# EDS 231: Topic 6

Wylie Hampson

5/4/2022

## In class lab:

### Assignment located below.

### Import 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/comment
```

```
## 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
```

### Create 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)
```

### Tokenize the corpus.

```
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")
```

### Create the DFM.

```
dfm_comm<- dfm(toks1, tolower = TRUE)
dfm <- dfm_wordstem(dfm_comm)
dfm <- dfm_trim(dfm, min_docfreq = 2) #remove terms only appearing in one doc (min_termfreq = 10)

print(head(dfm))
```

```
## 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 with all zeros.**

```
#remove rows (docs) with all zeros
sel_idx <- slam::row_sums(dfm) > 0
dfm <- dfm[sel_idx, ]
#comments_df <- dfm[sel_idx, ]
```

**Here's where we actually create out first model. This model has 9 different topics.**

```
k <- 9

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)
attributes(tmResult)
```

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

**The beta value shows the likelihood that a token will show up in a topic.**

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

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

**The top 10 terms for each topic.**

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

```
##       Topic 1          Topic 2    Topic 3    Topic 4     Topic 5     Topic 6
##  [1,] "communiti"      "prison"   "state"    "communiti" "framework" "communiti"
##  [2,] "water"          "farmwork" "permit"   "plan"      "effort"    "enforc"
##  [3,] "clean"          "pesticid" "consid"   "local"     "develop"   "comment"
##  [4,] "can"            "health"   "use"      "strategi"  "draft"     "includ"
##  [5,] "econom"         "facil"    "air"      "comment"   "action"    "monitor"
##  [6,] "energi"         "center"   "feder"    "agenda"    "communiti" "action"
##  [7,] "polici"         "sourc"    "meet"     "particip"  "agenc"     "air"
##  [8,] "overburden"     "popul"    "grant"    "help"      "tool"      "pollut"
##  [9,] "infrastructur" "project"  "carolina" "work"      "agenda"    "provid"
## [10,] "power"          "exposur"  "comment"  "like"      "advanc"    "see"
##       Topic 7   Topic 8     Topic 9
##  [1,] "right"   "communiti" "program"
##  [2,] "civil"   "impact"    "issu"
##  [3,] "health"  "pollut"    "state"
##  [4,] "peopl"   "rule"      "agenc"
##  [5,] "vi"      "air"       "epa"
##  [6,] "citi"    "state"     "feder"
##  [7,] "park"    "health"    "polici"
##  [8,] "includ"  "popul"     "requir"
##  [9,] "address" "also"      "regul"
## [10,] "law"     "area"      "guidanc"
```

**This function helps calculate different metrics to estimate the most preferable number of topics for an LDA model.**

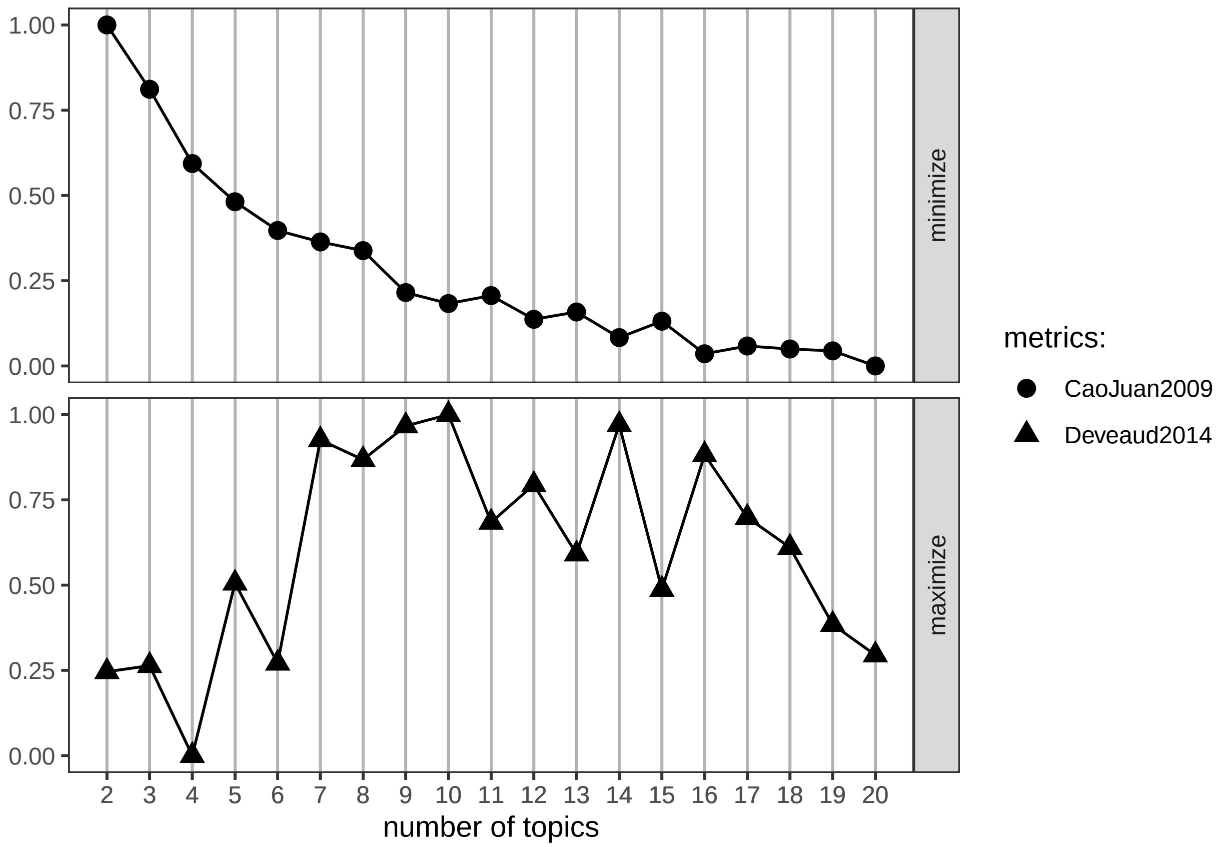
```
#
result <- FindTopicsNumber(
  dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)
```

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

**Here we plot the results from the above function.**

```
FindTopicsNumber_plot(result)
```

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



**Here we try to make a model with 7 topics instead of 9. We can see from the above plot that 7 seems to be a valid number from the Deveaud2014 metric.**

```
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,] "communiti" "prison"  "state"    "framework"  "communiti" "communiti"
##  [2,] "state"     "health"  "permit"   "communiti"  "enforc"    "plan"     
##  [3,] "pollut"    "peopl"   "consid"   "draft"      "monitor"   "local"    
##  [4,] "rule"      "project" "feder"    "develop"    "comment"   "comment"  
##  [5,] "popul"     "citi"    "polici"   "effort"     "pollut"    "strategi" 
##  [6,] "also"      "site"    "air"      "action"     "action"    "particip" 
##  [7,] "impact"    "park"    "comment"  "agenda"     "data"      "use"      
##  [8,] "health"    "law"     "meet"     "state"      "air"       "make"     
##  [9,] "provid"    "center"  "program"  "epa"        "requir"    "like"     
## [10,] "air"       "can"     "opportun" "overburden" "region"    "action"   
##       Topic 7    
##  [1,] "agenc"    
##  [2,] "issu"     
##  [3,] "right"    
##  [4,] "feder"    
##  [5,] "vi"       
##  [6,] "titl"     
##  [7,] "civil"    
##  [8,] "work"
##  [9,] "address"
## [10,] "includ"
```

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

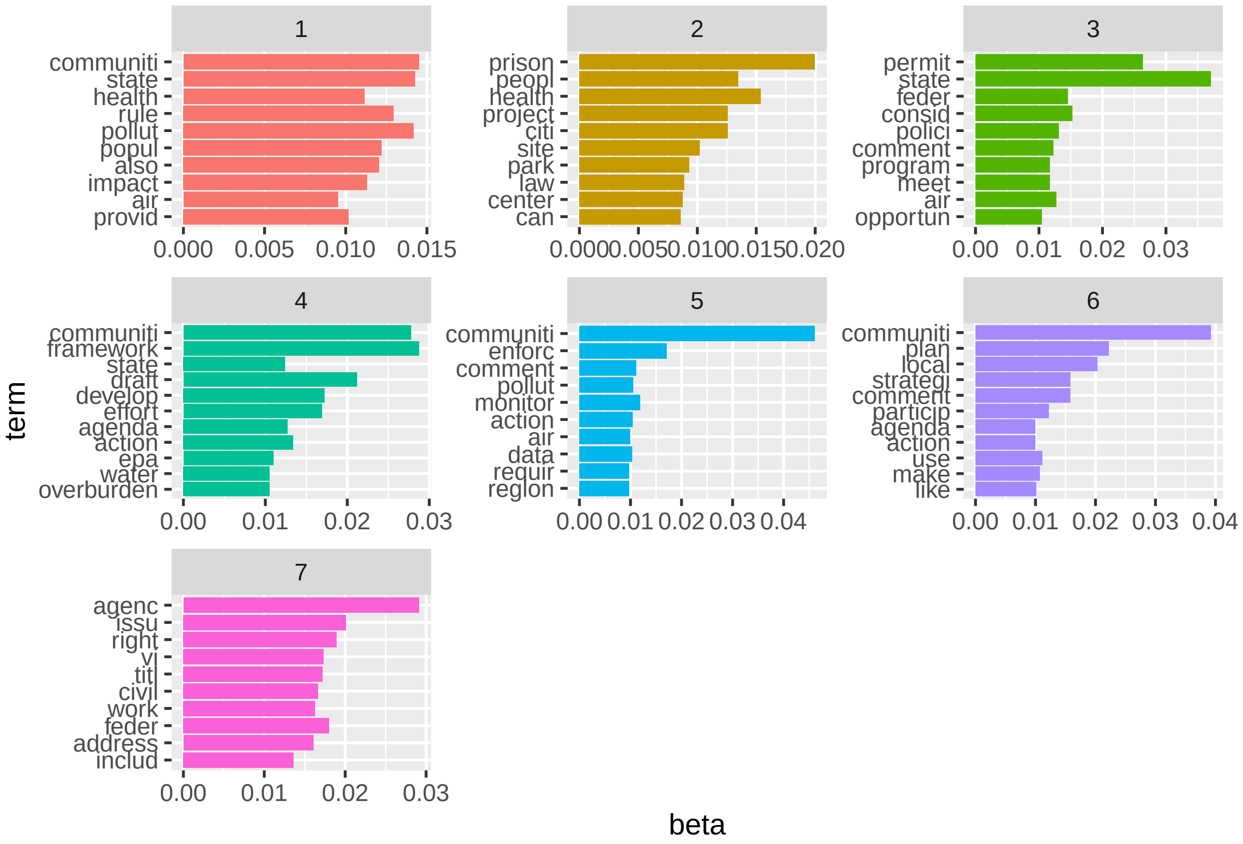
```
comment_topics <- tidy(topicModel_k7, matrix = "beta")

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

top_terms
```

```
## # A tibble: 72 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 communiti 0.0145
##  2     1 state     0.0143
##  3     1 pollut    0.0142
##  4     1 rule      0.0130
##  5     1 popul     0.0122
##  6     1 also      0.0121
##  7     1 impact    0.0113
##  8     1 health    0.0112
##  9     1 provid    0.0102
## 10     1 air       0.00951
## # ... with 62 more rows
```

```
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()
```

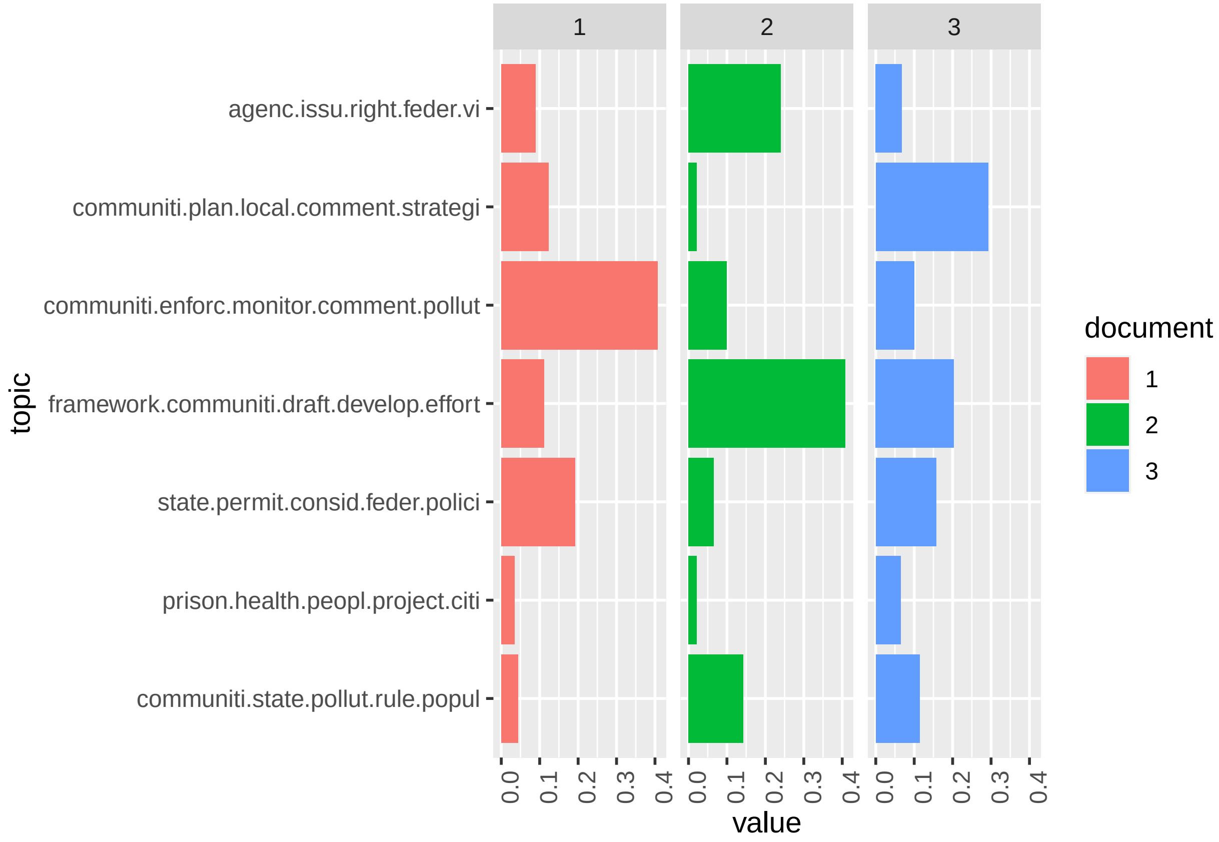


```
top5termsPerTopic <- terms(topicModel_k7, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
```

**This plot shows the theta value, which shows topics were most common in which documents.**

```
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)
```



**This creates an interactive plot that helps summarize each topics.**

```r
library(LDAvis)
library("tsne")
```

```
## Warning: package 'tsne' was built under R version 4.1.1
```

```r
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: 17.4563898252664

## Epoch: Iteration #200 error is: 0.859739299183723

## Epoch: Iteration #300 error is: 0.316594408689853
## Epoch: Iteration #400 error is: 0.213001893837484
```

```
## Epoch: Iteration #500 error is: 0.173374565210033
```

```
## Epoch: Iteration #600 error is: 0.1648196907724
```

```
## Epoch: Iteration #700 error is: 0.160093588073606
```

```
## Epoch: Iteration #800 error is: 0.158779800985257
```

```
## Epoch: Iteration #900 error is: 0.158657639370979
```

```
## Epoch: Iteration #1000 error is: 0.158652093997657
```

```
serVis(json)
```

```
## Loading required namespace: servr
```

## Assignment:

**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**

*For this assignemnt I based my choices for how many topics to analyize from the FindTopicsNumbers plot above. From the Deveaud2014 metrics we can see that 10 and 14 perform best, and from the CaoJuan2009 metric we can see that 20 topics performs best.*

**First model, with 10 topics.**

```
k <- 10

topicModel_k10 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 10; 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_k10)
terms(topicModel_k10, 10)
```

```
##       Topic 1     Topic 2     Topic 3    Topic 4   Topic 5     Topic 6
##  [1,] "framework" "communiti" "prison"   "program" "permit"    "health"
##  [2,] "draft"     "plan"      "center"   "titl"    "state"     "peopl"
##  [3,] "communiti" "strategi"  "facil"    "feder"   "consid"    "citi"
##  [4,] "effort"    "local"     "project"  "state"   "use"       "right"
##  [5,] "action"    "agenda"    "sourc"    "issu"    "air"       "park"
##  [6,] "develop"   "comment"   "may"      "vi"      "grant"     "includ"
##  [7,] "agenc"     "govern"    "contamin" "agenc"   "implement" "green"
##  [8,] "state"     "way"       "report"   "act"     "carolina"  "color"
##  [9,] "epa"       "help"      "popul"    "epa"     "feder"     "project"
## [10,] "comment"   "mani"      "york"     "civil"   "draft"     "climat"
##       Topic 7     Topic 8    Topic 9         Topic 10
##  [1,] "communiti" "data"     "water"         "rule"
##  [2,] "pollut"    "use"      "communiti"     "state"
##  [3,] "enforc"    "nation"   "energi"        "impact"
##  [4,] "includ"    "comment"  "econom"        "pollut"
##  [5,] "complianc" "farmwork" "infrastructur" "health"
##  [6,] "air"       "pesticid" "e.g"           "also"
##  [7,] "action"    "agenc"    "can"           "popul"
##  [8,] "permit"    "texa"     "area"          "ejscreen"
##  [9,] "monitor"   "process"  "clean"         "asthma"
## [10,] "requir"    "health"   "assist"        "air"
```

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

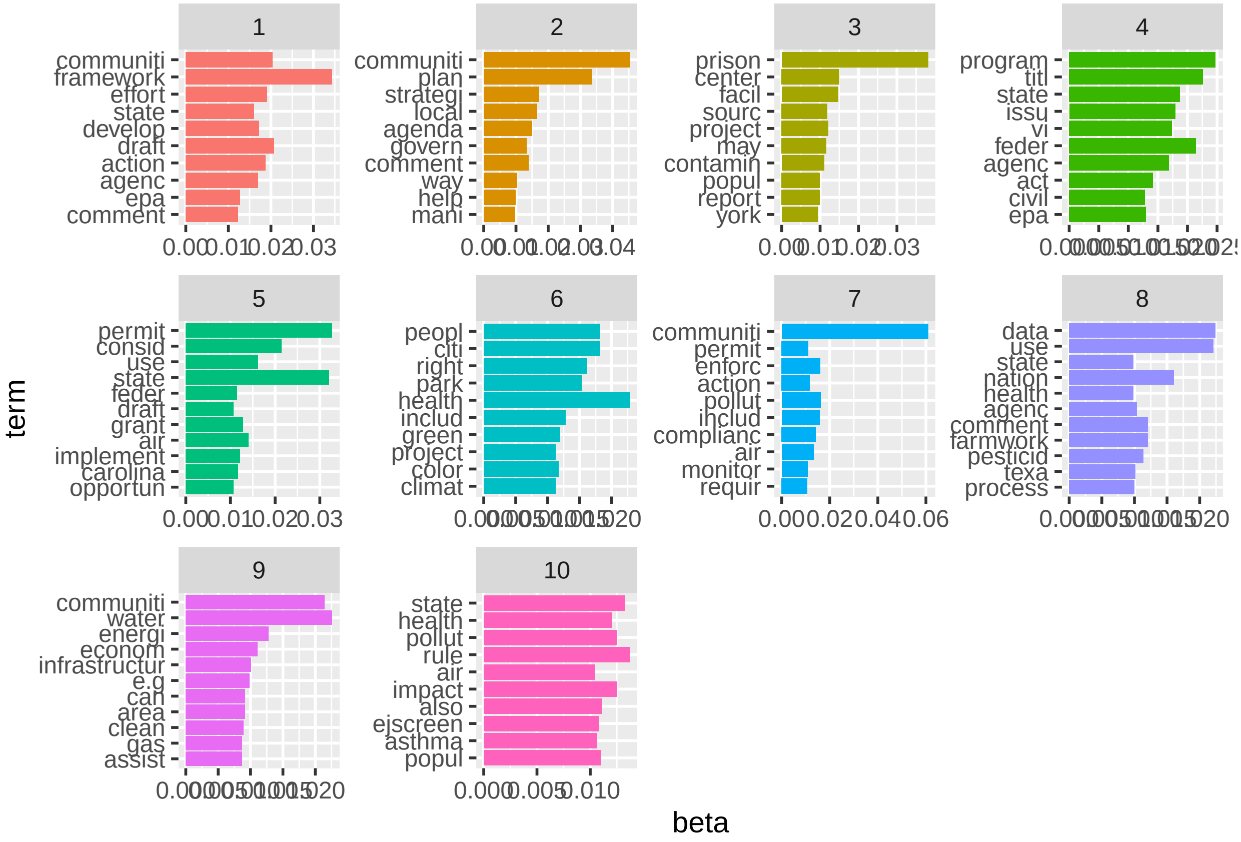
```
comment_topics <- tidy(topicModel_k10, matrix = "beta")

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

top_terms
```

```
## # A tibble: 103 x 3
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 framework 0.0344
##  2     1 draft     0.0207
##   3      1 communiti 0.0204
##   4      1 effort    0.0190
##   5      1 action    0.0188
##   6      1 develop   0.0173
##   7      1 agenc     0.0170
##   8      1 state     0.0160
##   9      1 epa       0.0127
##  10      1 comment   0.0123
## # ... with 93 more rows
```

```
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()
```

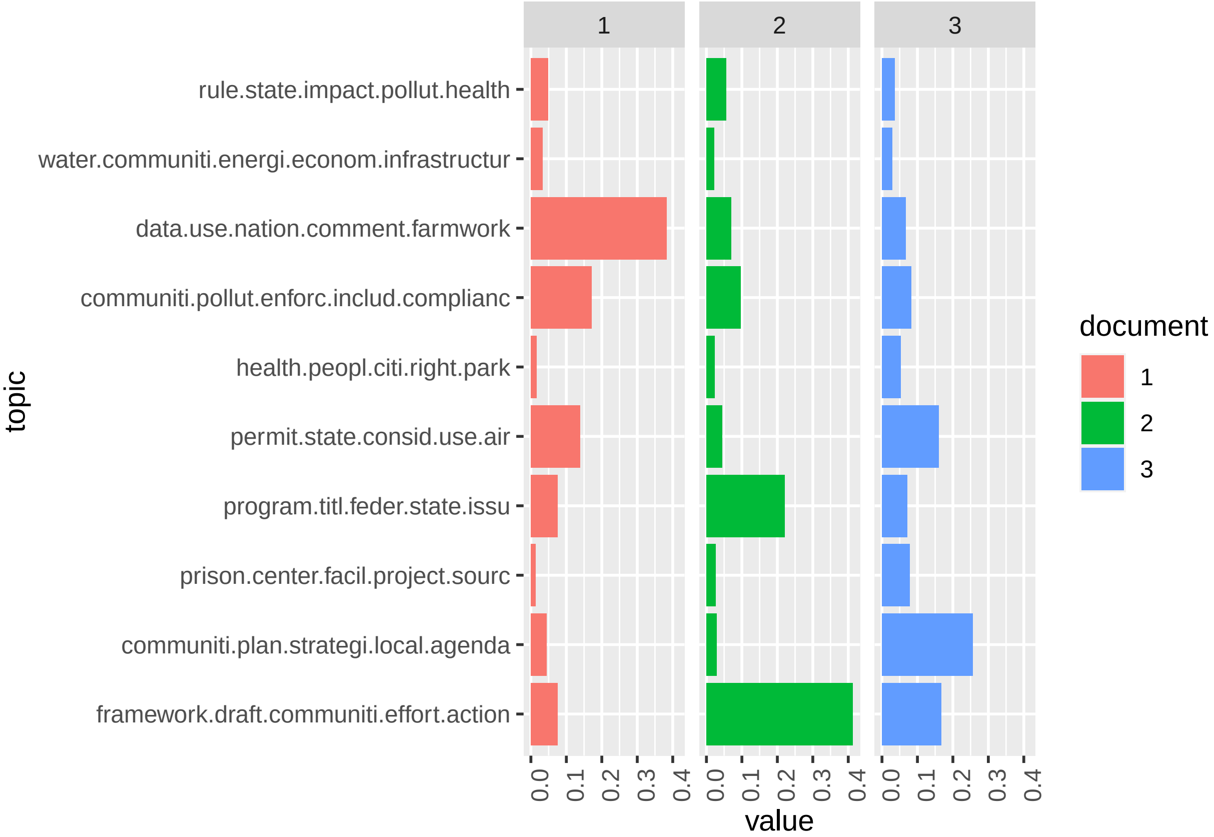


```
top5termsPerTopic <- terms(topicModel_k10, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
```

```
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)
```



```
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: 8.223844709511

## Epoch: Iteration #200 error is: 0.324402512448517
## Epoch: Iteration #300 error is: 0.265154476473876
```

```
## Epoch: Iteration #400 error is: 0.254376848773314
```

```
## Epoch: Iteration #500 error is: 0.253151327346259
```

```
## Epoch: Iteration #600 error is: 0.252745388179318
```

```
## Epoch: Iteration #700 error is: 0.2526188887118
```

```
## Epoch: Iteration #800 error is: 0.252461986671255
```

```
## Epoch: Iteration #900 error is: 0.25178083701019
```

```
## Epoch: Iteration #1000 error is: 0.250904533591916
```

```
serVis(json)
```

**Second model, with 16 topics. I think 16 is a good number to try because it scores well with both metrics from the FindTopicsNumber plot, and from the lab it we know that it could have 9 topics for the EPA priority areas, and 7 topics for the EPA's additional topics.**

```
k <- 16

topicModel_k16 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 16; 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_k16)
terms(topicModel_k16, 10)
```

```
##       Topic 1   Topic 2     Topic 3      Topic 4     Topic 5      Topic 6
##  [1,] "program" "communiti" "framework"  "communiti" "help"       "permit"
##  [2,] "state"   "enforc"    "communiti"  "pollut"    "work"       "state"
##  [3,] "feder"   "monitor"   "draft"      "polici"    "sent"       "consid"
##  [4,] "requir"  "air"       "effort"     "energi"    "comment"    "grant"
##  [5,] "nation"  "pollut"    "state"      "comment"   "us"         "carolina"
##  [6,] "epa"     "complianc" "support"    "new"       "strategi"   "air"
##  [7,] "follow"  "permit"    "agenda"     "protect"   "way"        "feder"
##  [8,] "polici"  "report"    "epa"        "reduct"    "lung"       "use"
##  [9,] "tribe"   "avail"     "will"       "reduc"     "subject"    "meet"
## [10,] "regul"   "includ"    "overburden" "counti"    "ejstrategi" "qualiti"
##       Topic 7     Topic 8   Topic 9   Topic 10   Topic 11    Topic 12
##  [1,] "data"      "health"  "comment" "water"    "agenc"     "right"
##  [2,] "communiti" "peopl"   "provid"  "econom"   "action"    "civil"
##  [3,] "permit"    "park"    "plan"    "often"    "director"  "titl"
##  [4,] "texa"      "citi"    "use"     "individu" "health"    "vi"
##  [5,] "particip"  "see"     "concern" "clean"    "scienc"    "agenc"
##  [6,] "feder"     "color"   "address" "distress" "develop"   "issu"
##  [7,] "citizen"   "green"   "includ"  "work"     "tool"      "feder"
##  [8,] "process"   "includ"  "need"    "health"   "comment"   "act"
##  [9,] "one"       "project" "also"    "peopl"    "communiti" "impact"
## [10,] "opportun"  "law"     "exampl"  "e.g"      "recommend" "epa"
##       Topic 13   Topic 14    Topic 15    Topic 16
##  [1,] "farmwork" "communiti" "communiti" "prison"
##  [2,] "health"   "local"     "rule"      "facil"
##  [3,] "pesticid" "plan"      "pollut"    "popul"
##  [4,] "exposur"  "govern"    "state"     "sourc"
##  [5,] "enforc"   "use"       "asthma"    "center"
##  [6,] "state"    "agenda"    "health"    "project"
##  [7,] "mercuri"  "chang"     "popul"     "water"
##  [8,] "risk"     "resourc"   "impact"    "incarcer"
##  [9,] "access"   "land"      "ejscreen"  "site"
## [10,] "level"    "particip"  "air"       "locat"
```

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

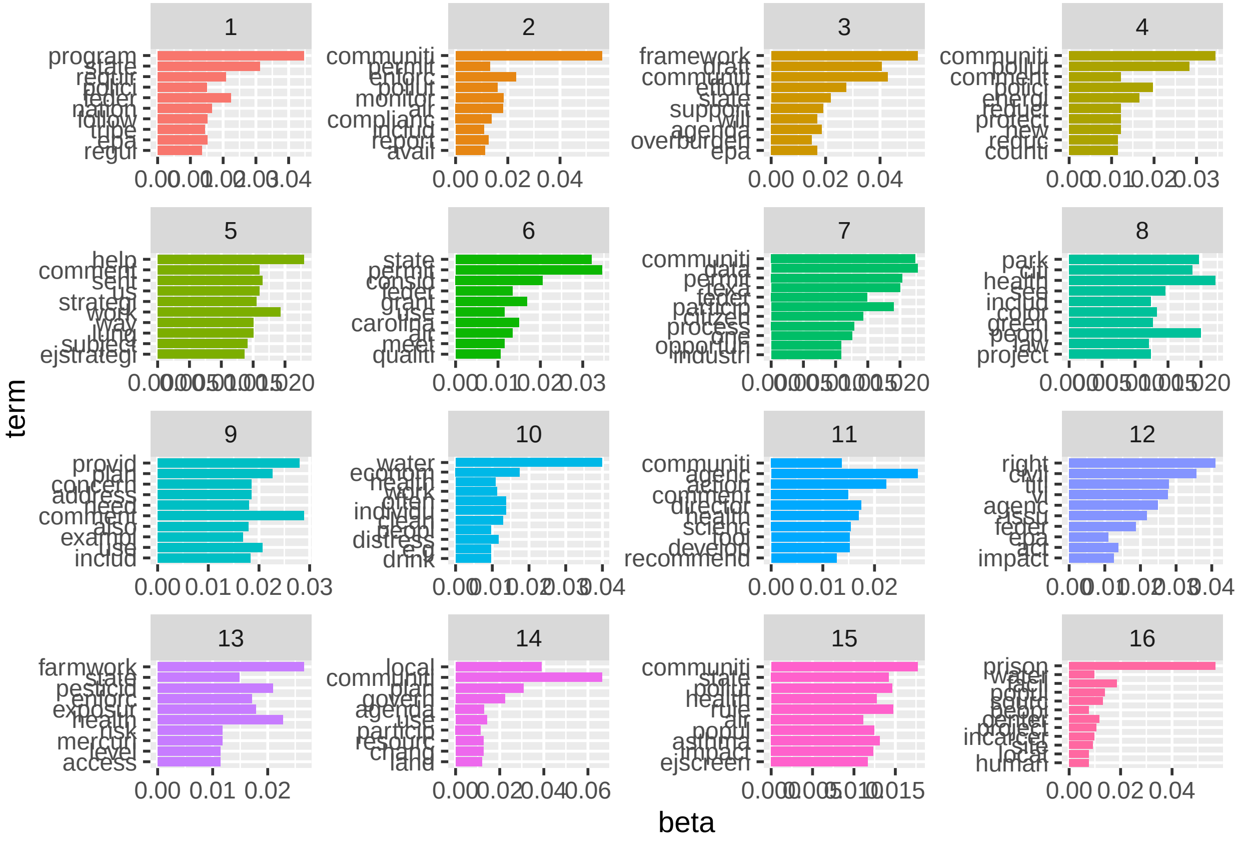
```
comment_topics <- tidy(topicModel_k16, matrix = "beta")

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

top_terms
```

```
## # A tibble: 164 x 3
##    topic term      beta
##    <int> <chr>    <dbl>
##  1     1 program 0.0447
##  2     1 state   0.0313
##  3     1 feder   0.0224
##  4     1 requir  0.0208
##  5     1 nation  0.0165
##  6     1 epa     0.0153
##  7     1 follow  0.0153
##  8     1 polici  0.0150
##  9     1 tribe   0.0145
## 10     1 regul   0.0135
## # ... with 154 more rows
```

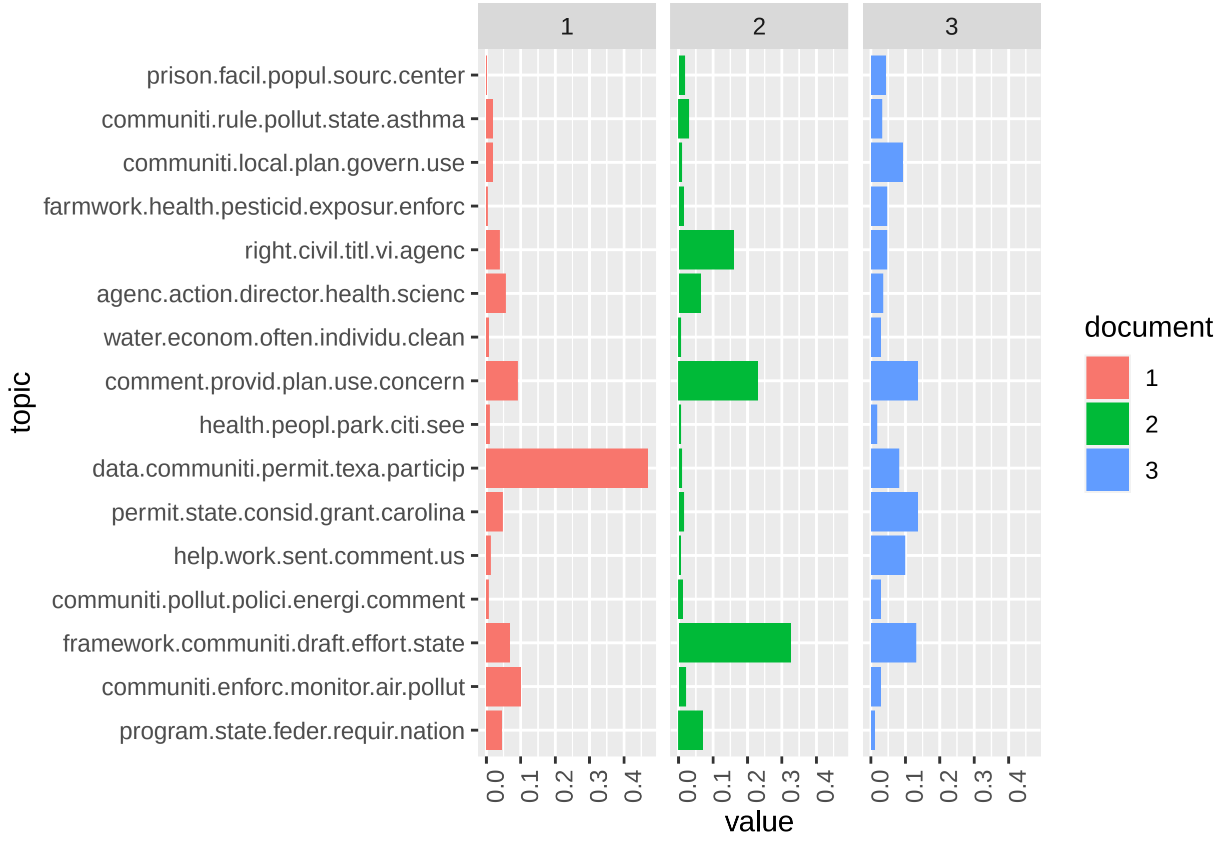
```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



```
top5termsPerTopic <- terms(topicModel_k16, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
```

```
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)
```



```
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: 14.151472968293

## Epoch: Iteration #200 error is: 0.721716595263361

## Epoch: Iteration #300 error is: 0.474816594697122

## Epoch: Iteration #400 error is: 0.373638104442202

## Epoch: Iteration #500 error is: 0.366055616536109

## Epoch: Iteration #600 error is: 0.360760242508478

## Epoch: Iteration #700 error is: 0.35786945665855

## Epoch: Iteration #800 error is: 0.357153981845132

## Epoch: Iteration #900 error is: 0.356659189112055

## Epoch: Iteration #1000 error is: 0.356080367984445
```

```r
serVis(json)
```

**Third model, with 20 topics.**

```r
k <- 20

topicModel_k20 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 20; 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!
```

```r
tmResult <- posterior(topicModel_k20)
terms(topicModel_k20, 10)
```

```
##       Topic 1     Topic 2      Topic 3      Topic 4     Topic 5   Topic 6
##  [1,] "framework" "communiti"  "communiti"  "draft"     "state"   "communiti"
##  [2,] "action"    "effort"     "pollut"     "framework" "program" "plan"
##  [3,] "agenda"    "support"    "will"       "state"     "feder"   "use"
##  [4,] "agenc"     "local"      "polici"     "develop"   "regul"   "govern"
##  [5,] "comment"   "impact"     "comment"    "final"     "tribe"   "local"
##  [6,] "epa"       "overburden" "new"        "tool"      "requir"  "land"
##  [7,] "includ"    "provid"     "impact"     "relat"     "follow"  "process"
##  [8,] "communiti" "industri"   "overburden" "continu"   "polici"  "agenda"
##  [9,] "goal"      "enhanc"     "develop"    "ejtg"      "epa"     "particip"
## [10,] "one"       "within"     "protect"    "will"      "train"   "adapt"
##       Topic 7      Topic 8     Topic 9     Topic 10  Topic 11    Topic 12
##  [1,] "water"      "communiti" "farmwork"  "park"    "state"     "permit"
##  [2,] "job"        "enforc"    "pesticid"  "peopl"   "asthma"    "state"
##  [3,] "clean"      "monitor"   "health"    "health"  "communiti" "use"
##  [4,] "local"      "action"    "enforc"    "project" "pollut"    "consid"
##  [5,] "site"       "includ"    "exposur"   "color"   "avail"     "carolina"
##  [6,] "drink"      "complianc" "work"      "see"     "ejscreen"  "air"
##  [7,] "area"       "permit"    "risk"      "citi"    "popul"     "grant"
##  [8,] "counti"     "report"    "implement" "green"   "rule"      "meet"
##  [9,] "econom"     "avail"     "children"  "space"   "guidanc"   "organ"
## [10,] "brownfield" "protect"   "report"    "law"     "agenc"     "feder"
##       Topic 13    Topic 14 Topic 15        Topic 16   Topic 17  Topic 18
##  [1,] "agenc"     "right"  "juli"          "data"     "air"     "mercuri"
##  [2,] "program"   "civil"  "energi"        "comment"  "plan"    "sent"
##  [3,] "director"  "agenc"  "natur"         "texa"     "provid"  "strategi"
##  [4,] "scienc"    "titl"   "infrastructur" "particip" "use"     "help"
##  [5,] "engag"     "vi"     "access"        "feder"    "health"  "tai"
##  [6,] "health"    "issu"   "gas"           "region"   "also"    "lung"
##  [7,] "tool"      "plan"   "site"          "citizen"  "requir"  ">"
##  [8,] "committe"  "includ" "pipelin"       "air"      "impact"  "us"
##  [9,] "recommend" "feder"  "ohio"          "process"  "comment" "ejstrategi"
## [10,] "advisori"  "act"    "district"      "organ"    "epa"     "<"
##       Topic 19   Topic 20
##  [1,] "work"     "prison"
##  [2,] "make"     "facil"
##  [3,] "year"     "center"
##  [4,] "re"       "popul"
##  [5,] "distress" "sourc"
##  [6,] "e.g"      "peopl"
##  [7,] "individu" "project"
##  [8,] "use"      "incarcer"
##  [9,] "often"    "report"
## [10,] "econom"   "initi"
```

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

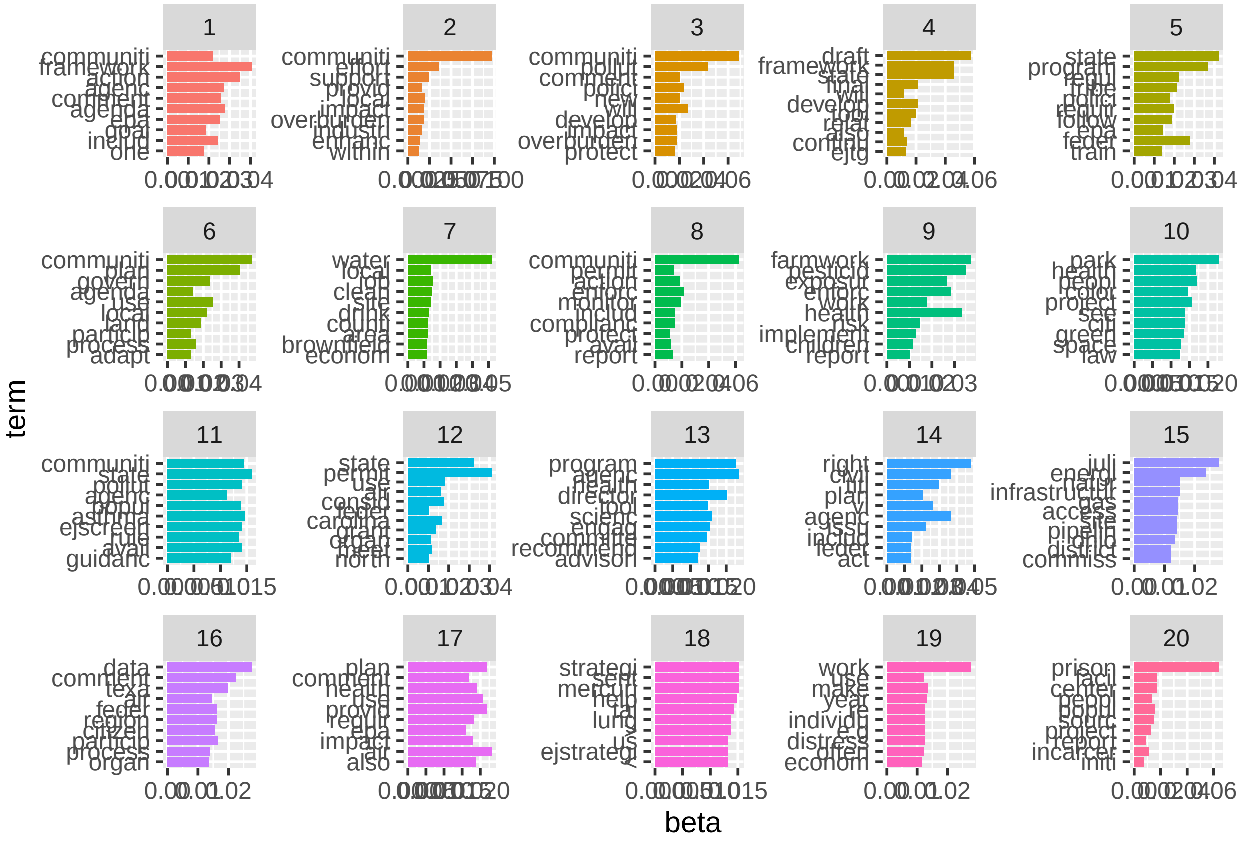
```
comment_topics <- tidy(topicModel_k20, matrix = "beta")

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

top_terms
```

```
## # A tibble: 203 x 3
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 framework 0.0407
##  2     1 action    0.0350
##  3     1 agenda    0.0279
##  4     1 agenc     0.0272
##  5     1 comment   0.0257
##  6     1 epa       0.0252
##  7     1 includ    0.0243
##  8     1 communiti 0.0218
##  9     1 goal      0.0184
## 10     1 one       0.0174
## # ... with 193 more rows
```

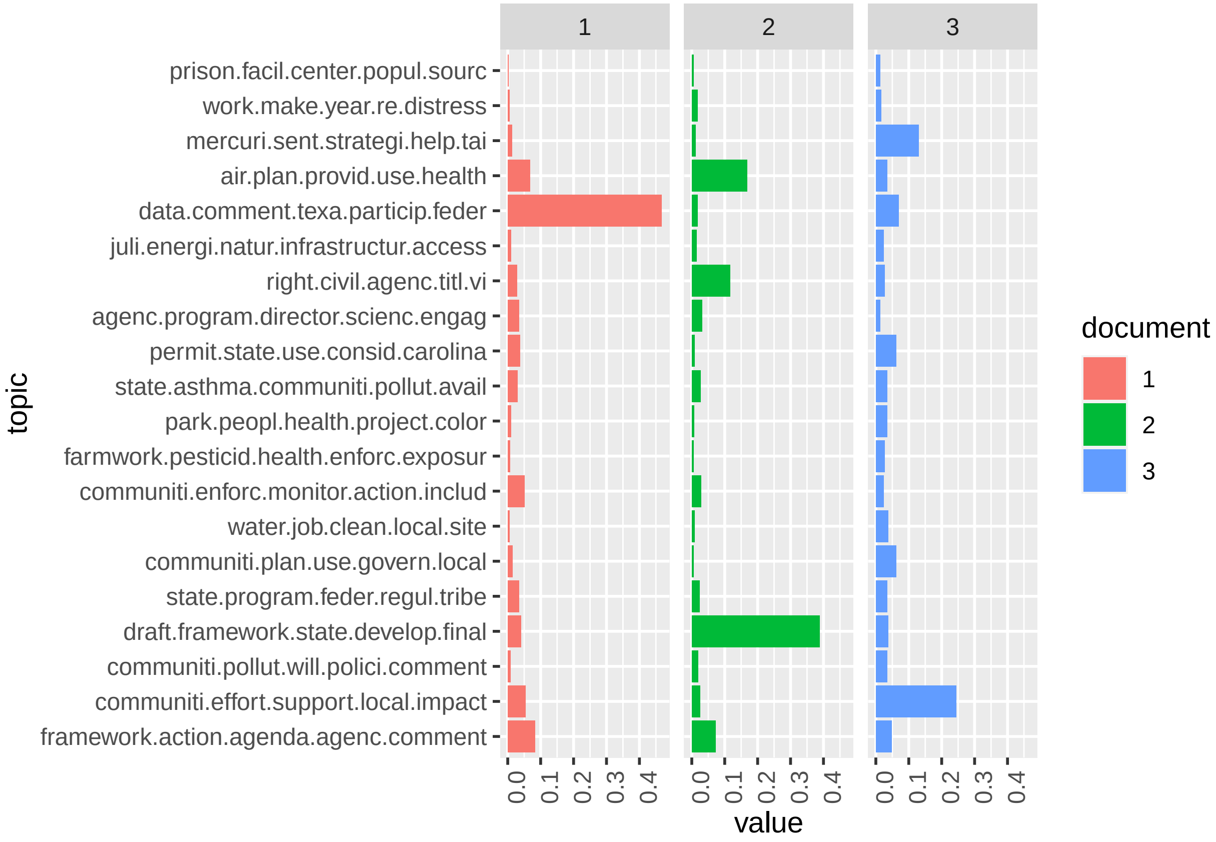
```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



```r
top5termsPerTopic <- terms(topicModel_k20, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
```

```r
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)
```



```r
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: 14.6819899533268

## Epoch: Iteration #200 error is: 0.854059028593678

## Epoch: Iteration #300 error is: 0.585455466322003

## Epoch: Iteration #400 error is: 0.456654994846496

## Epoch: Iteration #500 error is: 0.414578121257224

## Epoch: Iteration #600 error is: 0.411923091520728
## Epoch: Iteration #700 error is: 0.411719722367583
```

```
## Epoch: Iteration #800 error is: 0.411546352621736
```

```
## Epoch: Iteration #900 error is: 0.411333631374701
```

```
## Epoch: Iteration #1000 error is: 0.411088867588581
```

```
serVis(json)
```

*I think that 16 topics is probably the best number to use because it makes sense with the EPA's priority and additional topics, and it scored well with both metrics from the FindTopicsNumbers function, where as 10 and 20 topics seemed to only score well with one or the other.*