Topic 6: Topic Analysis

Juliet Cohen

2022-05-09

```
library(here)
```

```
## here() starts at /Users/juliet/Documents/MEDS/Text_Analysis/Text_Analysis
```

```
library(pdftools)
```

```
## Using poppler version 22.02.0
```

```
library(quanteda)
```

```
## Package version: 3.2.1
## Unicode version: 13.0
## ICU version: 69.1
```

```
## Parallel computing: 8 of 8 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following objects are masked from 'package:quanteda':
##
##     meta, meta<-
```

```
##
## Attaching package: 'tm'
```

```
## The following object is masked from 'package:quanteda':
##
##     stopwords
```

```
library(topicmodels)
library(ldatuning)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::filter()     masks stats::filter()
```

```
## x dplyr::lag()        masks stats::lag()
```

```
library(tidytext)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

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

Load the data

```
##Topic 6 .Rmd here:https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/main/topic_6.Rmd
#grab data here:
comments_df<-read_csv("https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/main/dat/commen
```

```
## Rows: 81 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Document, text
##
## 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.
```

```
#comments_df <- read_csv(here("dat", "comments_df.csv")) #if reading from local
```

Now we'll build and clean the corpus

```
epa_corp <- corpus(x = comments_df, text_field = "text")
```

```
## Warning: NA is replaced by empty string
```

```
epa_corp.stats <- summary(epa_corp)
head(epa_corp.stats, n = 25)
```

```
##      Text Types Tokens Sentences
## 1   text1  1196   3973       178
## 2   text2   830   2509       111
## 3   text3   279    571        31
## 4   text4  1745   6904       251
## 5   text5   581   1534        49
## 6   text6   469   1187        53
## 7   text7   424    903        38
## 8   text8  3622  22270       655
## 9   text9   373    717        25
## 10 text10   404    971        42
## 11 text11   710   2190        77
## 12 text12   636   1896        82
## 13 text13   146    206         3
## 14 text14  1124   3197        86
## 15 text15   914   2943        90
## 16 text16    13     45         1
## 17 text17  1043   3190       103
## 18 text18   313    601        24
## 19 text19   152    229         6
## 20 text20   341    786        35
## 21 text21   211    403        15
```

```
## 22 text22   186    322        12
## 23 text23   211    398        14
## 24 text24   325    696        33
## 25 text25  1749   5382       115
##                                                 Document
## 1                                      1_Air Alliance.pdf
## 2                                          10_Bus NEJ.pdf
## 3                                    11_Carlton Ginny.pdf
## 4                                     15_City Project.pdf
## 5                                    16_Corporate EEC.pdf
## 6                                  17_Detriot Sierra Club.pdf
## 7                                     18_District DOE.pdf
## 8                                    19_Earth Justice.pdf
## 9                                        2_Alex Kidd.pdf
## 10                                  20_Elizabeth Mooney.pdf
## 11                                         21_Env COS.pdf
## 12                                     22_Env Def Fund.pdf
## 13                                   23_Env Health Watch.pdf
## 14 24_Env Justice Leadership Forum on Climate Change.pdf
## 15                                    25_Env Law at Duke.pdf
## 16                                     26_Farm worker AF.pdf
## 17                                  27_Farm Worker Justice.pdf
## 18                                       28_Faulker County.pdf
## 19                                     29_First Peoples.pdf
## 20                                   3_Alliance for Metro.pdf
## 21                                         30_Gage Blasi.pdf
## 22                                          31_Gull Leon.pdf
## 23                                       32_Hilary Kramer.pdf
## 24                                    33_Housing Land Advoc.pdf
## 25                                         34_Human rights.pdf
```

```
# create tokens obj, remove punct and numeral and stop words
toks <- tokens(epa_corp, remove_punct = TRUE, remove_numbers = TRUE)

# I added some project-specific stop words here
add_stops <- c(stopwords("en"),"environmental", "justice", "ej", "epa", "public", "comment")
toks1 <- tokens_select(toks, pattern = add_stops, selection = "remove")
```

And now convert to a document-feature matrix

```
# convert to document feature matrix
dfm_comm<- dfm(toks1, tolower = TRUE)
# reduce words to base word
dfm <- dfm_wordstem(dfm_comm)
# remove terms only appearing in one doc (min_termfreq = 10)
dfm <- dfm_trim(dfm, min_docfreq = 2)

print(head(dfm)) # each comment is a row, each col is a term
```

```
## Document-feature matrix of: 6 documents, 2,781 features (82.75% sparse) and 1 docvar.
##        features
## docs    charl lee deputi associ assist administr usepa offic 2201-a
##   text1     1   2      1      1      6         6     1     7      1
##   text2     1   1      1      4      3         1     0     5      0
##   text3     0   0      0      0      1         0     0     2      0
##   text4     0   0      0      0      1         9     0     1      0
```

```
##   text5     4   5      1      1      1         1     0     1      1
##   text6     1   1      1      3      1         3     0     4      0
##        features
## docs    pennsylvania
##   text1            1
##   text2            0
##   text3            0
##   text4            0
##   text5            1
##   text6            0
## [ reached max_nfeat ... 2,771 more features ]
```

```
# remove rows (docs) with all zeros (these 0's are present bc we removed stop words)
sel_idx <- slam::row_sums(dfm) > 0
dfm <- dfm[sel_idx, ]
#comments_df <- dfm[sel_idx, ]
```

We somehow have to come up with a value for k,the number of latent topics present in the data. How do we do this? There are multiple methods. Let's use what we already know about the data to inform a prediction. The EPA has 9 priority areas: Rulemaking, Permitting, Compliance and Enforcement, Science, States and Local Governments, Federal Agencies, Community-based Work, Tribes and Indigenous People, National Measures. Maybe the comments correspond to those areas?

```
k <- 9

# feed in the DFM and the num of topics to look for and number of iterations, this function estimates t
topicModel_k9 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 9; V = 2781; M = 77
## 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!
```

```
#nTerms(dfm_comm)

tmResult <- posterior(topicModel_k9)
#tmResult
```

```
attributes(tmResult)
```

```
## $names
## [1] "terms"  "topics"
```

```
#ncol(tmResult) # does not run

#nTerms(dfm_comm)
beta <- tmResult$terms   # get beta from results
dim(beta)                # K distributions over nTerms(DTM) terms# lengthOfVocab
```

```
## [1]    9 2781
```

```
terms(topicModel_k9, 10)
```

```
##       Topic 1 Topic 2   Topic 3     Topic 4     Topic 5     Topic 6
##  [1,] "agenc" "program" "communiti" "communiti" "health"    "permit"
##  [2,] "right" "state"   "water"     "enforc"    "peopl"     "state"
##  [3,] "issu"  "includ"  "plan"      "action"    "citi"      "consid"
##  [4,] "plan"  "feder"   "local"     "monitor"   "park"      "air"
##  [5,] "titl"  "polici"  "work"      "pollut"    "communiti" "comment"
##  [6,] "vi"    "regul"   "use"       "comment"   "see"       "opportun"
##  [7,] "civil" "epa"     "govern"    "air"       "execut"    "feder"
##  [8,] "work"  "requir"  "make"      "complianc" "law"       "organ"
##  [9,] "mani"  "may"     "comment"   "provid"    "green"     "like"
## [10,] "act"   "effect"  "strategi"  "requir"    "can"       "grant"
##       Topic 7    Topic 8     Topic 9
##  [1,] "prison"   "pollut"    "framework"
##  [2,] "site"     "communiti" "communiti"
##  [3,] "sourc"    "impact"    "draft"
##  [4,] "project"  "state"     "develop"
##  [5,] "center"   "rule"      "effort"
##  [6,] "popul"    "health"    "action"
##  [7,] "facil"    "air"       "comment"
##  [8,] "industri" "also"      "agenda"
##  [9,] "water"    "popul"     "epa"
## [10,] "mercuri"  "ejscreen"  "agenc"
```

Some of those topics seem related to the cross-cutting and additional topics identified in the EPA's response to the public comments:

1. Title VI of the Civil Rights Act of 1964

2.EJSCREEN

3. climate change, climate adaptation and promoting greenhouse gas reductions co-benefits
4. overburdened communities and other stakeholders to meaningfully, effectively, and transparently participate in aspects of EJ 2020, as well as other agency processes
5. utilize multiple Federal Advisory Committees to better obtain outside environmental justice perspectives
6. environmental justice and area-specific training to EPA staff
7. air quality issues in overburdened communities

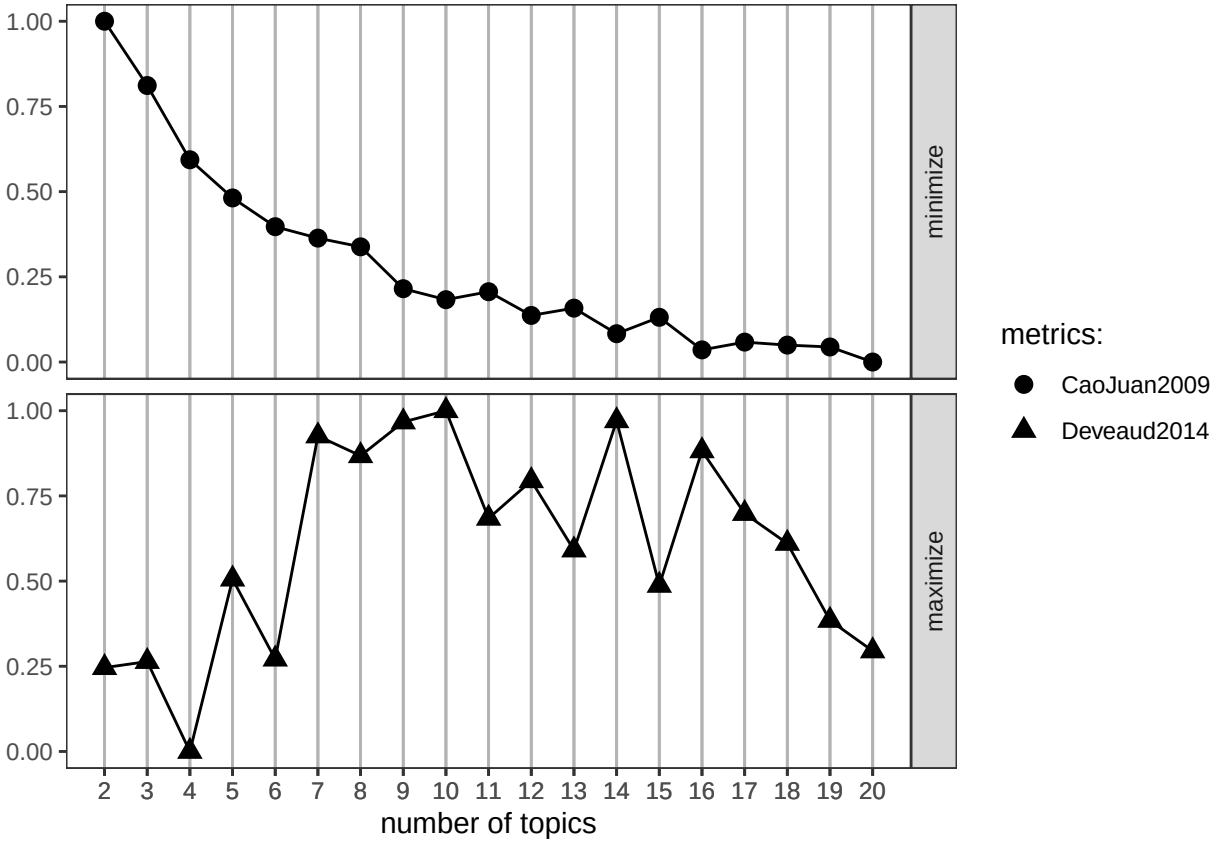
So we could guess that there might be a 16 topics (9 priority + 7 additional). Or we could calculate some metrics from the data. (what initial value of k gives us the best model)

```
# fit the model by running a series of models, starting with 2 topics and ranging to 20 topics
result <- FindTopicsNumber(
  dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)
```

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

```
FindTopicsNumber_plot(result)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```




```
# interpretation:
# top line: the lower the y-axis number the better, so the more topics you add, the better
# bottom line: the higher the number the better, so 7 and 14 looks good
# y-axis has no units

k <- 7

topicModel_k7 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 7; V = 2781; M = 77
## 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 <- posterior(topicModel_k7)
terms(topicModel_k7, 10)
```

```
##       Topic 1  Topic 2     Topic 3      Topic 4     Topic 5  Topic 6
##  [1,] "state"  "framework" "state"      "communiti" "issu"   "prison"
##  [2,] "pollut" "draft"     "permit"     "local"     "titl"   "health"
##  [3,] "rule"   "program"   "use"        "plan"      "vi"     "peopl"
##  [4,] "popul"  "agenc"     "consid"     "water"     "right"  "citi"
##  [5,] "also"   "action"    "organ"      "agenda"    "civil"  "project"
##  [6,] "impact" "state"     "comment"    "comment"   "agenc"  "impact"
##  [7,] "health" "effort"    "communiti"  "particip"  "work"   "park"
##  [8,] "air"    "epa"       "make"       "econom"    "health" "includ"
##  [9,] "area"   "will"      "feder"      "govern"    "feder"  "nation"
## [10,] "must"   "polici"    "overburden" "work"      "offic"  "center"
##       Topic 7
##  [1,] "communiti"
##  [2,] "enforc"
##  [3,] "includ"
##  [4,] "comment"
##  [5,] "monitor"
##  [6,] "action"
##  [7,] "air"
##  [8,] "provid"
##  [9,] "complianc"
## [10,] "pollut"
```

```
theta <- tmResult$topics
beta <- tmResult$terms
vocab <- (colnames(beta))
```

There are multiple proposed methods for how to measure the best k value. You can go down the rabbit hole

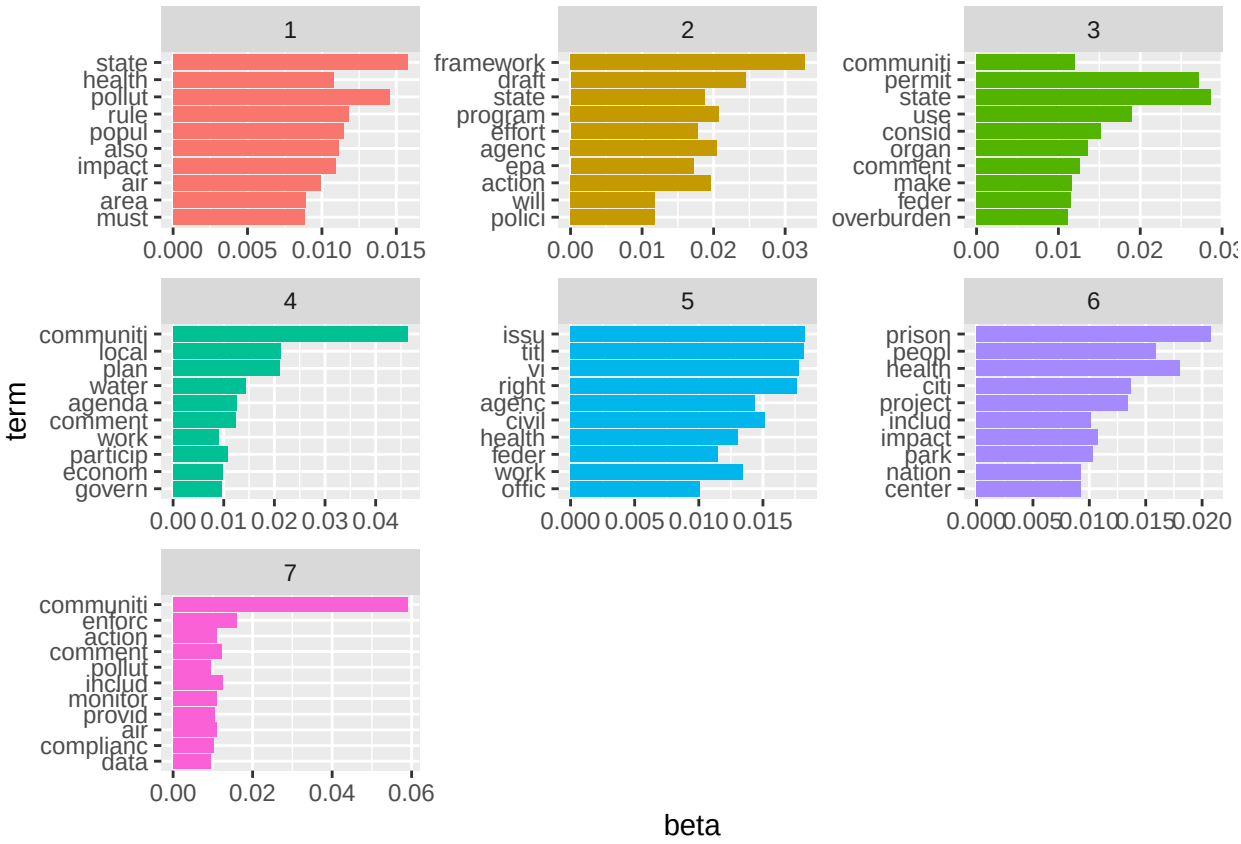
here

```
comment_topics <- tidy(topicModel_k7, matrix = "beta")

top_terms <- comment_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

```
## # A tibble: 71 x 3
##    topic term      beta
##    <int> <chr>    <dbl>
##  1     1 state  0.0158
##  2     1 pollut 0.0145
##  3     1 rule   0.0118
##  4     1 popul  0.0114
##  5     1 also   0.0111
##  6     1 impact 0.0109
##  7     1 health 0.0108
##  8     1 air    0.00994
##  9     1 area   0.00894
## 10     1 must   0.00885
## # ... with 61 more rows
```

```
# beta is the probability of that term in that topic
```

```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

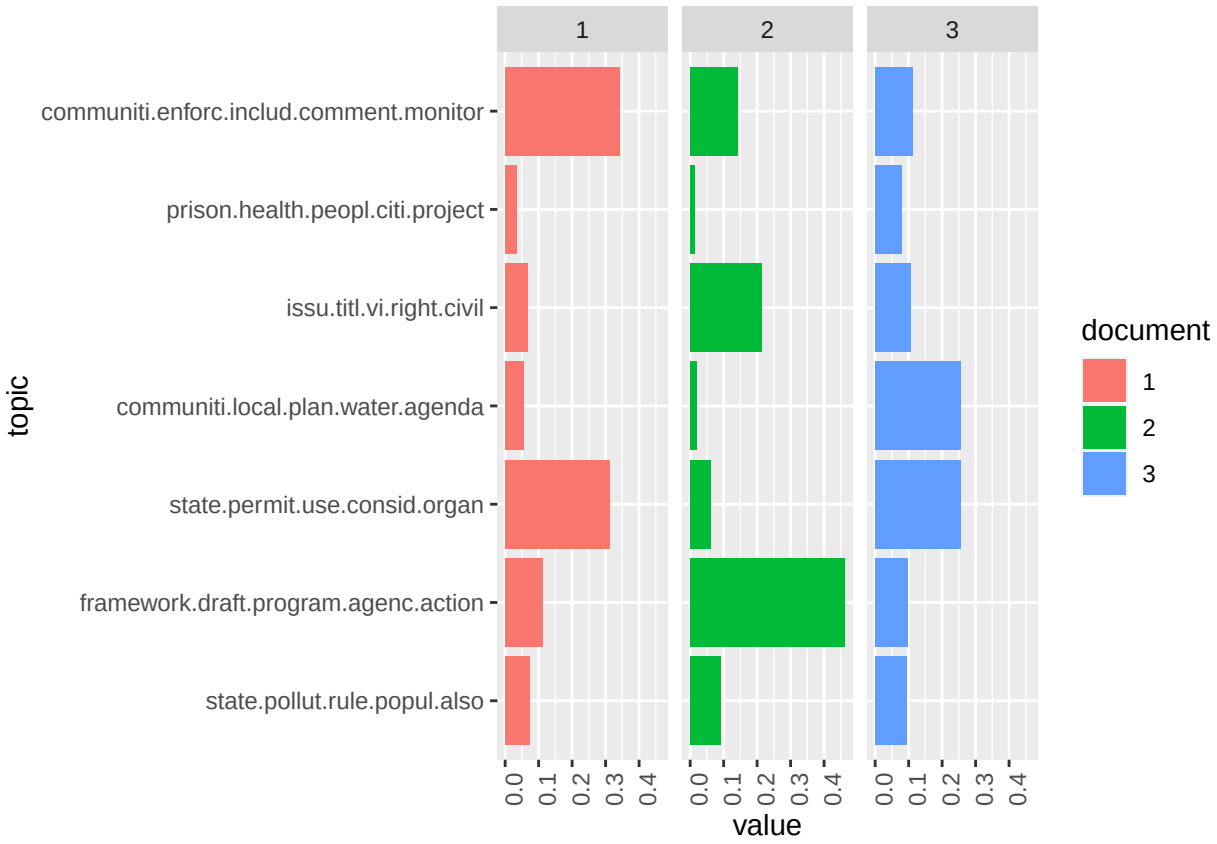


Let's assign names to the topics so we know what we are working with. We can name them by their top terms

```
top5termsPerTopic <- terms(topicModel_k7, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
# guess the names for the topics
```

We can explore the theta matrix, which contains the distribution of each topic over each document

```
exampleIds <- c(1, 2, 3)
N <- length(exampleIds)

#lapply(epa_corp[exampleIds], as.character) #uncomment to view example text
# get topic proportions form example documents
topicProportionExamples <- theta[exampleIds,]
colnames(topicProportionExamples) <- topicNames
vizDataFrame <- melt(cbind(data.frame(topicProportionExamples), document=factor(1:N)), variable.name =
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)
```



Here's a neat JSON-based model visualizer

```
library(LDAvis)
library("tsne")
svd_tsne <- function(x) tsne(svd(x)$u)
json <- createJSON(
  phi = tmResult$terms,
  theta = tmResult$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab="")
)
```

```
## sigma summary: Min. : 33554432 |1st Qu. : 33554432 |Median : 33554432 |Mean : 33554432 |3rd Qu. : 335
## Epoch: Iteration #100 error is: 10.9455830360371
## Epoch: Iteration #200 error is: 0.314512839375737
## Epoch: Iteration #300 error is: 0.215382326464989
## Epoch: Iteration #400 error is: 0.184273297684405
## Epoch: Iteration #500 error is: 0.160414856663629
## Epoch: Iteration #600 error is: 0.158651916805303
## Epoch: Iteration #700 error is: 0.158651648771023
## Epoch: Iteration #800 error is: 0.158651648602134
```

```
## Epoch: Iteration #900 error is: 0.158651648601466
## Epoch: Iteration #1000 error is: 0.158651648600633
serVis(json)
## Loading required namespace: servr
# relevance metric llamda = weighting things highly if they are highly focused, or can choose to weight
```

Assignment:

Either:

A) continue on with the analysis we started (choose diff values for k and justify the decision for that k-value):

Run three more models and select the overall best value for k (the number of topics) - include some justification for your selection: theory, FindTopicsNumber() optimization metrics, interpretability, LDAvis

Assignment Model #1

For my first experimental value of `k`, I will test out $k = 3$ because each of the EPA's priority areas seem to fall into one of three main categories: rules and regulations, science and moniroing, and culture and humanities. I would expect to see the most words in the rules and regulations category, because the EPA is focused on that the most, while I expect the least to fall into the culture and humanities section. I think that manually choosing a value of `k` gives me a better understanding of the workflow without having a function choose the best value of `k` right off the bat. This value of `k` is small relative to the values we chose in class. Next, I will use a much larger `k` and compare results.

```
k <- 3

topicModel_k3 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 3; V = 2781; M = 77
## 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 <- posterior(topicModel_k3)

attributes(tmResult)
```

```
## $names
## [1] "terms"  "topics"
```

```
beta <- tmResult$terms   # get beta from results
dim(beta)                # K distributions over nTerms(DTM) terms# lengthOfVocab
```

```
## [1]    3 2781
```

```
terms(topicModel_k3, 30)
```

```
##       Topic 1     Topic 2     Topic 3
##  [1,] "communiti" "communiti" "state"
##  [2,] "air"       "includ"    "framework"
##  [3,] "pollut"    "enforc"    "draft"
##  [4,] "state"     "health"    "comment"
##  [5,] "impact"    "right"     "communiti"
##  [6,] "also"      "action"    "permit"
##  [7,] "requir"    "comment"   "use"
##  [8,] "epa"       "protect"   "agenda"
##  [9,] "agenc"     "nation"    "effort"
## [10,] "health"    "complianc" "overburden"
## [11,] "provid"    "civil"     "will"
## [12,] "popul"     "agenc"     "program"
## [13,] "rule"      "monitor"   "goal"
## [14,] "plan"      "titl"      "develop"
## [15,] "avail"     "see"       "consid"
## [16,] "must"      "peopl"     "work"
## [17,] "inform"    "prison"    "make"
## [18,] "guidanc"   "project"   "local"
## [19,] "comment"   "address"   "feder"
## [20,] "use"       "plan"      "opportun"
## [21,] "area"      "new"       "polici"
## [22,] "program"   "need"      "govern"
## [23,] "implement" "execut"    "process"
## [24,] "assess"    "data"      "agenc"
## [25,] "effect"    "creat"     "action"
## [26,] "risk"      "region"    "water"
## [27,] "facil"     "access"    "epa"
## [28,] "clean"     "can"       "engag"
## [29,] "standard"  "offic"     "issu"
## [30,] "final"     "vi"        "support"
```

Assigment Model #2

For my first experimental value of `k`, I will test out a much larger number; `k` = 15 because using `k` = 3 did not show much distinction in each category (I saw significant overlap of words). I think that since the EPA's documents seem to include a broader range of topics than just 3, perhaps I should try to create much smaller topics that have a clear focus. The results show that there is more distinction between categories, such as `Topic 11` that seems to include more industry and population-wide industrial issues, compared to `Topic 12` that is more nature-focused.

```
k <- 15

# feed in the DFM and the num of topics to look for and number of iterations, this function estimates t
topicModel_k15 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 15; V = 2781; M = 77
## 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 <- posterior(topicModel_k15)

attributes(tmResult)
```

```
## $names
## [1] "terms"  "topics"
```

```
beta <- tmResult$terms   # get beta from results
dim(beta)                # K distributions over nTerms(DTM) terms# lengthOfVocab
```

```
## [1]   15 2781
```

```
terms(topicModel_k15, 10)
```

```
##       Topic 1   Topic 2     Topic 3    Topic 4     Topic 5   Topic 6
##  [1,] "prison"  "communiti" "need"     "state"     "right"   "state"
##  [2,] "sourc"   "pollut"    "peopl"    "program"   "civil"   "health"
##  [3,] "facil"   "health"    "work"     "epa"       "vi"      "popul"
##  [4,] "popul"   "air"       "help"     "feder"     "titl"    "rule"
##  [5,] "center"  "reduc"     "subject"  "farmwork"  "agenc"   "ejscreen"
##  [6,] "initi"   "comment"   "make"     "pesticid"  "issu"    "asthma"
##  [7,] "report"  "polici"    "sent"     "tribe"     "feder"   "pollut"
##  [8,] "project" "impact"    "lung"     "work"      "act"     "communiti"
##  [9,] "site"    "protect"   "strategi" "implement" "plan"    "agenc"
## [10,] "peopl"   "develop"   "tai"      "follow"    "program" "avail"
##       Topic 7     Topic 8   Topic 9     Topic 10  Topic 11     Topic 12
##  [1,] "data"      "health"  "permit"    "requir"  "framework"  "communiti"
```

```
##  [2,] "execut"    "park"    "state"     "comment" "draft"      "water"
##  [3,] "director"  "citi"    "consid"    "use"     "agenc"      "econom"
##  [4,] "state"     "peopl"   "air"       "impact"  "effort"     "local"
##  [5,] "process"   "green"   "use"       "also"    "communiti"  "june"
##  [6,] "texa"      "law"     "framework" "concern" "state"      "agenda"
##  [7,] "citizen"   "project" "carolina"  "address" "action"     "clean"
##  [8,] "feder"     "see"     "feder"     "provid"  "develop"    "comment"
##  [9,] "communiti" "poor"    "grant"     "exampl"  "overburden" "area"
## [10,] "address"   "space"   "meet"      "includ"  "epa"        "effort"
##       Topic 13    Topic 14    Topic 15
##  [1,] "communiti" "communiti" "juli"
##  [2,] "plan"      "enforc"    "polici"
##  [3,] "local"     "monitor"   "infrastructur"
##  [4,] "govern"    "includ"    "part"
##  [5,] "use"       "action"    "energi"
##  [6,] "action"    "complianc" "access"
##  [7,] "particip"  "air"       "natur"
##  [8,] "land"      "assess"    "comment"
##  [9,] "develop"   "region"    "regul"
## [10,] "level"     "report"    "pipelin"
```

Assignment Model #3

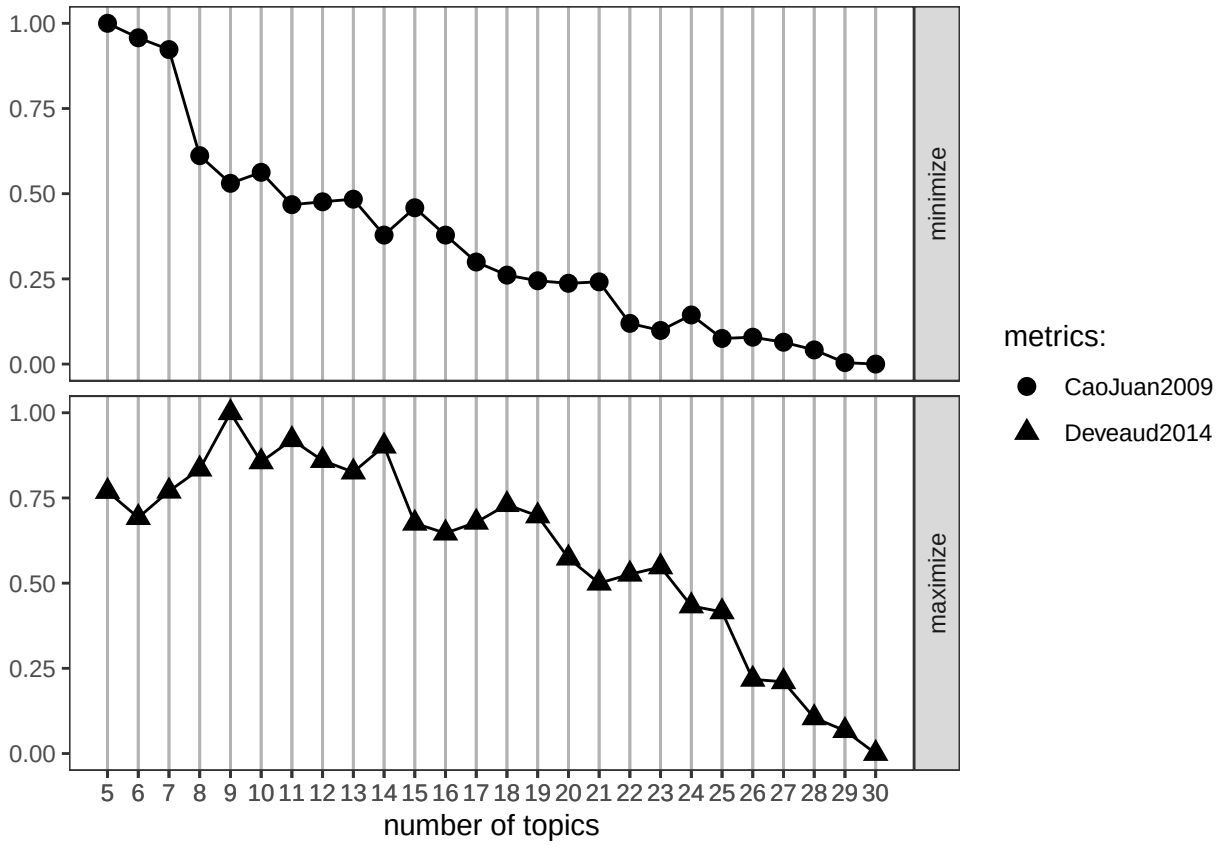
For the final model for this data, I will use the function `FindTopicsNumber()` to help me visualize the best value of `k`, but I will change it from the class code by adjusting the `topics` argument. I also tried changing the method argument (there are other options than the ones here in the function documentation) but they did not run.

```
# fit the model by running a series of models, starting with 2 topics and ranging to 20 topics
result <- FindTopicsNumber(
  dfm,
  topics = seq(from = 5, to = 30, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 100),
  verbose = TRUE
)
```

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

```
FindTopicsNumber_plot(result)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```


1.00
0.75
0.50
0.25
0.00
minimize
maximize
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
number of topics
metrics:
CaoJuan2009
Deveaud2014