04 1.815016e-04 9.302572e-01 6.937976e-02  
## 103 103 1.844507e-04 9.994466e-01 1.844507e-04 1.844507e-04  
## 104 104 9.992985e-01 2.338448e-04 2.338448e-04 2.338448e-04  
## 105 105 9.994764e-01 1.745256e-04 1.745256e-04 1.745256e-04  
## 106 106 1.972719e-04 9.994082e-01 1.972719e-04 1.972719e-04  
## 107 107 9.993393e-01 2.202348e-04 2.202348e-04 2.202348e-04  
## 108 108 1.705917e-04 9.994882e-01 1.705917e-04 1.705917e-04  
## 109 109 1.906460e-04 1.906460e-04 9.994281e-01 1.906460e-04  
## 110 110 9.994030e-01 1.990010e-04 1.990010e-04 1.990010e-04  
## 111 111 2.590063e-01 2.362784e-04 2.362784e-04 7.405211e-01  
## 112 112 1.427121e-04 9.995719e-01 1.427121e-04 1.427121e-04  
## 113 113 2.492478e-04 9.992523e-01 2.492478e-04 2.492478e-04  
## 114 114 2.081218e-04 2.081218e-04 9.993756e-01 2.081218e-04  
## 115 115 1.786453e-04 9.994641e-01 1.786453e-04 1.786453e-04  
## 116 116 3.106298e-04 3.106298e-04 3.106298e-04 9.990681e-01  
## 117 117 9.993923e-01 2.025517e-04 2.025517e-04 2.025517e-04  
## 118 118 9.991995e-01 2.668229e-04 2.668229e-04 2.668229e-04  
## 119 119 1.334827e-04 9.995996e-01 1.334827e-04 1.334827e-04  
## 120 120 3.023563e-04 3.023563e-04 3.023563e-04 9.990929e-01  
## 121 121 9.995551e-01 1.483053e-04 1.483053e-04 1.483053e-04  
## 122 122 8.373878e-01 2.291250e-04 2.291250e-04 1.621539e-01  
## 123 123 9.993456e-01 2.181190e-04 2.181190e-04 2.181190e-04  
## 124 124 2.870647e-04 6.337869e-01 3.656390e-01 2.870647e-04  
## 125 125 9.993813e-01 2.062314e-04 2.062314e-04 2.062314e-04  
## 126 126 2.548431e-04 2.548431e-04 2.548431e-04 9.992355e-01  
## 127 127 2.577361e-04 9.992268e-01 2.577361e-04 2.577361e-04  
## 128 128 9.991995e-01 2.668229e-04 2.668229e-04 2.668229e-04  
## 129 129 2.160434e-04 2.160434e-04 9.993519e-01 2.160434e-04  
## 130 130 2.699960e-04 9.991900e-01 2.699960e-04 2.699960e-04  
## 131 131 2.438928e-04 9.992683e-01 2.438928e-04 2.438928e-04  
## 132 132 1.906460e-04 1.906460e-04 1.906460e-04 9.994281e-01  
## 133 133 2.338448e-04 9.992985e-01 2.338448e-04 2.338448e-04  
## 134 134 9.993056e-01 2.314608e-04 2.314608e-04 2.314608e-04  
## 135 135 1.260707e-04 1.260707e-04 1.260707e-04 9.996218e-01  
## 136 136 1.319314e-04 9.996042e-01 1.319314e-04 1.319314e-04  
## 137 137 2.907407e-04 9.991278e-01 2.907407e-04 2.907407e-04  
## 138 138 1.829642e-04 1.829642e-04 1.829642e-04 9.994511e-01  
## 139 139 7.908362e-05 7.908362e-05 7.908362e-05 9.997627e-01  
## 140 140 1.772507e-04 1.772507e-04 1.772507e-04 9.994682e-01  
## 141 141 9.993640e-01 2.120087e-04 2.120087e-04 2.120087e-04  
## 142 142 2.291250e-04 2.291250e-04 9.348800e-01 6.466171e-02  
## 143 143 1.844507e-04 1.844507e-04 1.844507e-04 9.994466e-01  
## 144 144 9.992683e-01 2.438928e-04 2.438928e-04 2.438928e-04  
## 145 145 2.834805e-04 2.791906e-01 2.834805e-04 7.202424e-01  
## 146 146 2.181190e-04 2.181190e-04 2.181190e-04 9.993456e-01  
## 147 147 2.577361e-04 2.577361e-04 2.577361e-04 9.992268e-01  
## 148 148 2.202348e-04 2.202348e-04 2.202348e-04 9.993393e-01  
## 149 149 1.990010e-04 1.990010e-04 1.990010e-04 9.994030e-01  
## 150 150 7.623225e-01 1.693194e-04 1.693194e-04 2.373388e-01  
## 151 151 1.718831e-04 9.994844e-01 1.718831e-04 1.718831e-04  
## 152 152 2.081218e-04 2.081218e-04 2.081218e-04 9.993756e-01  
## 153 153 9.994920e-01 1.693194e-04 1.693194e-04 1.693194e-04  
## 154 154 2.606955e-04 9.992179e-01 2.606955e-04 2.606955e-04  
## 155 155 6.440146e-01 2.338448e-04 2.338448e-04 3.555177e-01  
## 156 156 3.193687e-04 9.990419e-01 3.193687e-04 3.193687e-04  
## 157 157 9.994183e-01 1.939023e-04 1.939023e-04 1.939023e-04  
## 158 158 1.586697e-04 1.586697e-04 1.586697e-04 9.995240e-01  
## 159 159 1.492804e-04 1.492804e-04 9.995522e-01 1.492804e-04  
## 160 160 1.922604e-04 1.922604e-04 8.509360e-01 1.486795e-01  
## 161 161 9.994183e-01 1.939023e-04 1.939023e-04 1.939023e-04  
## 162 162 9.993923e-01 2.025517e-04 2.025517e-04 2.025517e-04  
## 163 163 1.745256e-04 1.745256e-04 1.745256e-04 9.994764e-01  
## 164 164 1.575686e-04 1.575686e-04 1.575686e-04 9.995273e-01  
## 165 165 2.140070e-04 2.140070e-04 9.993580e-01 2.140070e-04  
## 166 166 2.202348e-04 2.202348e-04 9.993393e-01 2.202348e-04  
## 167 167 1.731943e-04 1.731943e-04 1.731943e-04 9.994804e-01  
## 168 168 9.994232e-01 1.922604e-04 1.922604e-04 1.922604e-04  
## 169 169 9.994183e-01 1.939023e-04 1.939023e-04 1.939023e-04  
## 170 170 9.992440e-01 2.520144e-04 2.520144e-04 2.520144e-04  
## 171 171 2.492478e-04 2.492478e-04 2.492478e-04 9.992523e-01  
## 172 172 2.140070e-04 2.140070e-04 2.140070e-04 9.993580e-01  
## 173 173 1.418206e-04 1.418206e-04 1.418206e-04 9.995745e-01  
## 174 174 2.413007e-04 2.413007e-04 9.992761e-01 2.413007e-04  
## 175 175 9.993869e-01 2.043750e-04 2.043750e-04 2.043750e-04  
## 176 176 9.992088e-01 2.637236e-04 2.637236e-04 2.637236e-04  
## 177 177 9.994232e-01 1.922604e-04 1.922604e-04 1.922604e-04  
## 178 178 9.992088e-01 2.637236e-04 2.637236e-04 2.637236e-04  
## 179 179 1.718831e-04 1.718831e-04 1.718831e-04 9.994844e-01  
## 180 180 2.160434e-04 2.160434e-04 2.160434e-04 9.993519e-01  
## 181 181 9.992268e-01 2.577361e-04 2.577361e-04 2.577361e-04  
## 182 182 1.939023e-04 1.939023e-04 9.994183e-01 1.939023e-04  
## 183 183 9.996398e-01 1.200703e-04 1.200703e-04 1.200703e-04  
## 184 184 1.718831e-04 1.718831e-04 1.718831e-04 9.994844e-01  
## 185 185 2.140070e-04 4.047723e-01 2.140070e-04 5.947997e-01  
## 186 186 9.994804e-01 1.731943e-04 1.731943e-04 1.731943e-04  
## 187 187 2.520144e-04 2.520144e-04 2.520144e-04 9.992440e-01  
## 188 188 1.656142e-04 1.656142e-04 9.995032e-01 1.656142e-04  
## 189 189 2.438928e-04 2.438928e-04 9.992683e-01 2.438928e-04  
## 190 190 1.859614e-04 1.859614e-04 9.994421e-01 1.859614e-04  
## 191 191 9.993328e-01 2.223920e-04 2.223920e-04 2.223920e-04  
## 192 192 9.994030e-01 1.990010e-04 1.990010e-04 1.990010e-04  
## 193 193 1.906460e-04 1.069329e-01 1.906460e-04 8.926858e-01  
## 194 194 1.815016e-04 9.994555e-01 1.815016e-04 1.815016e-04  
## 195 195 1.874971e-04 1.874971e-04 9.994375e-01 1.874971e-04  
## 196 196 3.239252e-04 3.239252e-04 9.990282e-01 3.239252e-04  
## 197 197 2.338448e-04 2.338448e-04 9.992985e-01 2.338448e-04  
## 198 198 1.874971e-04 9.994375e-01 1.874971e-04 1.874971e-04  
## 199 199 2.043750e-04 2.043750e-04 9.993869e-01 2.043750e-04  
## 200 200 1.939023e-04 9.994183e-01 1.939023e-04 1.939023e-04  
## 201 201 1.758776e-04 1.758776e-04 1.758776e-04 9.994724e-01  
## 202 202 3.384097e-04 3.384097e-04 9.989848e-01 3.384097e-04  
## 203 203 9.993056e-01 2.314608e-04 2.314608e-04 2.314608e-04  
## 204 204 9.994764e-01 1.745256e-04 1.745256e-04 1.745256e-04  
## 205 205 1.829642e-04 1.829642e-04 9.994511e-01 1.829642e-04  
## 206 206 2.100473e-04 9.993699e-01 2.100473e-04 2.100473e-04  
## 207 207 1.939023e-04 1.939023e-04 9.994183e-01 1.939023e-04  
## 208 208 2.268358e-04 2.268358e-04 9.993195e-01 2.268358e-04  
## 209 209 9.992761e-01 2.413007e-04 2.413007e-04 2.413007e-04  
## 210 210 2.140070e-04 2.140070e-04 9.993580e-01 2.140070e-04  
## 211 211 2.081218e-04 9.993756e-01 2.081218e-04 2.081218e-04  
## 212 212 7.216863e-01 1.207086e-04 1.207086e-04 2.780723e-01  
## 213 213 1.922604e-04 1.922604e-04 1.922604e-04 9.994232e-01  
## 214 214 3.542502e-04 9.989372e-01 3.542502e-04 3.542502e-04  
## 215 215 9.993393e-01 2.202348e-04 2.202348e-04 2.202348e-04  
## 216 216 1.972719e-04 1.972719e-04 9.994082e-01 1.972719e-04

## Create confusion matrix

Check using confusion matrix accuracy of our topic modeling, steps -

* Indentify highest probability as per gamma matrix for each document
* Indentify highest proportion of sentences in each of our document initially created
* Create a comparision matrix

topics <- c('Brexit', 'Donald.Trump', 'Game.of.Thrones', 'Bitcoin')  
  
gamma\_prop\_matrix$highest\_gamma <- colnames(gamma\_prop\_matrix[topics])[max.col(gamma\_prop\_matrix[topics],ties.method="first")]  
  
sentence\_prop\_matrix$highest\_proportion <- colnames(sentence\_prop\_matrix[topics])[max.col(sentence\_prop\_matrix[topics],ties.method="first")]  
  
confusion\_matrix <- data.frame(matrix(ncol = 4, nrow = 0))  
  
for (topic in topics) {  
 #print(paste0("Identifying accuracy for : ",topic))  
 accurate\_counts <- 0  
 error\_counts <- 0  
 sentence\_prop\_matrix\_subset <- subset(sentence\_prop\_matrix, sentence\_prop\_matrix$highest\_proportion == topic)  
 for (i in (1:nrow(sentence\_prop\_matrix\_subset))) {  
 document\_id <- sentence\_prop\_matrix\_subset[i, ]$Document.Id  
 if(gamma\_prop\_matrix[gamma\_prop\_matrix$Document.Id == document\_id,]$highest\_gamma == topic) {  
 accurate\_counts <- accurate\_counts + 1  
 }  
 else {  
 error\_counts <- error\_counts + 1  
 }  
 }  
 confusion\_matrix <- rbind(confusion\_matrix,   
 data.frame('Topic'=topic,  
 'Accurate' = accurate\_counts,   
 'Error' = error\_counts,  
 'Accuracy Percentage' = accurate\_counts/(accurate\_counts+error\_counts)))  
}  
  
confusion\_matrix

## Topic Accurate Error Accuracy.Percentage  
## 1 Brexit 21 51 0.2916667  
## 2 Donald.Trump 27 83 0.2454545  
## 3 Game.of.Thrones 5 16 0.2380952  
## 4 Bitcoin 3 10 0.2307692

## Conclusion

Accuracy is not that great, however reason behind this is most of our documents have equal mix of all 4 topics. Also, one of the topics “Donald Trump” has its words all over, and Brexit/ Trump/Bitcoin do have some relations