Topic 6: Topic Analysis Homework

Steven Cognac

2022-05-10

Load the data

```
comments_df <- read_csv("https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/main/dat/comme
```

```
## 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.
```

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
```

Now let's tokenize our dataset and remove stopwords

```
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

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

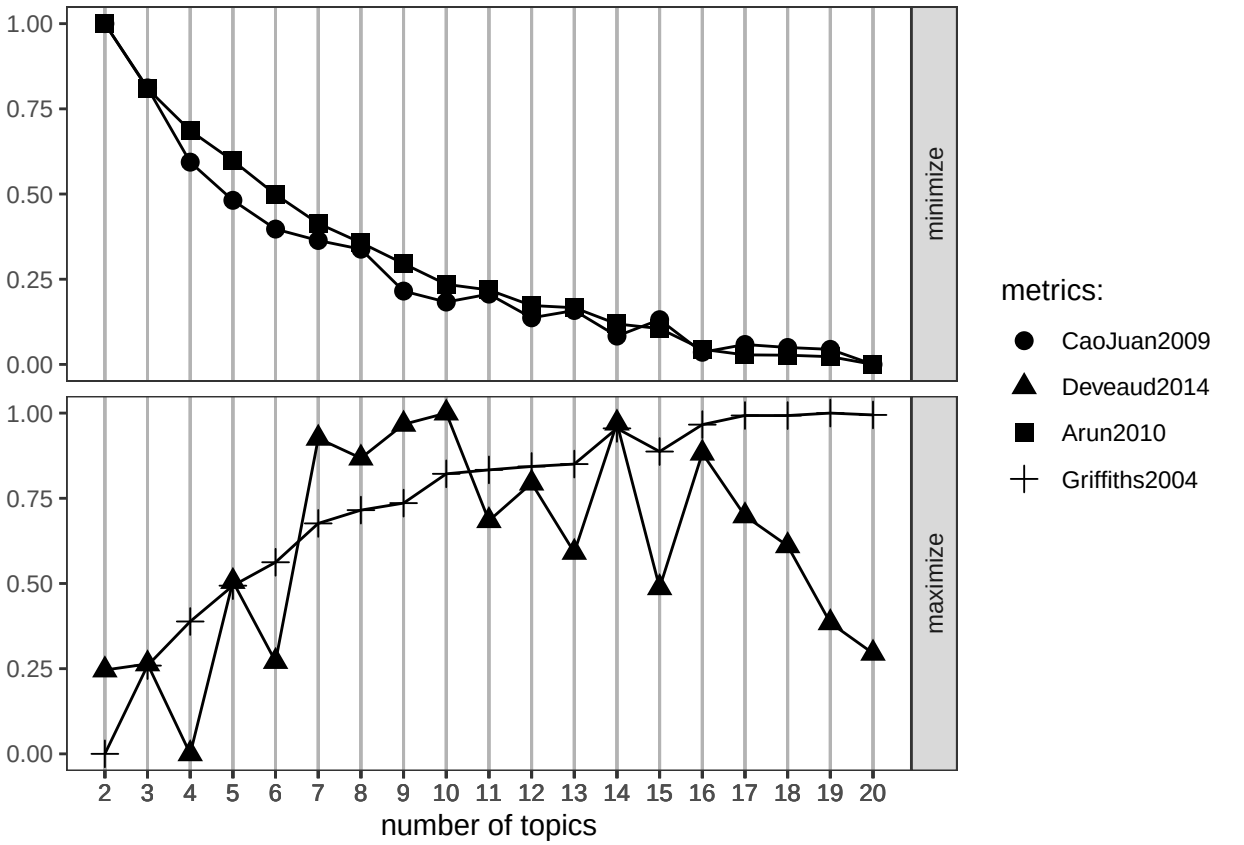
Find optimal number of topics

```
#
result <- FindTopicsNumber(
  dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014", "Arun2010", "Griffiths2004"), # can run up to 4 simultaneo
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)
```

```
## fit models... done.
## calculate metrics:
##   CaoJuan2009... done.
##   Deveaud2014... done.
##   Arun2010... done.
##   Griffiths2004... done.
```

```
FindTopicsNumber_plot(result)
```

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



Selecting number of latent topics present from the comment letters

9 Latent Topics

This is the original number of topics selected based on the **9 EPA priority areas**: Rulemaking, Permitting, Compliance and Enforcement, Science, States and Local Governments, Federal Agencies, Community-based Work, Tribes and Indigenous People, National Measures.

4 Latent Topics

The 2017-2022 EPA Environmental Justice Report's focus of 4 major goals/themes: (1) delivering environmental results; (2) cooperative federalism; (3) rule of law and fair process; and (4) building community capacity and engagement.

10 Latent topics

I choose 10 latent topics from results of the initial results of the k = 9 metrics where 10 was shown as the maximum number of

16 Latent topics

16 topics (9 priority + 7 additional)

```
# select number of topics
k1 <- 9
k4 <- 4
k10 <- 10
k16 <- 16

# run LDA model
topicModel_k9 <- LDA(dfm, k1, method="Gibbs", control=list(iter = 500, verbose = 25))
topicModel_k4 <- LDA(dfm, k4, method="Gibbs", control=list(iter = 500, verbose = 25))
topicModel_k10 <- LDA(dfm, k10, method="Gibbs", control=list(iter = 500, verbose = 25))
topicModel_k16 <- LDA(dfm, k16, method="Gibbs", control=list(iter = 500, verbose = 25))

#nTerms(dfm_comm)
tmResult9 <- posterior(topicModel_k9)
tmResult4 <- posterior(topicModel_k4)
tmResult10 <- posterior(topicModel_k10)
tmResult16 <- posterior(topicModel_k16)
# attributes(tmResult9)


# nTerms(dfm_comm)
# get beta from results
theta9 <- tmResult9$topics
theta4 <- tmResult4$topics
theta10 <- tmResult10$topics
theta16 <- tmResult16$topics

beta <- tmResult9$terms

# K distributions over nTerms(DTM) terms
# lengthOfVocab
dim(beta)
```

Let's pull out the top 10 likelihood / probability of frequency in each topic for each latent topic number. Remember, just because we choose those number of topics, it doesn't mean LDS actually picked up on the correct topic.

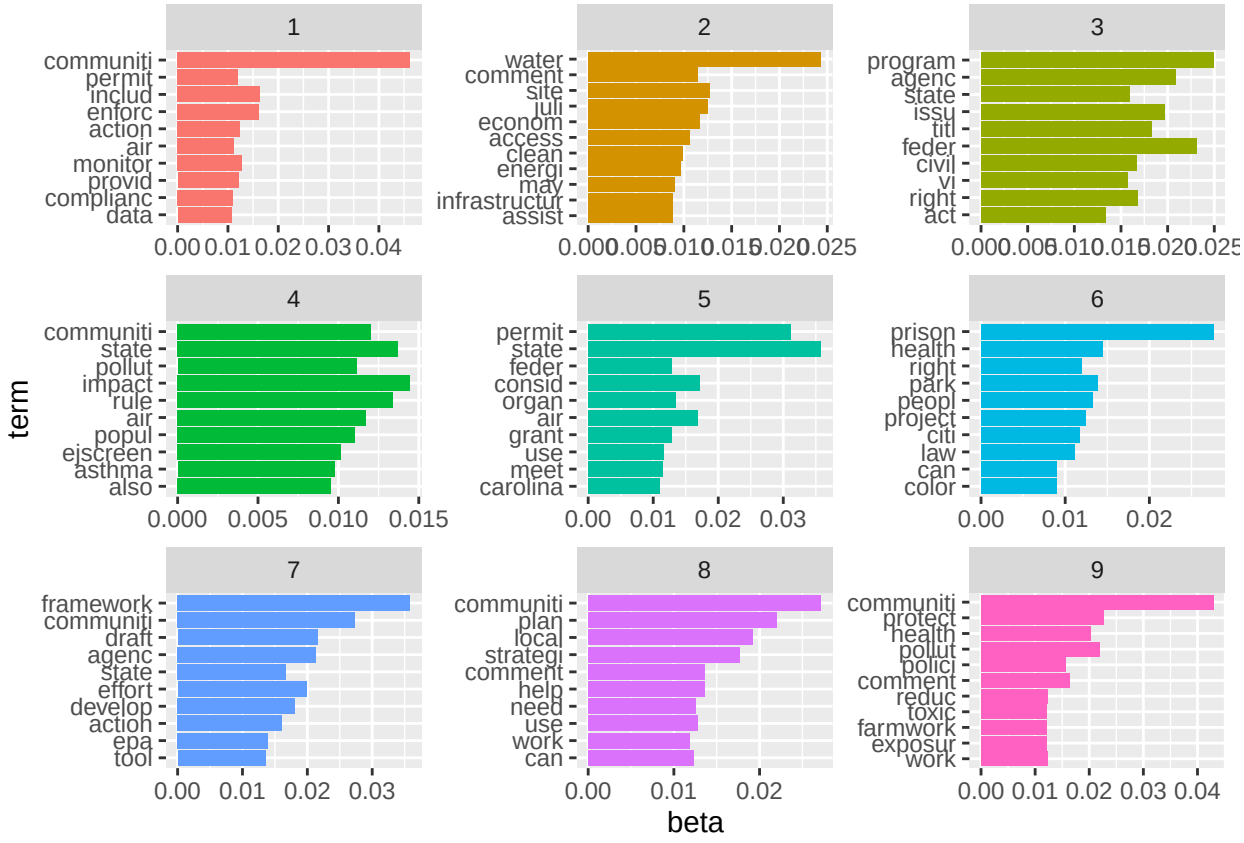
```
#terms(topicModel_k9, 10)
terms(topicModel_k4, 10)
```

```
##       Topic 1     Topic 2   Topic 3     Topic 4
##  [1,] "state"     "right"   "communiti" "communiti"
##  [2,] "permit"    "civil"   "pollut"    "framework"
##  [3,] "feder"     "peopl"   "health"    "action"
##  [4,] "consid"    "prison"  "air"       "local"
##  [5,] "comment"   "project" "impact"    "agenc"
##  [6,] "program"   "health"  "state"     "agenda"
##  [7,] "requir"    "vi"      "enforc"    "develop"
##  [8,] "polici"    "citi"    "also"      "comment"
##  [9,] "implement" "nation"  "agenc"     "draft"
## [10,] "draft"     "law"     "rule"      "plan"
```

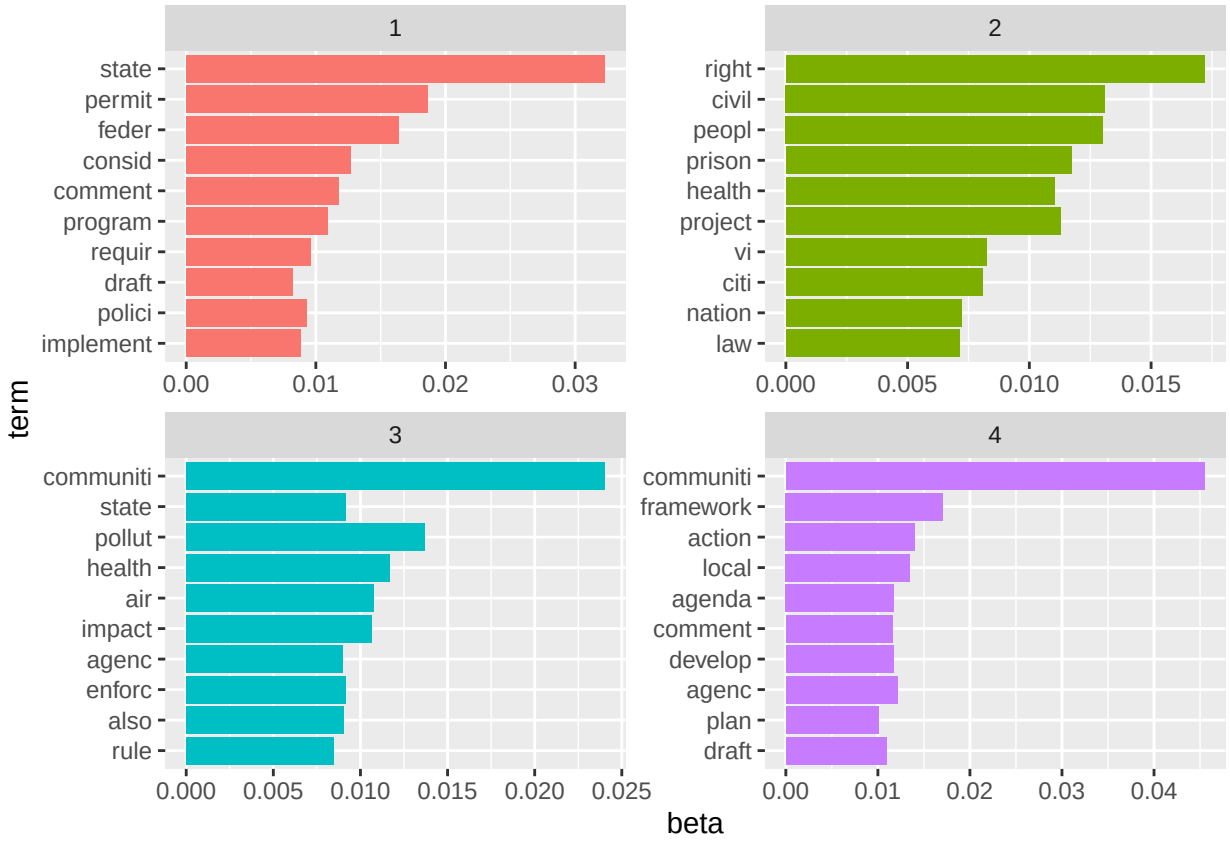
```
#terms(topicModel_k10, 10)
#terms(topicModel_k16, 10)
```

```
# tidy terms
comment_topics9 <- tidy(topicModel_k9, matrix = "beta")
comment_topics4 <- tidy(topicModel_k4, matrix = "beta")
comment_topics10 <- tidy(topicModel_k10, matrix = "beta")
comment_topics16 <- tidy(topicModel_k16, matrix = "beta")

top_terms9 <- comment_topics9 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

top_terms4 <- comment_topics4 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

top_terms10 <- comment_topics10 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

top_terms16 <- comment_topics16 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

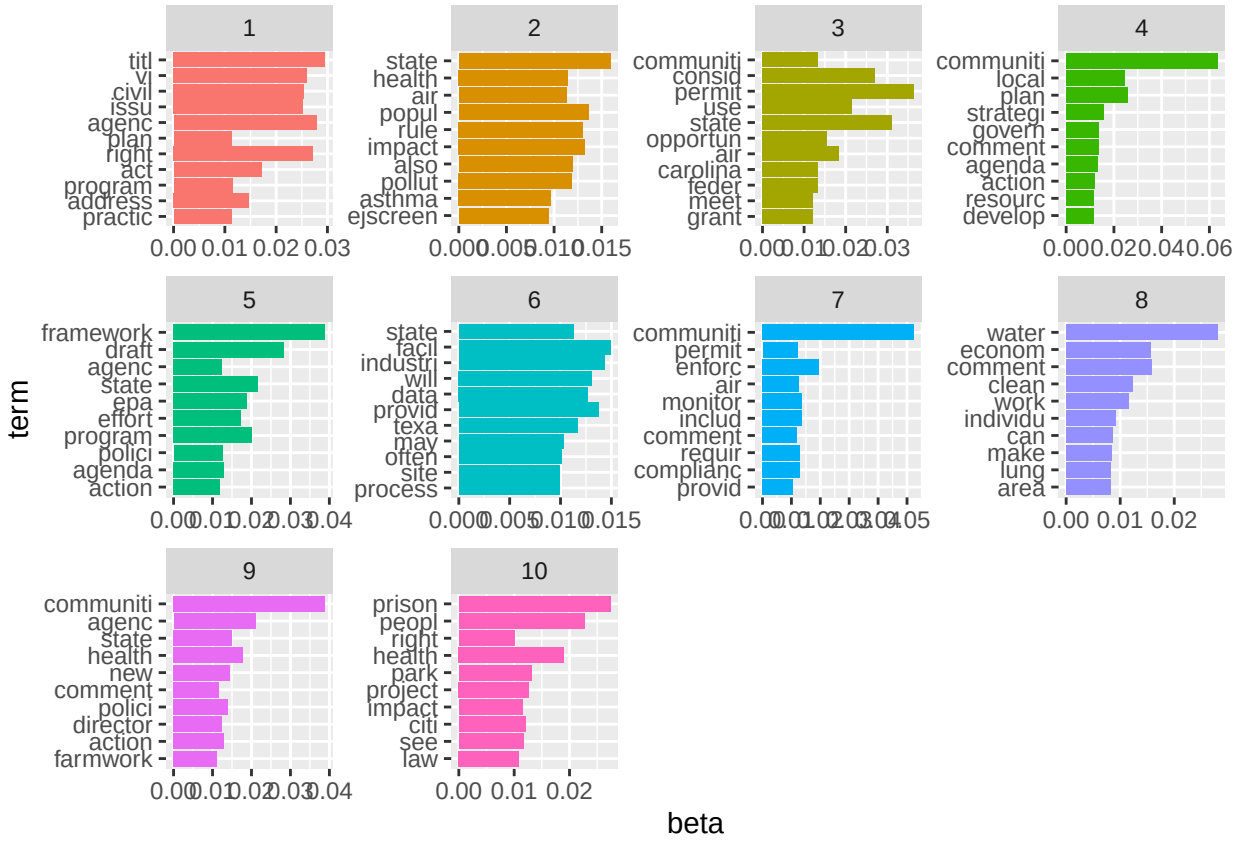
```
top_terms9
```



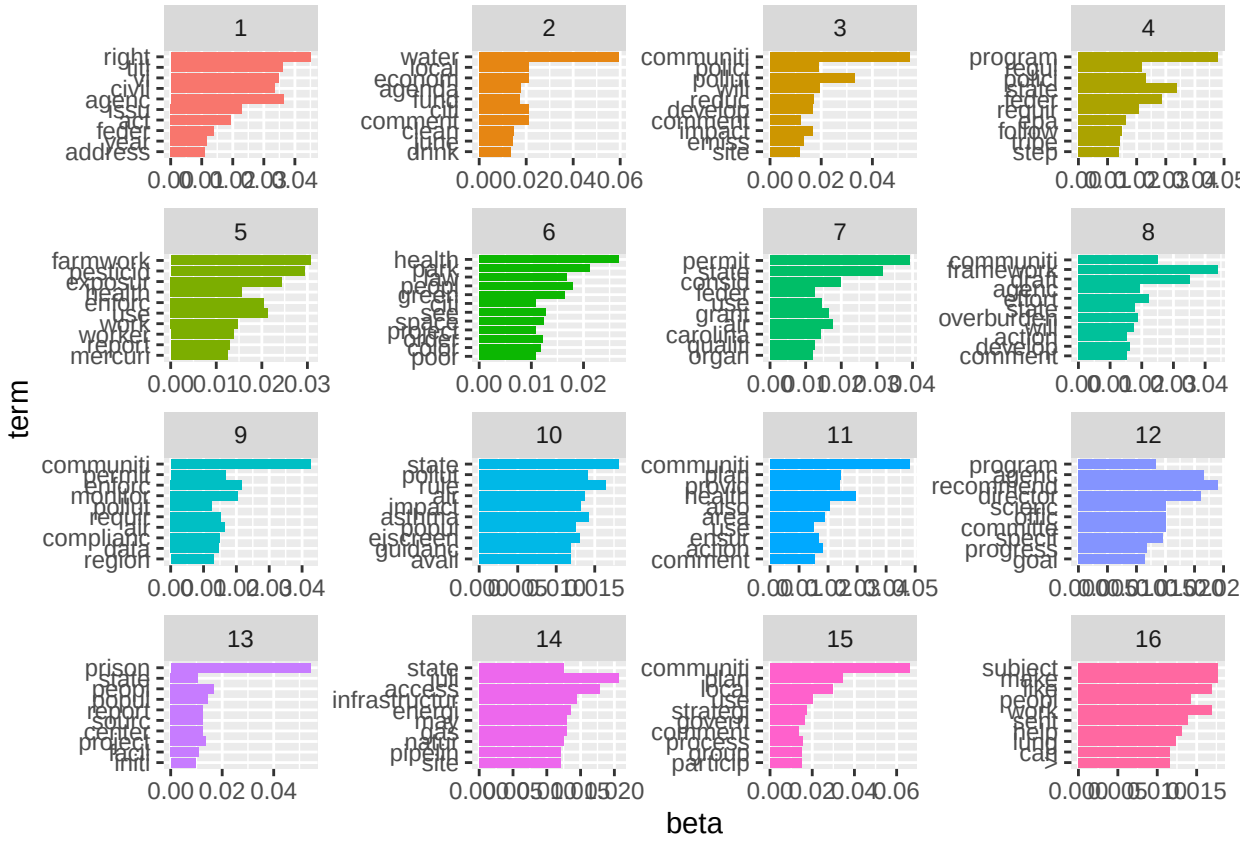

```
top_terms4
```


1
state
permit
feder
consid
comment
program
requir
draft
polici
implement
0.00
0.01
0.02
0.03
2
right
civil
peopl
prison
health
project
vi
citi
nation
law
0.000
0.005
0.010
0.015
3
communiti
state
pollut
health
air
impact
agenc
enforc
also
rule
0.000
0.005
0.010
0.015
0.020
0.025
4
communiti
framework
action
local
agenda
comment
develop
agenc
plan
draft
0.00
0.01
0.02
0.03
0.04
term
beta

```
top_terms10
```


1
titl
vi
civil
issu
agenc
plan
right
act
program
address
practic
0.00 0.01 0.02 0.03
2
state
health
air
popul
rule
impact
also
pollut
asthma
ejscreen
0.000 0.005 0.010 0.015
3
communiti
consid
permit
use
state
opportun
air
carolina
feder
meet
grant
0.00 0.01 0.02 0.03
4
communiti
local
plan
strategi
govern
comment
agenda
action
resourc
develop
0.000 0.020 0.040 0.060
5
framework
draft
agenc
state
epa
effort
program
polici
agenda
action
0.00 0.01 0.02 0.03 0.04
6
state
facil
industri
will
data
provid
texa
may
often
site
process
0.000 0.005 0.010 0.015
7
communiti
permit
enforc
air
monitor
includ
comment
requir
complianc
provid
0.00 0.01 0.02 0.03 0.04 0.05
8
water
econom
comment
clean
work
individu
can
make
lung
area
0.00 0.01 0.02
9
communiti
agenc
state
health
new
comment
polici
director
action
farmwork
0.00 0.01 0.02 0.03 0.04
10
prison
peopl
right
health
park
project
impact
citi
see
law
0.00 0.01 0.02
term
beta

```
top_terms16
```



Let's assign names to the topics so we know what we are working with. We can name them by their top terms. Let's reassign the topic names to the top 5 words per topic.

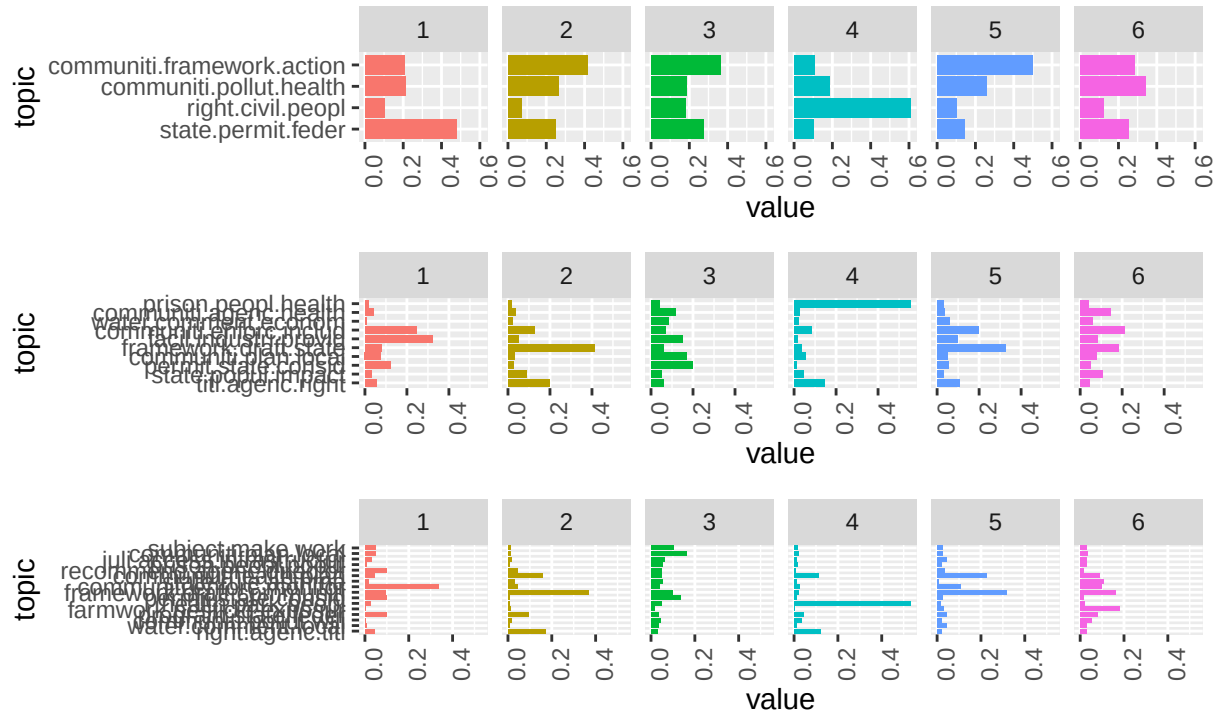
```
# re-assign names
top5termsPerTopic9 <- terms(topicModel_k9, 3)
top5termsPerTopic4 <- terms(topicModel_k4, 3)
top5termsPerTopic10 <- terms(topicModel_k10, 3)
top5termsPerTopic16 <- terms(topicModel_k16, 3)

# remove spaces
topicNames9 <- apply(top5termsPerTopic9, 2, paste, collapse=" ")
topicNames4 <- apply(top5termsPerTopic4, 2, paste, collapse=" ")
topicNames10 <- apply(top5termsPerTopic10, 2, paste, collapse=" ")
topicNames16 <- apply(top5termsPerTopic16, 2, paste, collapse=" ")
```

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

```
exampleIds <- c(1, 2, 3, 4, 5, 6)
N <- length(exampleIds)

# get topic proportions from example documents
topicProportionExamples9 <- theta9[exampleIds,]
colnames(topicProportionExamples9) <- topicNames9
vizDataFrame9 <- melt(cbind(data.frame(topicProportionExamples9), document=factor(1:N)), variable.name =

topicProportionExamples4 <- theta4[exampleIds,]
```

```
colnames(topicProportionExamples4) <- topicNames4
vizDataFrame4 <- melt(cbind(data.frame(topicProportionExamples4), document=factor(1:N)), variable.name =

topicProportionExamples10 <- theta10[exampleIds,]
colnames(topicProportionExamples10) <- topicNames10
vizDataFrame10 <- melt(cbind(data.frame(topicProportionExamples10), document=factor(1:N)), variable.name

topicProportionExamples16 <- theta16[exampleIds,]
colnames(topicProportionExamples16) <- topicNames16
vizDataFrame16 <- melt(cbind(data.frame(topicProportionExamples16), document=factor(1:N)), variable.name
```

```
# plot
terms9 <- ggplot(data = vizDataFrame9, 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)

terms4 <- ggplot(data = vizDataFrame4, 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) +
  theme(legend.position="none")

terms10 <- ggplot(data = vizDataFrame10, 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) +
  theme(legend.position="none")

terms16 <- ggplot(data = vizDataFrame16, 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) +
  theme(legend.position="none")

(terms4 / terms10 / terms16)
```




```
svd_tsne <- function(x){
  tsne(svd(x)$u)
}

json9 <- createJSON(
  phi = tmResult9$terms,
  theta = tmResult9$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: 11.0158365115072

## Epoch: Iteration #200 error is: 0.634158538655611

## Epoch: Iteration #300 error is: 0.291982844687135

## Epoch: Iteration #400 error is: 0.247914628557036

## Epoch: Iteration #500 error is: 0.237956935566942
```

```
## Epoch: Iteration #600 error is: 0.236047711259704

## Epoch: Iteration #700 error is: 0.23287305010911

## Epoch: Iteration #800 error is: 0.228036702790441

## Epoch: Iteration #900 error is: 0.226158470645397

## Epoch: Iteration #1000 error is: 0.226018231997731
```

```
json4 <- createJSON(
  phi = tmResult4$terms,
  theta = tmResult4$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: 13.6499680253603

## Epoch: Iteration #200 error is: 0.0487272675081749

## Epoch: Iteration #300 error is: 0.0487271972733961

## Epoch: Iteration #400 error is: 0.0487271972277124

## Epoch: Iteration #500 error is: 0.048727197219421

## Epoch: Iteration #600 error is: 0.0487271972062299

## Epoch: Iteration #700 error is: 0.048727197188409

## Epoch: Iteration #800 error is: 0.0487271971660766

## Epoch: Iteration #900 error is: 0.0487271971392923

## Epoch: Iteration #1000 error is: 0.0487271971080911
```

```
json10 <- createJSON(
  phi = tmResult10$terms,
  theta = tmResult10$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: 10.2848223323962

## Epoch: Iteration #200 error is: 0.268631048214639

## Epoch: Iteration #300 error is: 0.252883047485933

## Epoch: Iteration #400 error is: 0.251946664339446

## Epoch: Iteration #500 error is: 0.25193647949944

## Epoch: Iteration #600 error is: 0.25193622843396

## Epoch: Iteration #700 error is: 0.251936227750156

## Epoch: Iteration #800 error is: 0.251936227652158

## Epoch: Iteration #900 error is: 0.251936227511142

## Epoch: Iteration #1000 error is: 0.251936227325718
```

```
json16 <- createJSON(
  phi = tmResult16$terms,
  theta = tmResult16$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: 11.0305408215505

## Epoch: Iteration #200 error is: 0.920525111919077

## Epoch: Iteration #300 error is: 0.646144285597631

## Epoch: Iteration #400 error is: 0.471819766960372

## Epoch: Iteration #500 error is: 0.396656346366665

## Epoch: Iteration #600 error is: 0.381405702944236

## Epoch: Iteration #700 error is: 0.370002192854724

## Epoch: Iteration #800 error is: 0.365926505017761

## Epoch: Iteration #900 error is: 0.364274100770888

## Epoch: Iteration #1000 error is: 0.362498958370841
```

```
#serVis(json9)
#serVis(json4)
#serVis(json10)
#serVis(json16)
```