Fundamentals of Computing and Data Display

Term paper template

Tim Raxworthy & Carlos Cristiano

2022-12-05

Contents

Introduction

There are many factors that impact people's decision making processes. Much is unknown as to how people formulate their decisions when choosing support for different conservation organizations that portray themselves to be following a mission statement agreeable to their own beliefs. One major question is, "Why do people select certain organizations over others to donate money to, participate in events, subscribe to newsletters, and read their online articles? Primates other than humans are a particularly important species for measuring concepts normally overlooked as having impact in how people formulate their support for organizations. Many conservation, biological and evolutionary scientists are keen on the acute impact that primates have on human society, their importance and their role in our society's understanding of evolutionary concepts but what is not known is how current human interests surrounding primates could be influencing (directly or indirectly) our own way of designing conservation organizations.

This research aims at gaining some insight into how people view primates on social media. To do this we will be exploring and analyzing twitter data to create different topics related to our search terms (monkey, ape, chimp, primate etc.). This information will then be interpreted within the framework of what aspects of organizations that interact with primate species are "valued" or "normally expected" over others, although we are not comparing different organizations but rather different aspects surrounding a respondents current interest towards primate species. We will also compare if any of the "primate values" outlined in (**Marshall2016?**) overlap with the topics that our LDR model generates.

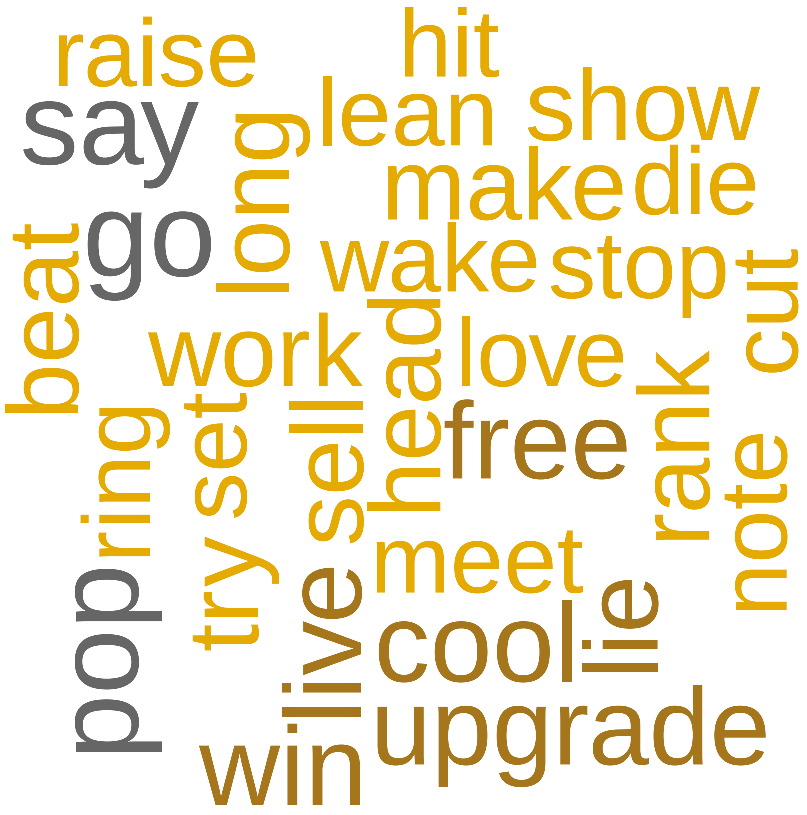
The fate of primate conservation has much do with human intervention whether that be positive or negative. Examining human interpretation of primates will be vital for those designing conservation projects currently and into the future. Primate conservation is valuable to humans for many different reasons, but a significant overarching reason is that primates are more similar to us than other orders of organisms on Earth. This similarity gives insight into our own species that no other animal can. It has been found that primates are excellent model animals for understanding physical and psychological illnesses that ail humans (**Estrada2017?**). Primates also possess similar cognitive abilities to humans and some captive chimpanzees have displayed a working memory that rivals that of humans (Inoue, 2007). Chimpanzees use of tools could imply that they have an understanding of causation and posses exceptional problem solving skills (Whiter, 2011). These features demonstrate some of the similarities that other primate species share with humans. This study is an exploration into what kinds of public support for primate conservation are being discussed on internet forums such as twitter. To determine this, we are building a topic model that can help distinguish and categorize these different discussions, quantifying which ones are happening at the highest frequency across tweets.

Data

This section describes the data sources and the data gathering process.

```r
Data.science <- search_tweets(
  q = "monkey", # search for Tweets with "data" AND "science",
  n = 4000
)
data = Data.science %>%
  select(full_text) %>%
  mutate(doc_id=seq(n())) %>%
  data.frame()

corpus_sotu_orig <- corpus(data,
                           docid_field = "doc_id",
                           text_field = "full_text")

corpus_sotu_proc <- tokens(corpus_sotu_orig,
                           remove_punct = TRUE, # remove punctuation
                           remove_numbers = TRUE, # remove numbers
                           remove_symbols = TRUE) %>%
  tokens_tolower()
lemmaData <- read.csv2("baseform_en.tsv",
                       sep="\t",
                       header=FALSE,
                       encoding = "UTF-8",
                       stringsAsFactors = F)

lemmaData = lemmaData %>%
  filter(!is.na(V1))
cloud =lemmaData %>%
  group_by(V2) %>%
  mutate(freq=n()) %>%
  distinct(freq,V2) %>%
  filter(freq<40) %>%
  arrange(desc(freq))

cloud = cloud[1:100,]

wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1,
          max.words=200, random.order=TRUE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : close
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : post
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : lead
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, :
## network could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : base
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : hope
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : send
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : double
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : drive
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : dream
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : smell
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : hack
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : burn
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : wind
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : run
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : aim
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : call
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : miss
## could not be fit on page. It will not be plotted.
## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : look
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : travel
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : flash
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : pick
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : wish
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : sound
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : lay
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : star
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : round
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : walk
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : need
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : dry
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : start
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : label
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : light
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : focus
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, :
## benefit could not be fit on page. It will not be plotted.
## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : score
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : fit
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : back
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : total
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, :
## channel could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : fire
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : think
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : level
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : hate
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : pat
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : delay
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : shine
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : fine
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : mean
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : well
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : face
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : cancel
## could not be fit on page. It will not be plotted.
## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : help
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : turn
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : model
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : change
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : save
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : fuel
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : bear
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : refuse
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : use
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : open
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : do
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : low
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : be
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : update
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : praise
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : move
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : top
## could not be fit on page. It will not be plotted.
## Warning in wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1, : pass
## could not be fit on page. It will not be plotted.
```



Results

This section presents the main results.

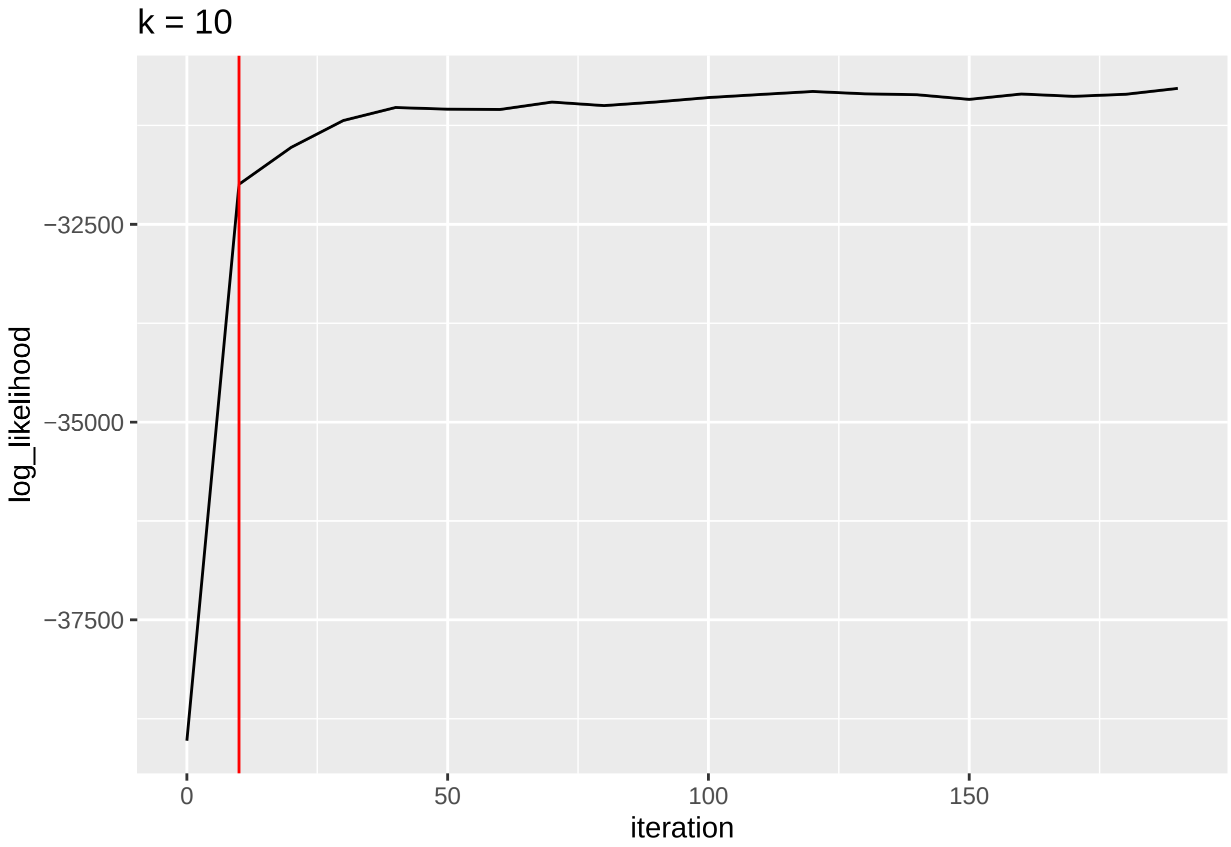
Data exploration

The results section may have a data exploration part, but in general the structure here depends on the specific project.

```
corpus_sotu_proc <-  tokens_replace(corpus_sotu_proc,
                                    lemmaData$V1,
                                    lemmaData$V2,
                                    valuetype = "fixed")

corpus_sotu_proc <- corpus_sotu_proc %>%
  tokens_remove(stopwords("english")) %>%
  tokens_ngrams(1)
DTM <- dfm(corpus_sotu_proc)
minimumFrequency <- 10
DTM <- dfm_trim(DTM,
                min_docfreq = minimumFrequency,
                max_docfreq = 100)
DTM  <- dfm_select(DTM,
                   pattern = "[a-z]",
                   valuetype = "regex",
                   selection = 'keep')
colnames(DTM) <- stringi::stri_replace_all_regex(colnames(DTM),
                                                 "[^_a-z]","")

DTM <- dfm_compress(DTM, "features")
sel_idx <- rowSums(DTM) > 0
DTM <- DTM[sel_idx, ]
textdata <- data[sel_idx, ]

model <- FitLdaModel(dtm = DTM,
                     k = 20,
                     iterations = 200, # I usually recommend at least 500 iterations or more
                     burnin = 180,
                     alpha = 0.1,
                     beta = 0.05,
                     optimize_alpha = TRUE,
                     calc_likelihood = TRUE,
                     calc_coherence = TRUE,
                     calc_r2 = TRUE,
                     cpus = 2)
```

```
## dtm is not of class dgCMatrix, attempting to convert...
```

```
model2=as.data.frame(model$log_likelihood)
ggplot(model2,aes(x=iteration,y=log_likelihood))+
  geom_line()+
  geom_vline(xintercept = 10, col="red")+
  labs(title = "k = 10")
```



```
K <- 10

topicModel <- LDA(DTM,
                  K,
                  method="Gibbs",
                  control=list(iter = 500,
                               verbose = 25))
```

```
## K = 10; V = 251; M = 1410
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

```
tmResult <- modeltools::posterior(topicModel)
beta <- tmResult$terms

theta <- tmResult$topics

#terms(topicModel, 10)
top5termsPerTopic <- terms(topicModel,
                           5)
# For the next steps, we want to give the topics more descriptive names
#than just numbers. Therefore, we simply concatenate the five most likely
#terms of each topic to a string that represents a pseudo-name for each topic.
topicNames <- apply(top5termsPerTopic,
                    2,
                    paste,
                    collapse=" ")
topicProportions <- colSums(theta) / nrow(DTM)  # average probability over all paragraphs
names(topicProportions) <- topicNames     # Topic Names
sort(topicProportions, decreasing = TRUE)
```

```
##           play important dusky leaf seed
##                               0.10079933
##                 good year come much feel
##                               0.10058433
##                   just can see think day
##                               0.10009262
##              go thread people new follow
##                               0.10001618
##                  say never call make pox
##                               0.09995509
##     man trump insecure secretly tribelaw
##                               0.09990503
## maymayentrata u bad even monkey__present
##                               0.09977479
##                   get one love kid watch
##                               0.09977272
##                jenna ortega tone back en
##                               0.09970699
##                    d amp now luffy human
##                               0.09939292
```

```
attr(topicModel, "alpha")
```

```
## [1] 5
```

```
topicModel2 <- LDA(DTM,
                   K,
                   method="Gibbs",
                   control=list(iter = 500,
                                verbose = 25,
                                alpha = 0.2))#replace alpha
```

```
## K = 10; V = 251; M = 1410
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

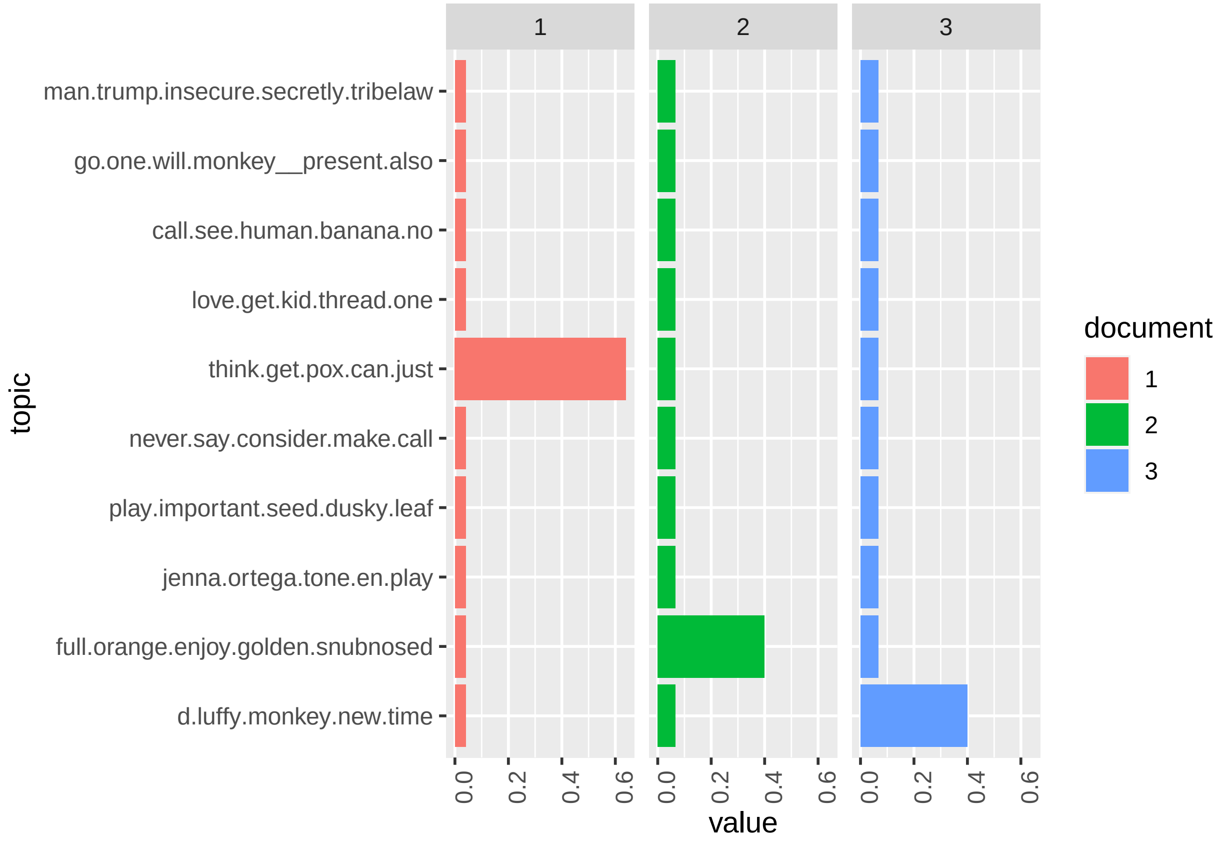
```
tmResult <- modeltools::posterior(topicModel2)
theta <- tmResult$topics
beta <- tmResult$terms

topicProportions <- colSums(theta) / nrow(DTM)  # average probability over all paragraphs
names(topicProportions) <- topicNames     # Topic Names
sort(topicProportions, decreasing = TRUE)
```

```
##           play important dusky leaf seed
##                               0.13844890
##                 good year come much feel
##                               0.12314043
##                  say never call make pox
##                               0.10926687
##              go thread people new follow
##                               0.09824203
##                   get one love kid watch
##                               0.09542866
##                   just can see think day
##                               0.09092658
##     man trump insecure secretly tribelaw
##                               0.09006988
## maymayentrata u bad even monkey__present
##                                0.08924591
##                jenna ortega tone back en
##                                0.08575856
##                    d amp now luffy human
##                                0.07947219
```

```
topicNames <- apply(terms(topicModel2, 5), 2, paste, collapse = " ")
exampleIds <- c(2, 100, 200)
N <- length(exampleIds)

topicProportionExamples <- as.tibble(theta) %>%
  slice(exampleIds)
```

```
## Warning: ‘as.tibble()‘ was deprecated in tibble 2.0.0.
## i Please use ‘as_tibble()‘ instead.
## i The signature and semantics have changed, see ‘?as_tibble‘.
```

```
colnames(topicProportionExamples) <- topicNames

vizDataFrame <- melt(cbind(data.frame(topicProportionExamples),
                           document = factor(1:N)),
                     variable.name = "topic",
                     id.vars = "document")

ggplot(data = vizDataFrame,
       aes(topic, value,
           fill = document),
       ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90,
                                   hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document,
             ncol = N)
```



```
# What happens here depends on the specific project
```

Analysis

This section presents the main results, such as (for example) stats and graphs that show relationships, model results and/or clustering, PCA, etc.

```
# What happens here depends on the specific project
```

```
# What happens here depends on the specific project
```

```
# What happens here depends on the specific project
```

Discussion

This section summarizes the results and may briefly outline advantages and limitations of the work presented.

References