Country Music Project

Matt Shu

2022-04-27

Prereqs

```
library(igraph)
library(tidyverse)
library(stm)
library(RSQLite)
library(RecordLinkage)
library(stringdist)
```

Preprocessing

```
conn <- dbConnect(RSQLite::SQLite(), "files/22-04-21-playback-fm-top-country.db")
dfSongs <- dbGetQuery(conn, 'SELECT * FROM lyrics')
dbDisconnect(conn)
```

Dataset Visualizations

```
cleaned_df <-dfSongs %>%
  # first, remove observation with missing values of the meta variables
  filter(!is.na(lyrics))  %>%
  # first, remove observation with missing values of the meta variables
  filter(!is.na(artist))  %>%
  as.data.frame()
cleaned_df$lyrics <- str_replace_all(cleaned_df$lyrics,"[\\s]+", " ")
```

Create Artist ID Hash

```
cleaned_df$artist_id <- as.character(as.numeric(as.factor(cleaned_df$artist)))
cleaned_df$song_id <- as.character((10000 + as.numeric(as.factor(cleaned_df$track))))
```

Filter out Mismatches

```
dim(cleaned_df)
```

```
## [1] 7094   10
```

```
cleaned_df$cleaned_lyrics <-
  str_replace_all(cleaned_df$lyrics, 'Chap\\. [0-9]', NA_character_) %>%
  str_replace_all(., 'Listening Log', NA_character_) %>%
  str_replace_all(., 'Favorite Songs Of', NA_character_) %>%
  str_replace_all(., 'Chapter [0-9]', NA_character_) %>%
```

```
  str_replace_all(., 'New Music ', NA_character_) %>%
  str_replace_all(., 'Nominees', NA_character_) %>%
  str_replace_all(., 'Best Songs of ', NA_character_) %>%
  str_replace_all(., "[0-9]+ U S", NA_character_) %>% # Court Cases
  str_replace_all(., "[0-9]+ U.S", NA_character_) %>% # Court Cases
  # keep only alphabet letters and numbers ("al" and "num")
  str_replace_all(., "[^[:alnum:]]", " ") %>%
  # make multiple spaces into one space
  str_replace_all(.,"[ ]+", " ") %>%
  str_replace(., ".*Lyrics", "")
cleaned_df <- cleaned_df %>%
  filter(!is.na(cleaned_lyrics)) %>%
  filter(levenshteinSim(track, str_match(lyrics, "(.*)Lyrics")[,2]) > .5) %>% # There are some false po
  as.data.frame()
dim(cleaned_df)
```

```
## [1] 6371   11
```

Preprocessing (and STM exploration)

```
# Dataframe containing the text
docs_df <- cleaned_df %>%
   dplyr::select(track_id, cleaned_lyrics) %>%
   # first, remove observation with missing values of the meta variables
   filter(!is.na(cleaned_lyrics))  %>%
   # the objects need to be class "data frame"
   as.data.frame()
```

```
# Dataframe containing (sample) documents' metadata of interest
meta_df <- cleaned_df %>%
   dplyr::select(track_id, rank, artist, track, year) %>%
   # the objects need to be class "data frame"
   as.data.frame()
```

```
processed_docs_1 <- textProcessor(documents = docs_df$cleaned_lyrics,
                                  metadata = meta_df,
                                  lowercase = TRUE,
                                  removestopwords = TRUE,
                                  removenumbers = TRUE,
                                  removepunctuation = TRUE,
                                  ucp = TRUE,
                                  stem = TRUE,
                                  striphtml = TRUE,
                                  wordLengths = c(3, Inf),
                                  language = "en")
```

```
meta <- processed_docs_1$meta
vocab <- processed_docs_1$vocab
docs <- processed_docs_1$documents
keep <- !is.na(meta$artist) && !is.na(meta$rank)
meta <- meta[keep,]
docs <- docs[keep]
```

```
prepped_data <- prepDocuments(docs,
                              vocab,
```

```
                            meta,
                            # the lower threshold value means that only words
                            # that appear more times than the value (in this
                            # example the value = 3) will be retained; this is
                            # another researcher decision
                            lower.thresh = 2)
```

Old code for removing unusual mismatch with no words despite past filters

```
length(docs_df$cleaned_lyrics) # original documents
length(prepped_data$meta$track_id) # off from the preceding count
dif <- setdiff(docs_df$track_id, # original vector of documents
               prepped_data$meta$track_id) # list of documents after prepDocuments
tmp <- docs_df
tmp2 <- tmp[!tmp$track_id %in% dif,]
tmp_doc <- tmp2 %>%
  select(track_id, cleaned_lyrics)
length(tmp_doc$track_id)
length(prepped_data$meta$track_id)

# View the track ids that were removed for some reason (often other language)
tmp3 <- tmp[tmp$track_id %in% dif,]
tmp3
```

See Cleaned Sample!

```
head(cleaned_df)
```

```
##   index track_id year             artist                                track
## 1     0        0 1944          Red Foley                   Smoke On The Water
## 2     1        1 1944 The King Cole Trio          Straighten Up And Fly Right
## 3     2        2 1944       Louis Jordan Is You Is or Is You Ain't (Ma' Baby)
## 4     4        4 1944       Louis Jordan                          Ration Blues
## 5     5        5 1944        Ernest Tubb                Soldier's Last Letter
## 6     8        8 1944        Ernest Tubb                 Try Me One More Time
##   rank
## 1    1
## 2    2
## 3    3
## 4    5
## 5    6
## 6    9
##                                                                    link
## 1                   /charts/country/video/1944/red-foley-smoke-on-the-water
## 2 /charts/country/video/1944/the-king-cole-trio-straighten-up-and-fly-right
## 3  /charts/country/video/1944/louis-jordan-is-you-is-or-is-you-aint-ma-baby
## 4                      /charts/country/video/1944/louis-jordan-ration-blues
## 5               /charts/country/video/1944/ernest-tubb-soldiers-last-letter
## 6               /charts/country/video/1944/ernest-tubb-try-me-one-more-time
##
## 1
## 2
## 3 Is You Is Or Is You Ain’t (ma Baby) LyricsBing Crosby Miscellaneous Is You Is Or Is You Ain't (ma
## 4
## 5
```

```
## 6
##   artist_id song_id
## 1       805   14521
## 2       958   14747
## 3       673   12721
## 4       673   14080
## 5       335   14556
## 6       335   15559
##
## 1
## 2
## 3 Bing Crosby Miscellaneous Is You Is Or Is You Ain t ma Baby Is You Is Or Is You Ain t Ma Baby Bing
## 4
## 5
## 6
```

Find K

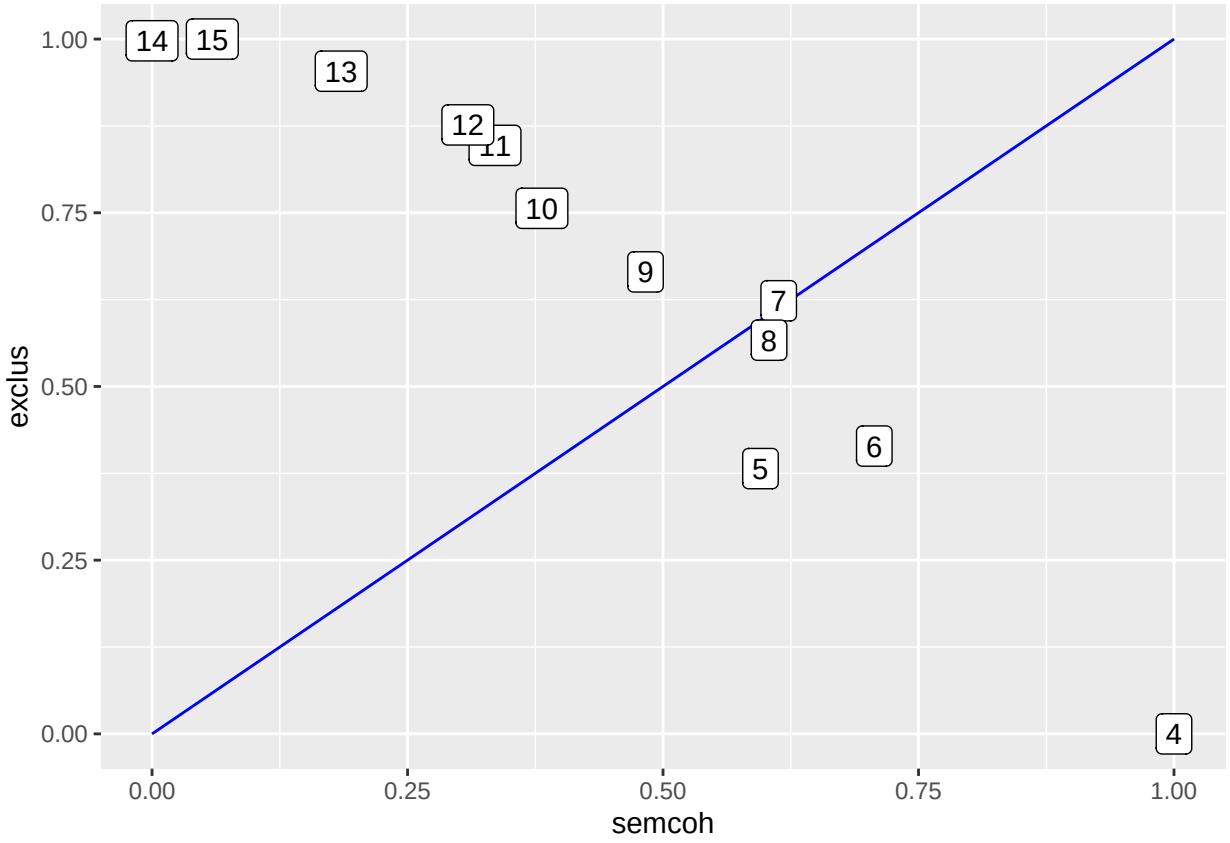
```
k_seq = seq(4, 15, 1)
```

```
## You can "watch" the algorithm model topics in the console
searched = searchK(prepped_data$documents,
                   prepped_data$vocab,
                   K = k_seq,
                   data = prepped_data$meta,
                   seed = 183654)
# saveRDS(searched, file = "22-04-22-searchK-4-15.RData")
```

Show K

```
searched <- readRDS("22-04-22-searchK-4-15.RData")
# Get values from `searchK` output
semcoh <- unlist(searched$results$semcoh)
exclus <- unlist(searched$results$exclus)

# Max/min semantic cohesion
max_sc <- max(semcoh)
min_sc<-min(semcoh)

# Max/min exclusivity
max_ex<-max(exclus)
min_ex<-min(exclus)

# Min-max normalization is (value - min)/(max - min)
x_vals <- (semcoh-min_sc)/(max_sc-min_sc)
y_vals <- (exclus-min_ex)/(max_ex-min_ex)
# add semantic cohesion and exclusivity together weighted evenly
ids = k_seq
search_plot_df <- tibble(id = ids,
                   semcoh = x_vals,
                   exclus = y_vals,
                   combine = x_vals*0.5 + y_vals*0.5)

# Plot
```

```
ggplot(search_plot_df, mapping = aes(x = semcoh, y = exclus)) +
  xlim(0,1) +
  ylim(0,1) +
  annotate("segment", x = 0, xend = 1, y = 0, yend = 1, color = "blue") +
  geom_label(aes(label=id))
```



Model Work

```
# 6 topics seems to also work nice, with a strong "Country" category
num_topics <- 7 # Chosen after above search and some playing around
out_covariates_7 <- stm(prepped_data$documents,
                        prepped_data$vocab,
                        K = num_topics,
                        prevalence = ~ rank + year,
                        max.em.its = 500,
                        data = prepped_data$meta,
                        seed = 592669)
```

```
terms = labelTopics(out_covariates_7, n = 10)
terms$prob # rows are topics; columns are most probable words (in order)
```

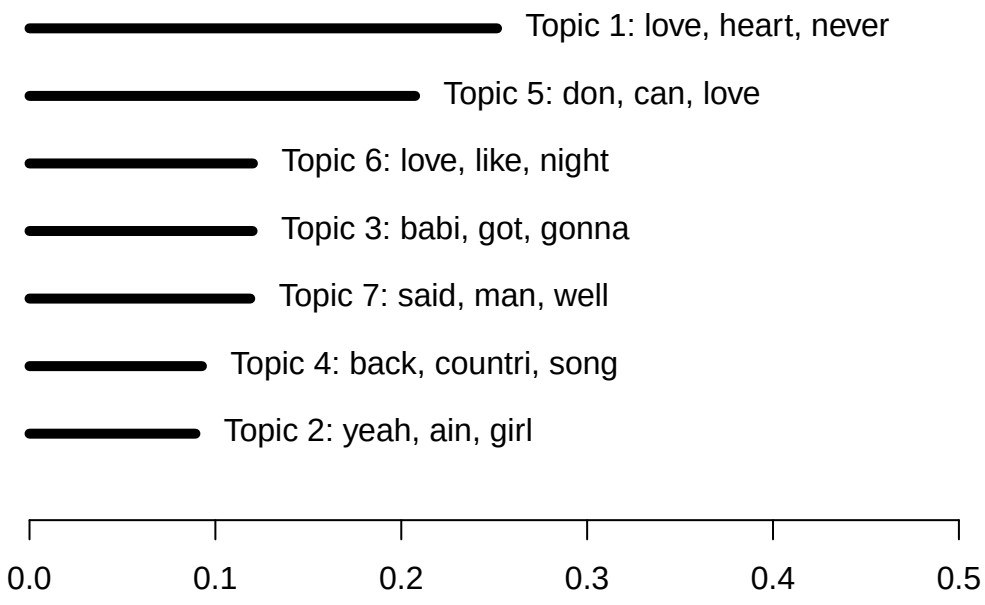
```
##      [,1]   [,2]      [,3]    [,4]   [,5]    [,6]    [,7]    [,8]    [,9]
## [1,] "love" "heart"   "never" "one"  "time"  "now"   "say"   "still" "just"
## [2,] "yeah" "ain"    "girl"  "like" "good"  "got"   "man"   "just"  "littl"
## [3,] "babi" "got"    "gonna" "time" "one"   "littl" "come"  "night" "now"
## [4,] "back" "countri" "song"  "roll" "get"   "old"   "road"  "town"  "like"
## [5,] "don"  "can"    "love"  "know" "want"  "let"   "just"  "make"  "like"
```

```
## [6,] "love" "like"    "night" "day"  "dream" "eye"   "blue"  "sweet" "rain"
## [7,] "said" "man"     "well"  "old"  "home"  "big"   "daddi" "boy"   "mama"
##      [,10]
## [1,] "will"
## [2,] "ooh"
## [3,] "right"
## [4,] "rock"
## [5,] "feel"
## [6,] "heaven"
## [7,] "just"
```

```
terms$frex # rows are topics; columns are most FREX words (in order)
```

```
##      [,1]     [,2]      [,3]         [,4]    [,5]     [,6]     [,7]
## [1,] "cri"    "goodby"  "fool"       "true"  "tear"   "hurt"   "lie"
## [2,] "ooh"    "boo"     "huh"        "gimm"  "yeah"   "whoa"   "girl"
## [3,] "bye"    "honki"   "tonk"       "honey" "shake"  "babi"   "drinkin"
## [4,] "boogi"  "countri" "hillbilli" "jone"  "santa"  "cowboy" "crank"
## [5,] "want"   "need"    "don"        "hold"  "let"    "feel"   "easi"
## [6,] "heaven" "angel"   "sail"       "shine" "wing"   "sea"    "rain"
## [7,] "mom"    "dad"     "hero"       "wife"  "twenti" "daddi"  "famili"
##      [,8]      [,9]     [,10]
## [1,] "memori"  "darl"   "still"
## [2,] "lovin"   "bit"    "woah"
## [3,] "thinkin" "gotta"  "batter"
## [4,] "cha"     "claus"  "tractor"
## [5,] "fall"    "enough" "give"
## [6,] "sky"     "storm"  "fli"
## [7,] "blah"    "father" "momma"
```

```
par(bty="n",lwd=5)
plot(out_covariates_7,
     type = "summary",
     main = "Prevalence of topics")
```




```
docs_examples_covar <- findThoughts(out_covariates_7,
                                    texts = tmp_doc$track_id,
                                    n = 10,
                                    topics = c(1:num_topics))

for(topic_num in c(1:num_topics)) {
  print(paste("Topic ", topic_num))
  for(track in docs_examples_covar$docs[[topic_num]]) {
    print(cleaned_df$track[cleaned_df$track_id == track])
  }
  print("")
}
```

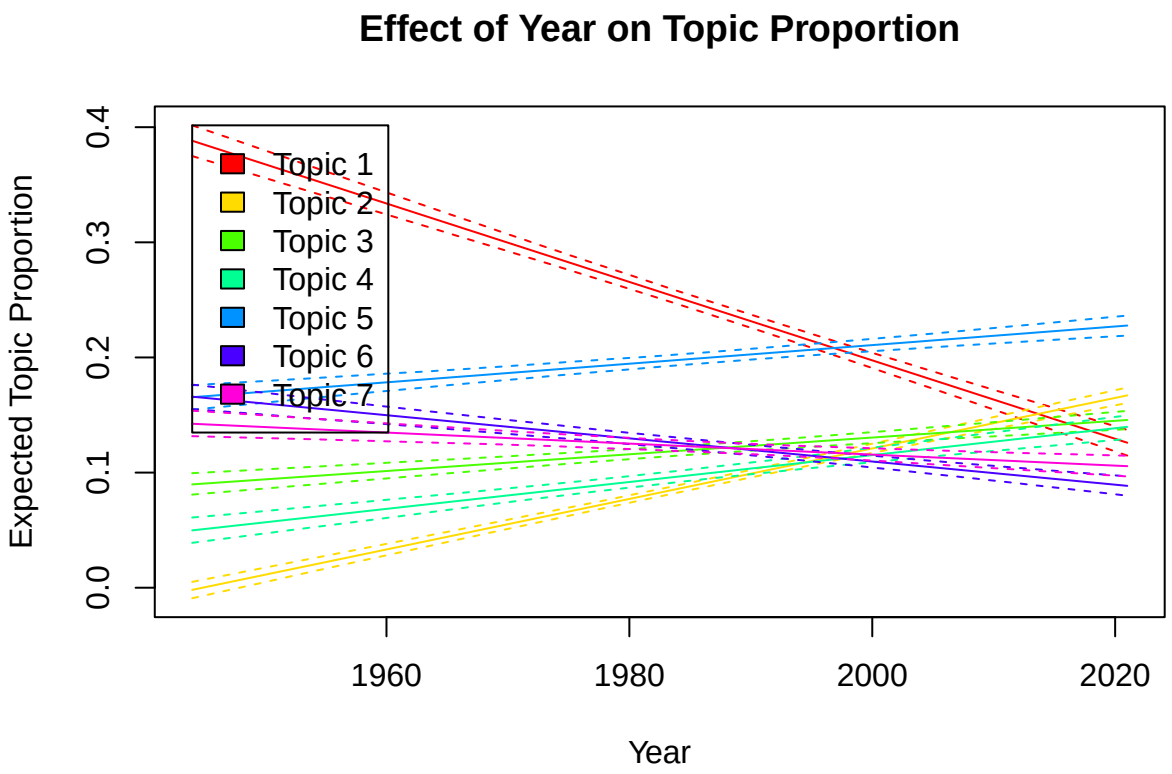
```
## [1] "Topic  1"
## [1] "Something Old, Something New"
## [1] "One Promise Too Late"
## [1] "Sweetheart You Done Me Wrong"
## [1] "All Alone in This World without You"
## [1] "Fool Fool Fool"
## [1] "Am I Losing You"
## [1] "Happy Journey"
## [1] "Am I Losing You"
## [1] "When You Are Lonely"
## [1] "Is It Wrong (For Loving You)"
## [1] ""
## [1] "Topic  2"
## [1] "Desperate Man"
## [1] "Gimmie That Girl"
## [1] "Gimmie That Girl"
## [1] "Just The Way"
```

```
## [1] "Just The Way"
## [1] "Just the Way"
## [1] "You Broke Up with Me"
## [1] "You Broke Up with Me"
## [1] "Uh-Huh--Mm"
## [1] "Uh-Huh-mm"
## [1] ""
## [1] "Topic  3"
## [1] "Swing"
## [1] "Honky Tonkin'"
## [1] "Heartache Medication"
## [1] "Heartache Medication"
## [1] "Honky Tonkin'"
## [1] "Trademark"
## [1] "Penny Arcade"
## [1] "If You've Got The Money I've Got The Time"
## [1] "If You've Got The Money I've Got The Time"
## [1] "It's A Little Too Late"
## [1] ""
## [1] "Topic  4"
## [1] "Teenage Boogie"
## [1] "Redneck Yacht Club"
## [1] "Cincinnati Dancing Pig"
## [1] "The Rhumba Boogie"
## [1] "Long Live"
## [1] "She Cranks My Tractor"
## [1] "Ragtime Cowboy Joe"
## [1] "Hula Rock"
## [1] "Mule Train"
## [1] "Pan American Boogie"
## [1] ""
## [1] "Topic  5"
## [1] "Don't Be Stupid (You Know I Love You)"
## [1] "Don't Be Stupid (You Know I Love You)"
## [1] "I Can't Get Close Enough"
## [1] "I Can't Get Close Enough"
## [1] "Losing Sleep"
## [1] "Losing Sleep"
## [1] "Love Lessons"
## [1] "It Matters To Me"
## [1] "It Matters to Me"
## [1] "Fall Into Me"
## [1] ""
## [1] "Topic  6"
## [1] "Ring Of Fire"
## [1] "Sweet Summer Lovin'"
## [1] "Your Name Is Beautiful"
## [1] "Mockin' Bird Hill"
## [1] "It's A Little More Like Heaven"
## [1] "A Fallen Star"
## [1] "The Red Strokes"
## [1] "Would You Lay With Me (In A Field Of Stone)"
## [1] "Wings Of A Dove"
## [1] "Kentucky Waltz"
```

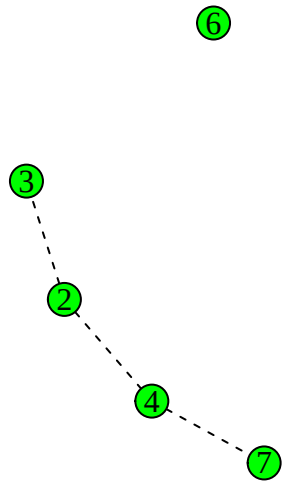
```
## [1] ""
## [1] "Topic  7"
## [1] "(Margie's At) The Lincoln Park Inn"
## [1] "Deck Of Cards"
## [1] "No Charge"
## [1] "Life Of A Poor Boy"
## [1] "History Repeats Itself"
## [1] "What's Your Mama's Name"
## [1] "Poor, Poor Pitiful Me"
## [1] "Po' Folks"
## [1] "Shiftwork"
## [1] "Sawmill"
## [1] ""
```

```
# Topic 1: Heartbreak Songs (Sad)
# Topic 2?: Country Rock/Pop
# Topic 3: Traditionalist Country (Pardi, Hank WIlliams)
# Topic 4: Bro-Country
# Topic 5: Sex Jams
# Topic 6: Love songs
# Topic 7: Family, Growing Up
```

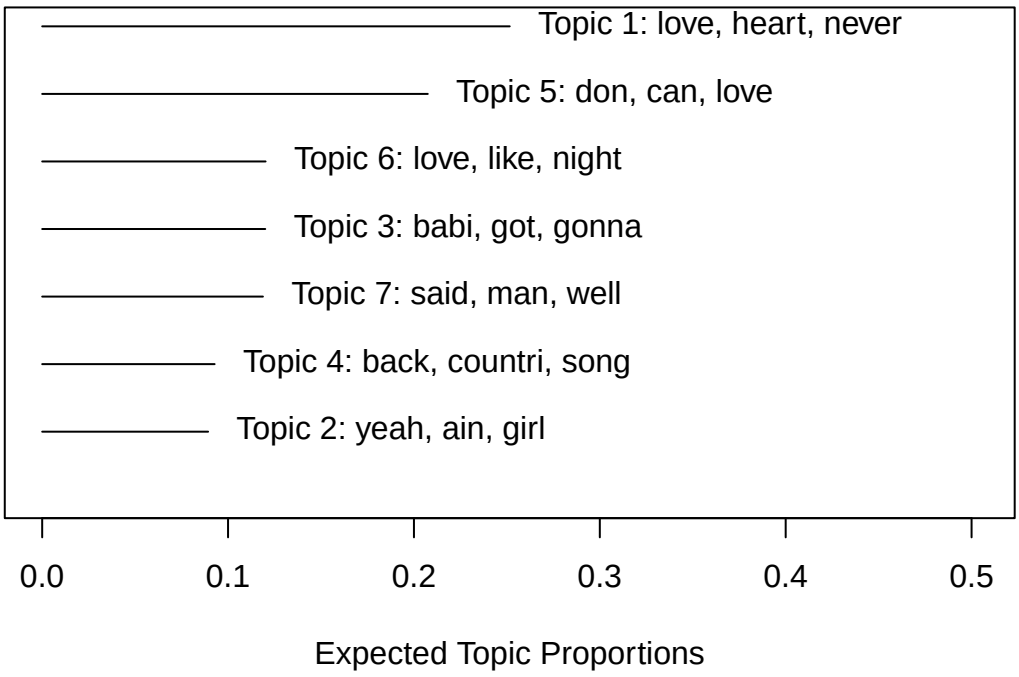
```
eff <- estimateEffect(formula = c(1:num_topics) ~ year,
                      # the line above matches the model specification we used
                      stmobj = out_covariates_7,
                      meta = prepped_data$meta,
                      uncertainty = "Global")

# Second, plot the results
plot(eff,
     covariate = "year",
     topics = c(1:num_topics),
     model = out_covariates_7,
     method = "continuous",
     xlab = "Year",
     main = "Effect of Year on Topic Proportion")
```




```
plot(topicCorr(out_covariates_7),
  vlabels = c(1:7), vertex.label.cex = 1.0, vertex.size = NULL)
```




```
plot.STM(out_covariates_7, type = "summary")
```


Top Topics
Topic 1: love, heart, never
Topic 5: don, can, love
Topic 6: love, like, night
Topic 3: babi, got, gonna
Topic 7: said, man, well
Topic 4: back, countri, song
Topic 2: yeah, ain, girl
0.0
0.1
0.2
0.3
0.4
0.5
Expected Topic Proportions