Topic Analysis

Shale

5/6/2022

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.
```

```
#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
```

```
toks <- tokens(epa_corp, remove_punct = TRUE, remove_numbers = TRUE)

# project-specific stop words
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, ]
#comments_df <- dfm[sel_idx, ]
```

Testing for Ideal `k`

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?

```
set.seed(25)
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"
```

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

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

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

```
##       Topic 1     Topic 2     Topic 3     Topic 4    Topic 5     Topic 6
##  [1,] "communiti" "communiti" "state"     "state"    "communiti" "framework"
##  [2,] "pollut"    "plan"      "permit"    "rule"     "enforc"    "draft"
##  [3,] "impact"    "local"     "feder"     "popul"    "monitor"   "effort"
##  [4,] "comment"   "particip"  "consid"    "provid"   "complianc" "agenc"
##  [5,] "protect"   "resourc"   "program"   "impact"   "includ"    "action"
##  [6,] "health"    "agenda"    "meet"      "health"   "action"    "state"
##  [7,] "result"    "engag"     "air"       "also"     "data"      "develop"
##  [8,] "air"       "use"       "opportun"  "asthma"   "requir"    "epa"
##  [9,] "polici"    "action"    "train"     "guidanc"  "report"    "agenda"
## [10,] "state"     "govern"    "implement" "ejscreen" "permit"    "will"
##       Topic 7   Topic 8   Topic 9
##  [1,] "work"    "agenc"   "prison"
##  [2,] "water"   "issu"    "peopl"
##  [3,] "comment" "right"   "project"
##  [4,] "subject" "civil"   "park"
##  [5,] "help"    "vi"      "law"
##  [6,] "need"    "titl"    "nation"
##  [7,] "make"    "includ"  "health"
##  [8,] "requir"  "program" "right"
##  [9,] "sent"    "feder"   "execut"
## [10,] "peopl"   "use"     "green"
```

Variation in Metrics

```r
# In class metrics
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.
```

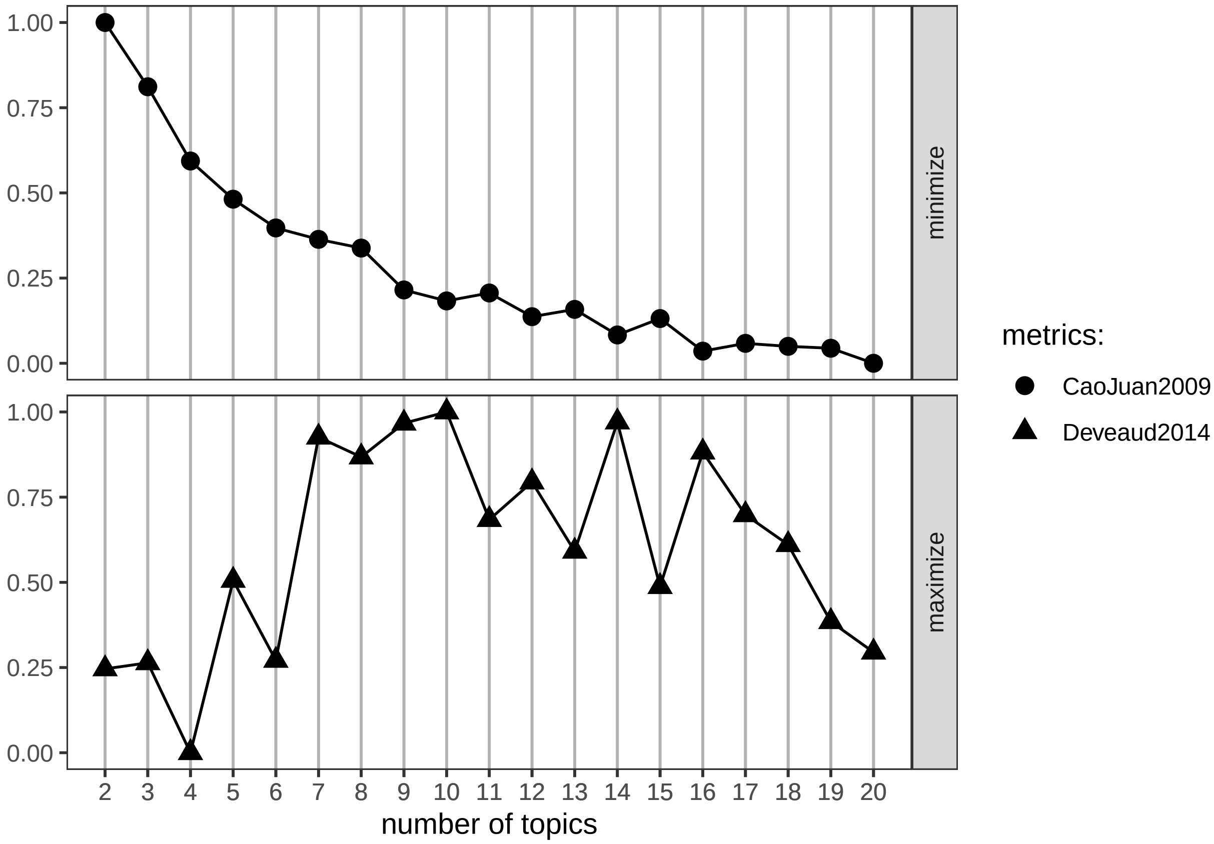
```r
# Griffiths/Arun
GA_topick <- FindTopicsNumber(
  dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("Griffiths2004",  "Arun2010"),
  method = "Gibbs",
```

```r
  control = list(seed = 77),
  verbose = TRUE
)
```

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

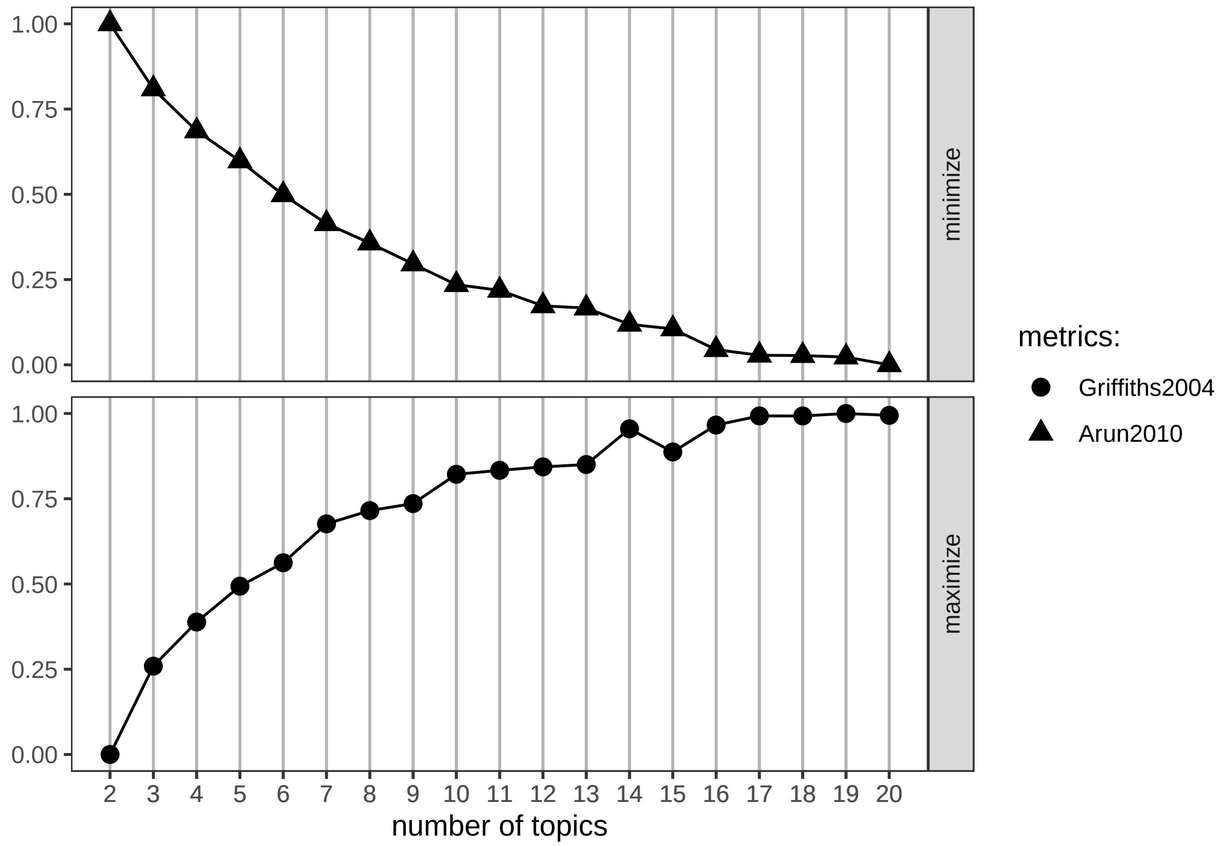
```r
FindTopicsNumber_plot(result)
```

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




```r
FindTopicsNumber_plot(GA_topick)
```

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



All metrics other than `Deveaud2014` are optimized by adding more and more topics. However, both `Deveaud2014` and `Griffiths2004` show a noticeable jump at 14. Based on this, additional models are run with 5, 10, and 14 topics.

```
set.seed(25)
k <- 14
topicModel_k7 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
```

```
## K = 14; 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" "communiti" "permit"     "work"     "program" "communiti"
##  [2,] "pollut"    "rule"      "state"      "subject"  "state"   "local"
##  [3,] "comment"   "health"    "consid"     "strategi" "polici"  "resourc"
##  [4,] "polici"    "state"     "air"        "sent"     "feder"   "govern"
##  [5,] "impact"    "asthma"    "feder"      "need"     "regul"   "particip"
##  [6,] "reduc"     "pollut"    "overburden" "help"     "tribe"   "social"
##  [7,] "air"       "popul"     "carolina"   "make"     "epa"     "group"
##  [8,] "new"       "impact"    "opportun"   "know"     "requir"  "collabor"
##  [9,] "power"     "air"       "grant"      "lung"     "order"   "juli"
## [10,] "state"     "avail"     "framework"  "tai"      "propos"  "agenda"
##       Topic 7     Topic 8     Topic 9     Topic 10  Topic 11    Topic 12
##  [1,] "communiti" "framework" "prison"    "plan"    "water"     "right"
##  [2,] "enforc"    "draft"     "project"   "comment" "effort"    "civil"
##  [3,] "monitor"   "agenc"     "facil"     "use"     "communiti" "agenc"
##  [4,] "permit"    "action"    "popul"     "action"  "comment"   "vi"
##  [5,] "data"      "develop"   "sourc"     "address" "framework" "titl"
##  [6,] "complianc" "state"     "mercuri"   "exampl"  "local"     "issu"
##  [7,] "air"       "communiti" "center"    "also"    "econom"    "act"
##  [8,] "report"    "tool"      "impact"    "includ"  "clean"     "feder"
##  [9,] "region"    "epa"       "incarcer"  "process" "lee"       "nation"
## [10,] "requir"    "effort"    "report"    "can"     "agenda"    "implement"
##       Topic 13   Topic 14
##  [1,] "health"   "health"
##  [2,] "work"     "park"
##  [3,] "farmwork" "peopl"
##  [4,] "pesticid" "citi"
##  [5,] "exposur"  "law"
##  [6,] "use"      "green"
##  [7,] "includ"   "project"
##  [8,] "enforc"   "space"
##  [9,] "worker"   "color"
## [10,] "state"    "includ"
```

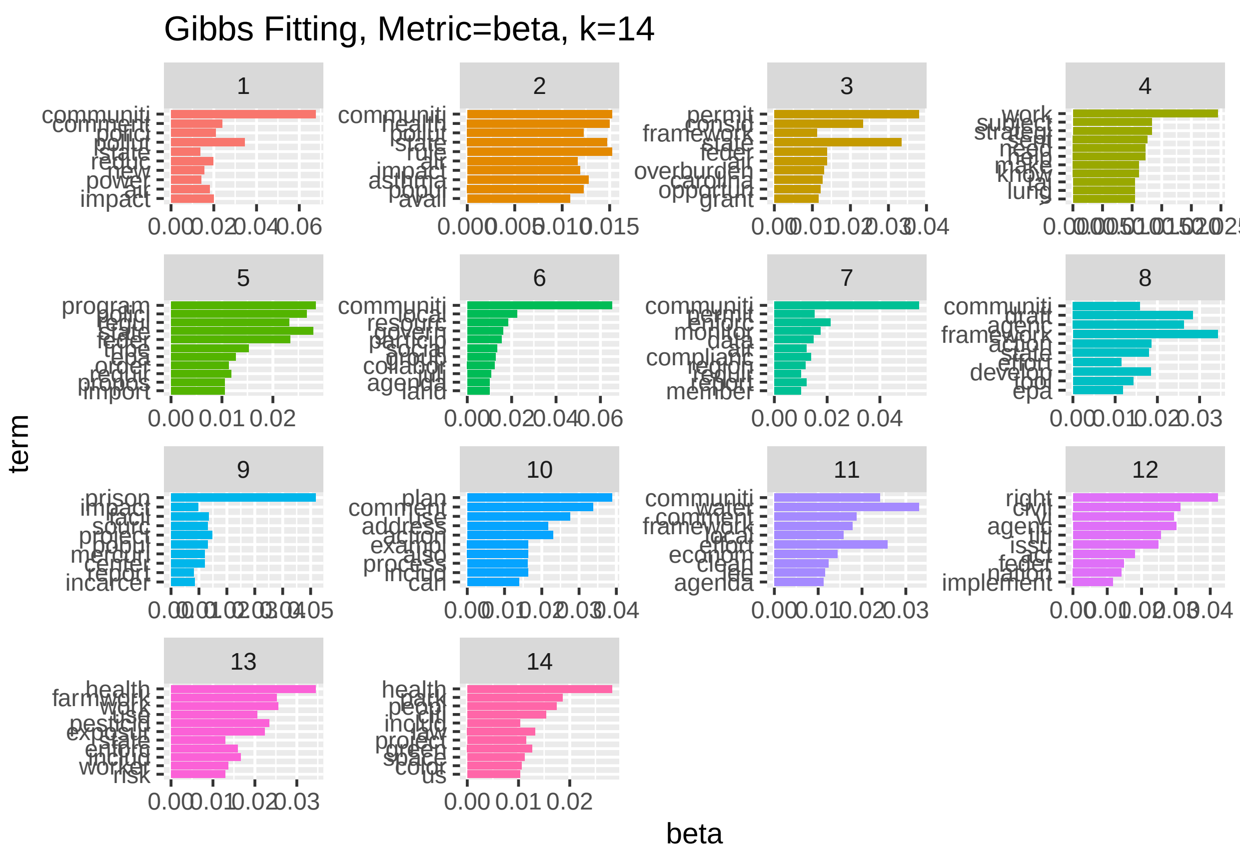
```
theta <- tmResult$topics
beta <- tmResult$terms
vocab <- (colnames(beta))


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: 146 x 3
```

```
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 communiti 0.0678
##  2     1 pollut    0.0345
##  3     1 comment   0.0241
##  4     1 polici    0.0209
##  5     1 impact    0.0199
##  6     1 reduc     0.0197
##  7     1 air       0.0180
##  8     1 new       0.0156
##  9     1 power     0.0141
## 10     1 state     0.0136
## # ... with 136 more rows
```

```r
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() +
  ggtitle(label = "Gibbs Fitting, Metric=beta, k=14")
```




```r
ggsave(path = here("plots"), filename = "topics14.png")
```

```
## Saving 6.5 x 4.5 in image
```

5 Topics:

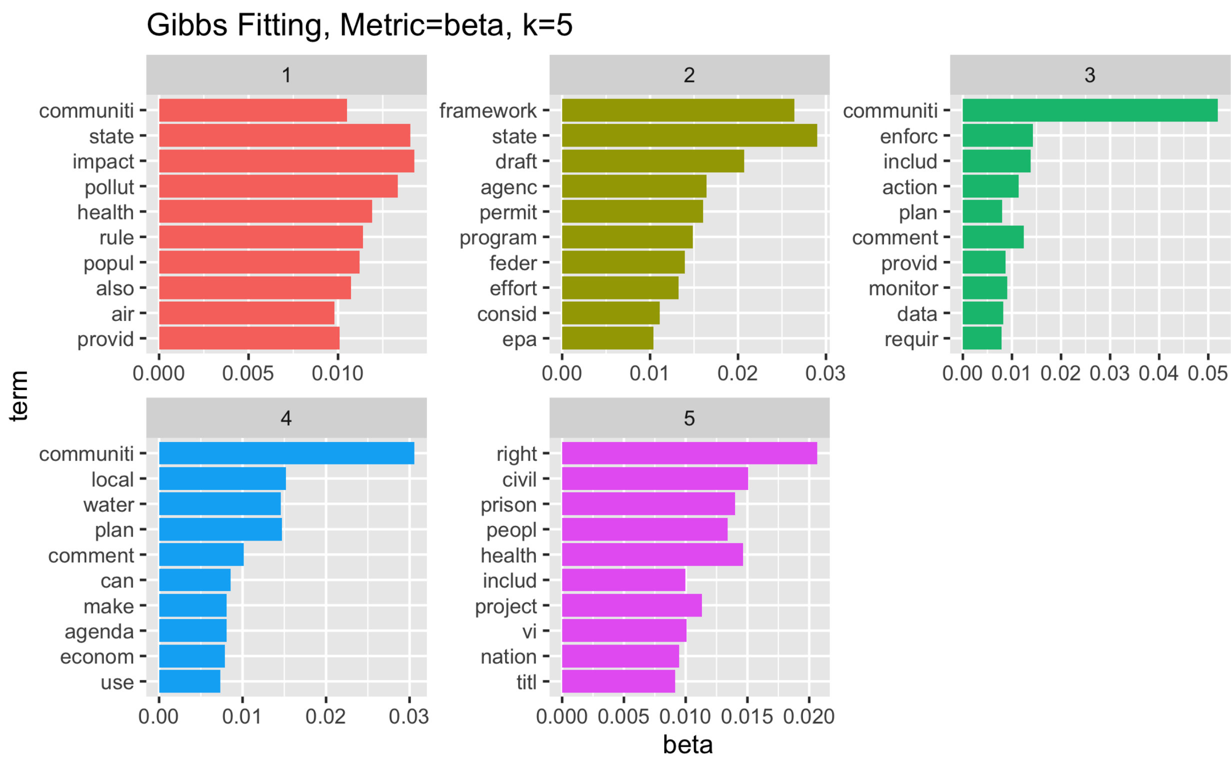


Figure 1: 5 topics

7 Topics:

10 Topics:

14 Topics:

Based on the distribution and overlap in these numbers of topics, I think I would choose either 5 or 10 topics depending on the audience. For a general audience, 5 topics is plenty to highlight the general areas of discussion in the documents: pollution/health, state and federal efforts, enforcement and monitoring, local planning, and title vi/civil rights. However, for a more technical or detailed audience, the 10 topic model splits into more detail while retaining enough distinction and meaning between topics (that, for example, the 14 topic model fails to achieve) that they can be useful categories.

Variation in Fitting Method

```
# Gibbs fitting method

set.seed(25)
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 ...
```

Figure 2: 7 topics

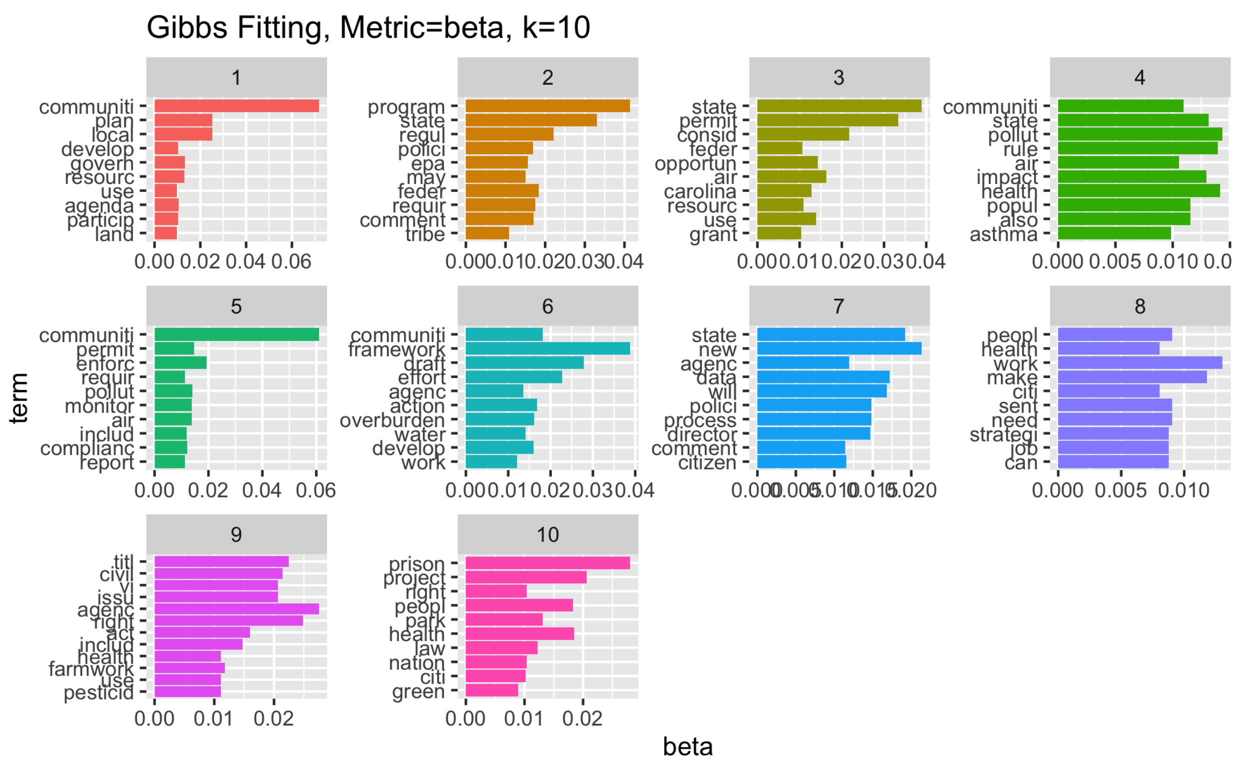


Figure 3: 10 topics

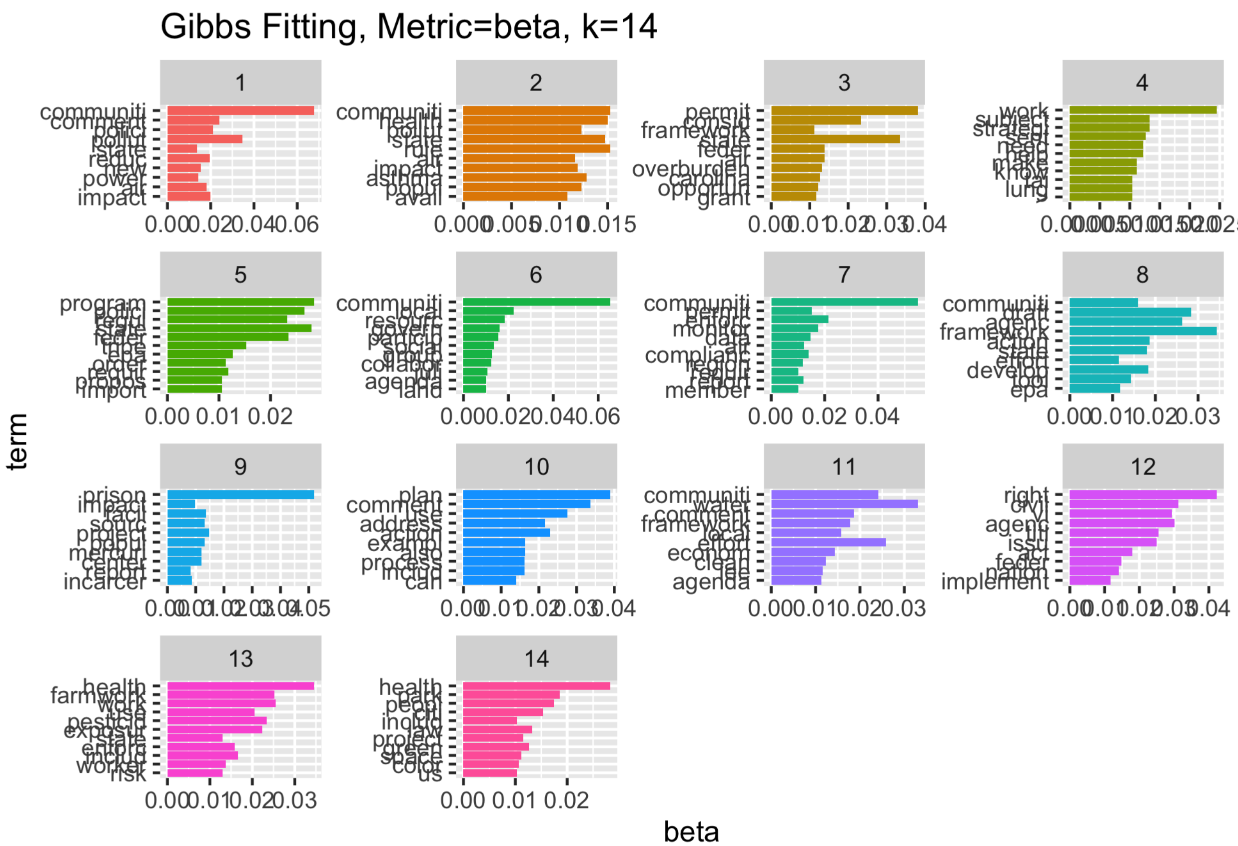


Figure 4: 14 topics

```
## 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" "health"    "state"     "communiti" "state"     "framework"
##  [2,] "plan"      "communiti" "impact"    "enforc"    "permit"    "draft"
##  [3,] "local"     "citi"      "rule"      "includ"    "consid"    "agenc"
##  [4,] "comment"   "can"       "pollut"    "comment"   "feder"     "state"
##  [5,] "agenda"    "park"      "popul"     "monitor"   "use"       "effort"
##  [6,] "use"       "econom"    "communiti" "complianc" "air"       "develop"
##  [7,] "particip"  "area"      "also"      "requir"    "implement" "program"
##  [8,] "action"    "see"       "health"    "health"    "organ"     "action"
##  [9,] "work"      "climat"    "air"       "action"    "qualiti"   "epa"
## [10,] "strategi"  "peopl"     "plan"      "data"      "comment"   "overburden"
##       Topic 7
##  [1,] "right"
##  [2,] "civil"
##  [3,] "prison"
##  [4,] "vi"
##  [5,] "titl"
##  [6,] "nation"
##  [7,] "feder"
##  [8,] "agenc"
##  [9,] "peopl"
## [10,] "impact"
```

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


# VEM fitting method

tModel_k7_vem <- LDA(dfm, k, method= "VEM")
tmResult_vem <- posterior(tModel_k7_vem)
terms(tModel_k7_vem, 10)
```

```
##       Topic 1     Topic 2     Topic 3     Topic 4     Topic 5     Topic 6
##  [1,] "communiti" "communiti" "right"     "communiti" "communiti" "state"
##  [2,] "prison"    "state"     "civil"     "framework" "comment"   "framework"
##  [3,] "plan"      "rule"      "communiti" "comment"   "agenc"     "draft"
##  [4,] "comment"   "impact"    "vi"        "water"     "pollut"    "permit"
##  [5,] "can"       "health"    "titl"      "local"     "state"     "communiti"
##  [6,] "peopl"     "pollut"    "agenc"     "effort"    "air"       "comment"
##  [7,] "use"       "popul"     "health"    "agenc"     "develop"   "program"
##  [8,] "state"     "air"       "park"      "impact"    "program"   "agenc"
##  [9,] "action"    "asthma"    "issu"      "action"    "will"      "feder"
## [10,] "health"    "also"      "includ"    "agenda"    "tool"      "consid"
##       Topic 7
##  [1,] "communiti"
##  [2,] "enforc"
##  [3,] "includ"
##  [4,] "health"
##  [5,] "air"
##  [6,] "monitor"
##  [7,] "comment"
##  [8,] "action"
##  [9,] "requir"
## [10,] "pollut"
```

```
theta_v <- tmResult_vem$topics
beta_v <- tmResult_vem$terms
vocab_v <- (colnames(beta_v))
```

There are multiple proposed methods for how to measure the best k value.

```
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 communiti 0.0432
##  2     1 plan      0.0212
##  3     1 local     0.0164
##  4     1 comment   0.0147
##  5     1 agenda    0.0140
##  6     1 use       0.0138
##  7     1 particip  0.0114
##  8     1 action    0.0110
##  9     1 work      0.0110
## 10     1 strategi  0.0103
## # ... with 61 more rows
```

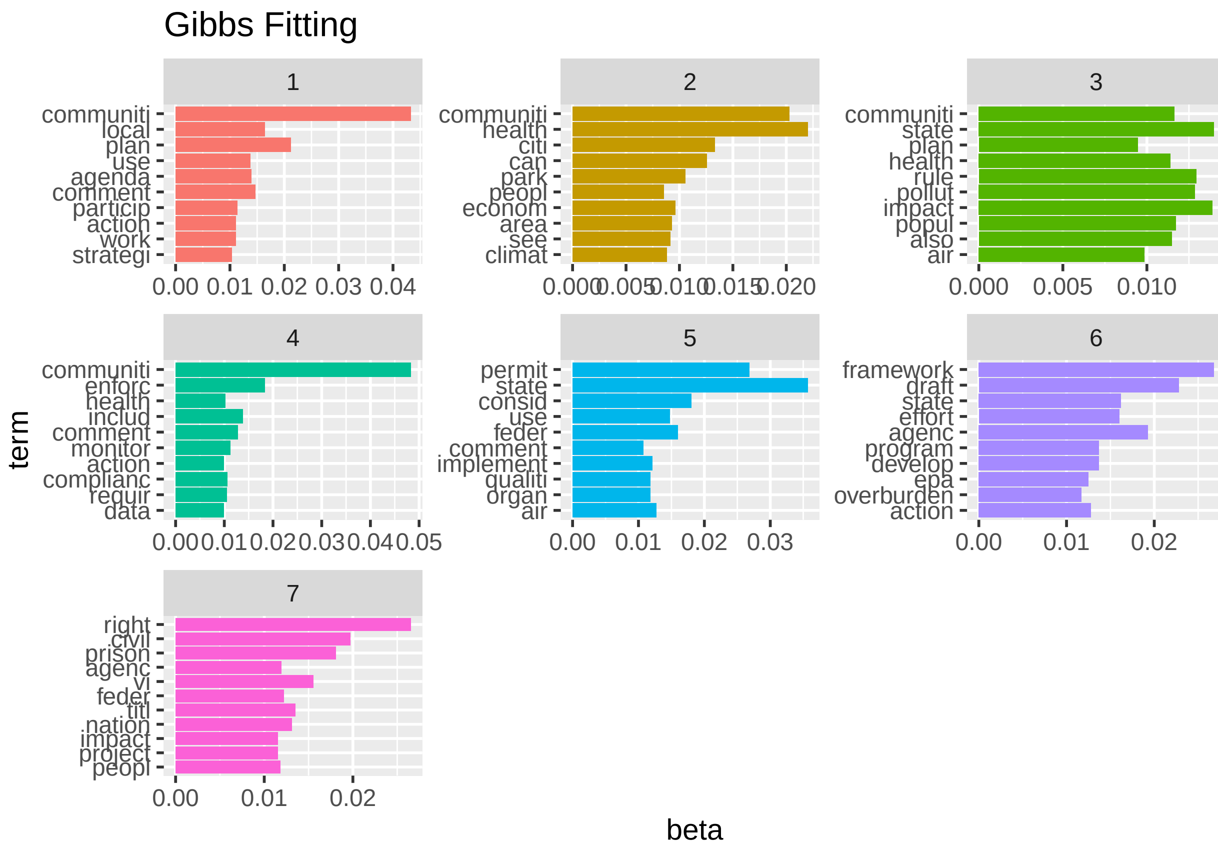
```
# for VEM fitting (note prison stuff)

ct_vem <- tidy(tModel_k7_vem, matrix = "beta")
top_terms_v <- ct_vem %>%
```

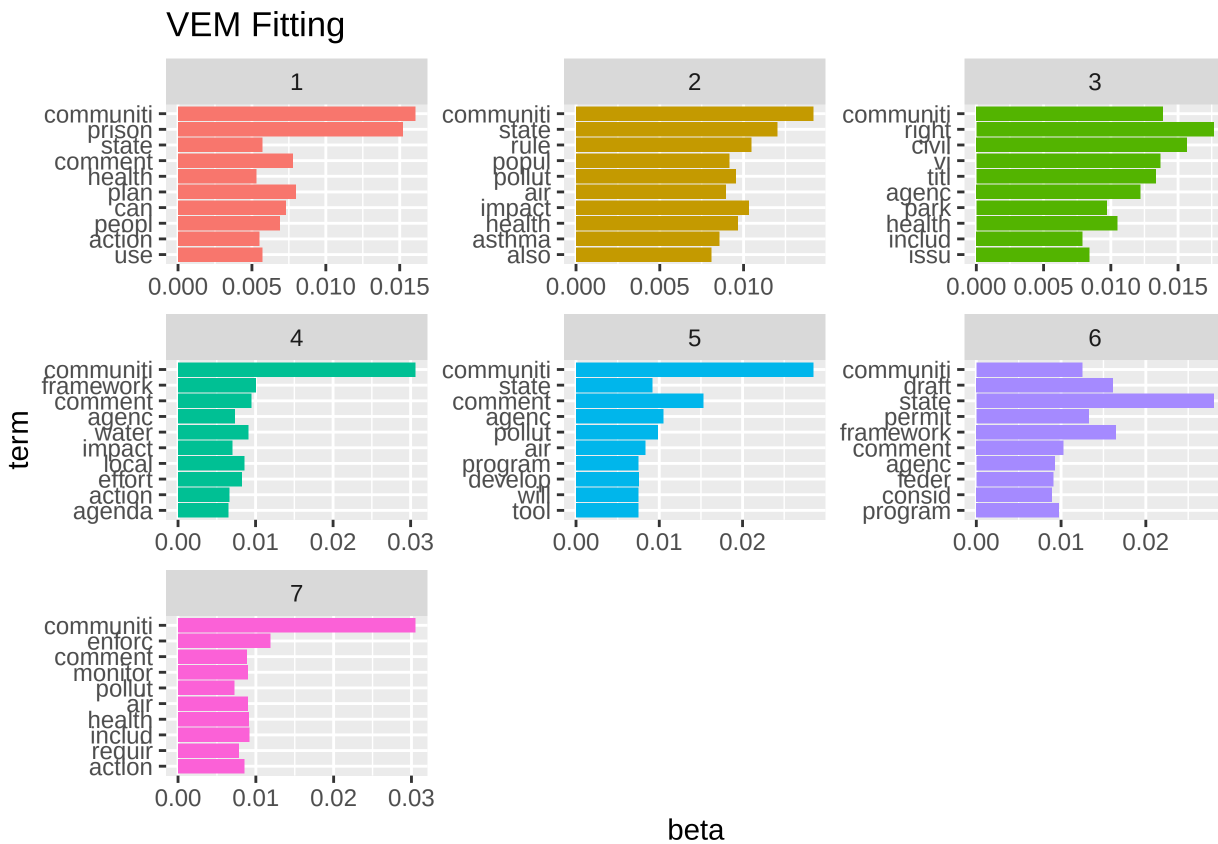
```r
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top_terms_v
```

```
## # A tibble: 70 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 communiti 0.0161
##  2     1 prison    0.0152
##  3     1 plan      0.00798
##  4     1 comment   0.00777
##  5     1 can       0.00730
##  6     1 peopl     0.00689
##  7     1 use       0.00570
##  8     1 state     0.00570
##  9     1 action    0.00551
## 10     1 health    0.00529
## # ... with 60 more rows
```

```r
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() +
  ggtitle(label = "Gibbs Fitting")
```




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



Based on a comparison of the top terms in each topic (7 topics) using `Gibbs` and `VEM` fitting methods, it seems like `Gibbs` provides more useful separations (note, for example, that `VEM`lists the term 'communiti'[es] in the top 10 terms in every group, so distinctiveness is not very high). This is using the `beta` metric.