HW4

James Zhao

May 21, 2019

Q1 For this problem, use the dataset of death row statements from Texas inmates.

```
packages <- c("dplyr", "ggplot2", "lubridate", "stringr", "foreign", "xml2", "rvest", "tm", "tid
ytext", "proxy", "viridis", "fields", "mixtools", "tidyr", "topicmodels", "stm")
load.packages <- function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
}
lapply(packages, load.packages)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## Loading required package: stringr
```

```
## Loading required package: foreign
```

```
## Loading required package: xml2
```

```
## Loading required package: rvest
```

```
## Loading required package: tm
```

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

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

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
## Loading required package: tidytext
```

```
## Loading required package: proxy
```

```
##
## Attaching package: 'proxy'
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
## Loading required package: viridis
```

```
## Loading required package: viridisLite
```

```
## Loading required package: fields
```

```
## Loading required package: spam
```

```
## Loading required package: dotCall64
```

```
## Loading required package: grid
```

```
## Spam version 2.2-2 (2019-03-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

```
## See https://github.com/NCAR/Fields for
##  an extensive vignette, other supplements and source code
```

```
## Loading required package: mixtools
```

```
## mixtools package, version 1.1.0, Released 2017-03-10
## This package is based upon work supported by the National Science Foundation under Grant No.
SES-0518772.
```

```
##
## Attaching package: 'mixtools'
```

```
## The following object is masked from 'package:grid':
##
##     depth
```

```
## Loading required package: tidyr
```

```
## Loading required package: topicmodels
```

```
## Loading required package: stm
```

```
## stm v1.3.3 (2018-1-26) successfully loaded. See ?stm for help.
##  Papers, resources, and other materials at structuraltopicmodel.com
```

Q1.1 Import the data, pre-process it, and set up a DTM. For this analysis, do not remove sparse terms, but do remove people who gave no statement.

```
setwd("~/GitHub/MMSS_311_2")

tx <- read.csv("tx_deathrow_full.csv")
statement <- Corpus(VectorSource(tx$Last.Statement)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeWords, stopwords('english')) %>%
  tm_map(stemDocument) %>%
  tm_map(stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., stemDocument): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., stripWhitespace): transformation drops
## documents
```

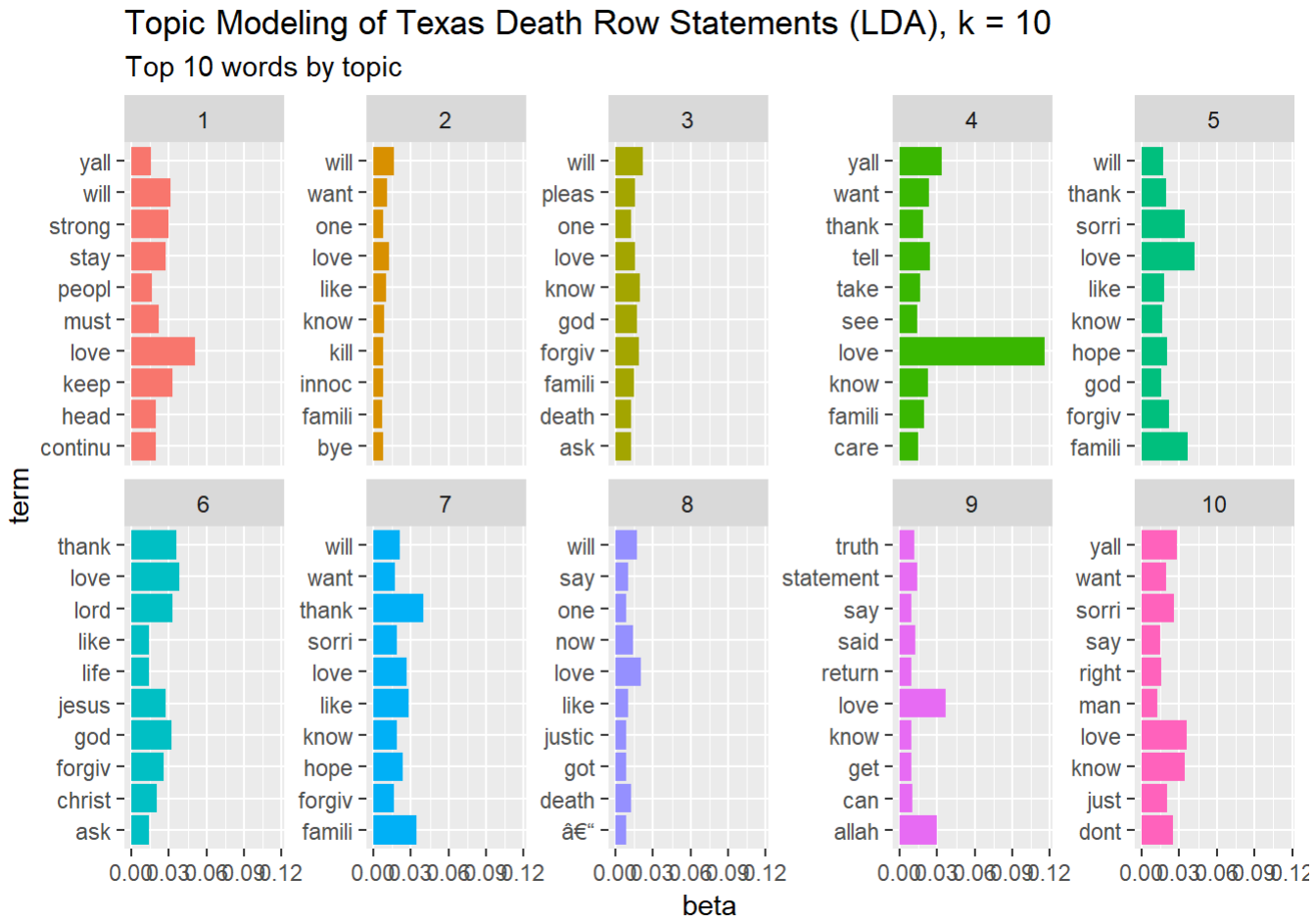
```
dtm.statement <- DocumentTermMatrix(statement) %>% as.matrix()

rowTotals <- apply(dtm.statement , 1, sum) #Find the sum of words in each Document
dtm.statement.new   <- dtm.statement[rowTotals> 0, ]

empty_rows <- which(rowSums(as.matrix(dtm.statement)) == 0)
```

Q1.2 Use LDA to assess topics in the statements, with k = 10 topics. Note: this may take a while to run.

```
mod.out.10 <- LDA(dtm.statement.new, k=10, control = list(seed=1))
```

Q1.3 Use tidy to get the results out of the model. Show the top 10 most likely words in each topic.

```
tidy(mod.out.10) %>%
  group_by(topic) %>%
  top_n(10,beta) %>%
  ungroup() %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y", nrow = 2) +
  coord_flip() +
  xlab("term") +
  labs(title = "Topic Modeling of Texas Death Row Statements (LDA), k = 10", subtitle = "Top 10
 words by topic")
```




```
tidy(mod.out.10)
```

```
## # A tibble: 26,980 x 3
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 execut 4.11e- 3
##  2     2 execut 2.02e- 3
##  3     3 execut 4.41e- 3
##  4     4 execut 5.32e-39
##  5     5 execut 7.02e- 4
##  6     6 execut 1.64e-14
##  7     7 execut 5.74e- 4
##  8     8 execut 4.34e- 3
##  9     9 execut 1.77e- 3
## 10    10 execut 1.05e- 3
## # ... with 26,970 more rows
```

Q2

Q2.1 Set up the DTM from Question 1 to be correctly formatted for use with the stm package. Use the code from lecture or the documentation for the package to assist with this.

```
out <- stm::readCorpus(dtm.statement.new)
```

Q2.2 Use stm to fit a structural topic model with 10 topics, conditioning on race. Note: This may take a while to run and produce a lot of output. You can use the chunk option {r, results = “hide”} to omit the output from your knitted pdf file.

```
statement.out <- stm(documents = out$documents, vocab = out$vocab, 10, prevalence = tx$race, dat
a = tx[-empty_rows, ])
```

Q2.3 Inspect the topics with summary.

```
summary(statement.out)
```

```
## A topic model with 10 topics, 442 documents and a 2698 word dictionary.
```

```
## Topic 1 Top Words:
##       Highest Prob: thank, like, mom, brother, sister, good, support
##       FREX: thank, mom, chaplain, wife, veronica, sister, book
##       Lift: veronica, cori, grew, amber, maria, doug, mage
##       Score: thank, veronica, dad, book, gomez, doug, buri
## Topic 2 Top Words:
##       Highest Prob: god, lord, jesus, forgiv, love, ask, christ
##       FREX: christ, jesus, heaven, lord, holi, ask, forgiv
##       Lift: roman, salvat, diann, light, damien, inflict, paradis
##       Score: holi, jesus, christ, lord, sin, forgiv, pray
## Topic 3 Top Words:
##       Highest Prob: yall, know, love, dont, want, just, kill
##       FREX: yall, alright, didnt, hate, dont, kid, man
##       Lift: transcript, alba, roger, sidethat, burn, chester, elroy
##       Score: yall, alright, didnt, mean, worri, dont, kill
## Topic 4 Top Words:
##       Highest Prob: know, dont, will, get, want, say, now
##       FREX: â“, bye, anyth, allah, wit, execut, mad
##       Lift: fast, happend, rocki, lilia, momrobert, youin, miriam
##       Score: bye, allah, fake, pinkerton, â“, robinson, ros
## Topic 5 Top Words:
##       Highest Prob: got, will, want, that, peopl, lie, day
##       FREX: thou, nobodi, adam, hous, lie, green, leadeth
##       Lift: counti, district, herebi, anointest, green, leadeth, maketh
##       Score: thou, green, leadeth, cup, oil, pastur, preparest
## Topic 6 Top Words:
##       Highest Prob: will, innoc, peopl, must, continu, thi, love
##       FREX: lynch, march, trespass, must, daili, thi, black
##       Lift: liber, draw, staci, trix, protest, princess, associ
##       Score: black, lynch, march, must, thi, forward, liber
## Topic 7 Top Words:
##       Highest Prob: sorri, hope, forgiv, famili, peac, will, pain
##       FREX: pain, sorri, caus, hope, find, bring, peac
##       Lift: kehler, sanchez, donovan, intend, resolv, horribl, bother
##       Score: sorri, pain, peac, forgiv, truli, hope, apolog
## Topic 8 Top Words:
##       Highest Prob: love, strong, take, stay, home, bless, care
##       FREX: strong, stay, home, spanish, care, bless, take
##       Lift: throughout, grandmoth, dwight, ranger, grate, joke, melyssa
##       Score: stay, final, strong, home, bless, spanish, crazi
## Topic 9 Top Words:
##       Highest Prob: love, famili, tell, want, yes, warden, friend
##       FREX: tell, appreci, warden, famili, yes, much, friend
##       Lift: mentor, patel, soldier, ten, toe, jean, nighti
##       Score: love, appreci, famili, friend, hurt, everybodi, tell
## Topic 10 Top Words:
##       Highest Prob: love, will, now, one, allah, know, shall
##       FREX: boswel, chanc, shall, act, unto, becom, gift
##       Lift: marcus, rhode, dishonor, dust, sadden, texan, boswel
##       Score: boswel, act, eardmann, woe, asdadu, gift, chief
```

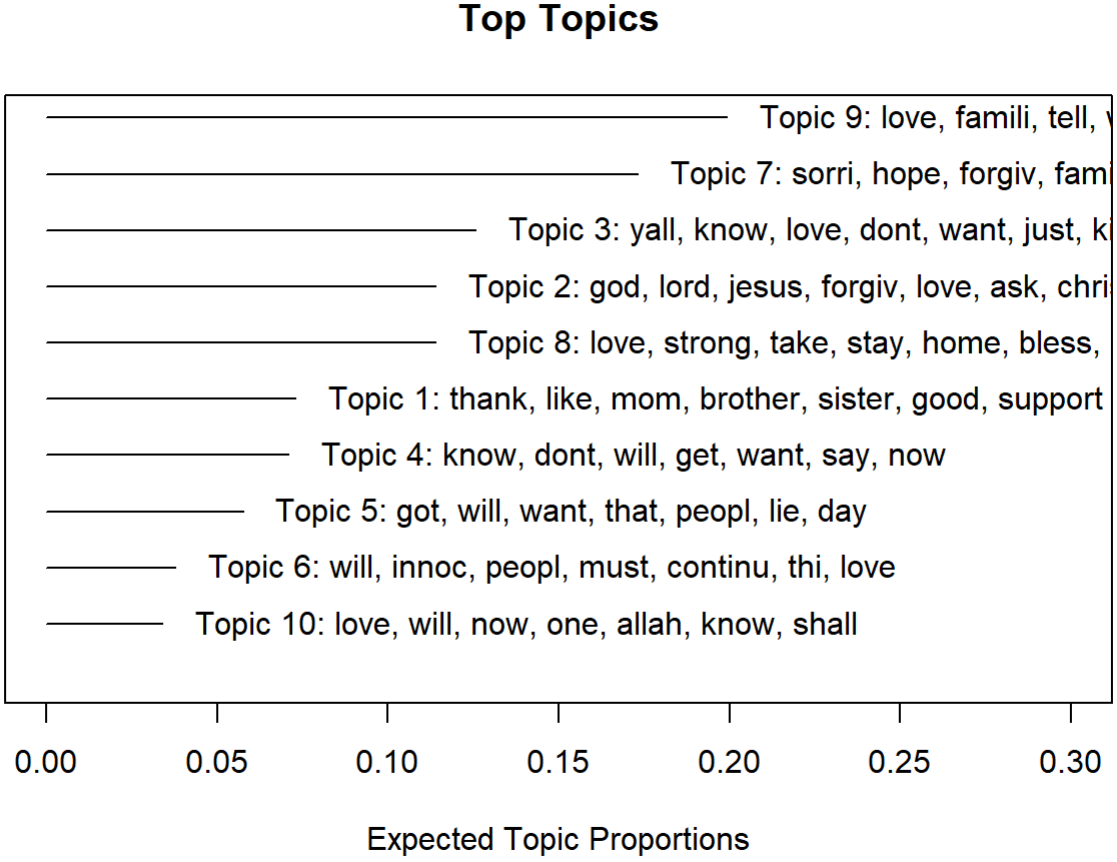
```
tidy(statement.out)
```

```
## # A tibble: 26,980 x 3
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 execut 6.64e- 96
##  2     2 execut 4.40e-  4
##  3     3 execut 1.16e- 45
##  4     4 execut 1.14e-  2
##  5     5 execut 6.21e- 68
##  6     6 execut 6.72e-  3
##  7     7 execut 6.03e-  4
##  8     8 execut 4.51e- 27
##  9     9 execut 1.38e- 42
## 10    10 execut 1.88e-150
## # ... with 26,970 more rows
```

```
labelTopics(statement.out)
```

```
## Topic 1 Top Words:
##       Highest Prob: thank, like, mom, brother, sister, good, support
##       FREX: thank, mom, chaplain, wife, veronica, sister, book
##       Lift: veronica, cori, grew, amber, maria, doug, mage
##       Score: thank, veronica, dad, book, gomez, doug, buri
## Topic 2 Top Words:
##       Highest Prob: god, lord, jesus, forgiv, love, ask, christ
##       FREX: christ, jesus, heaven, lord, holi, ask, forgiv
##       Lift: roman, salvat, diann, light, damien, inflict, paradis
##       Score: holi, jesus, christ, lord, sin, forgiv, pray
## Topic 3 Top Words:
##       Highest Prob: yall, know, love, dont, want, just, kill
##       FREX: yall, alright, didnt, hate, dont, kid, man
##       Lift: transcript, alba, roger, sidethat, burn, chester, elroy
##       Score: yall, alright, didnt, mean, worri, dont, kill
## Topic 4 Top Words:
##       Highest Prob: know, dont, will, get, want, say, now
##       FREX: â“, bye, anyth, allah, wit, execut, mad
##       Lift: fast, happend, rocki, lilia, momrobert, youin, miriam
##       Score: bye, allah, fake, pinkerton, â“, robinson, ros
## Topic 5 Top Words:
##       Highest Prob: got, will, want, that, peopl, lie, day
##       FREX: thou, nobodi, adam, hous, lie, green, leadeth
##       Lift: counti, district, herebi, anointest, green, leadeth, maketh
##       Score: thou, green, leadeth, cup, oil, pastur, preparest
## Topic 6 Top Words:
##       Highest Prob: will, innoc, peopl, must, continu, thi, love
##       FREX: lynch, march, trespass, must, daili, thi, black
##       Lift: liber, draw, staci, trix, protest, princess, associ
##       Score: black, lynch, march, must, thi, forward, liber
## Topic 7 Top Words:
##       Highest Prob: sorri, hope, forgiv, famili, peac, will, pain
##       FREX: pain, sorri, caus, hope, find, bring, peac
##       Lift: kehler, sanchez, donovan, intend, resolv, horribl, bother
##       Score: sorri, pain, peac, forgiv, truli, hope, apolog
## Topic 8 Top Words:
##       Highest Prob: love, strong, take, stay, home, bless, care
##       FREX: strong, stay, home, spanish, care, bless, take
##       Lift: throughout, grandmoth, dwight, ranger, grate, joke, melyssa
##       Score: stay, final, strong, home, bless, spanish, crazi
## Topic 9 Top Words:
##       Highest Prob: love, famili, tell, want, yes, warden, friend
##       FREX: tell, appreci, warden, famili, yes, much, friend
##       Lift: mentor, patel, soldier, ten, toe, jean, nighti
##       Score: love, appreci, famili, friend, hurt, everybodi, tell
## Topic 10 Top Words:
##       Highest Prob: love, will, now, one, allah, know, shall
##       FREX: boswel, chanc, shall, act, unto, becom, gift
##       Lift: marcus, rhode, dishonor, dust, sadden, texan, boswel
##       Score: boswel, act, eardmann, woe, asdadu, gift, chief
```

```
plot.STM(statement.out, type="summary", xlim=c(0,0.3), n=7)
```



Q2.4 Compare your results. How do the topics you find when conditioning on venue differ from those you found using standard LDA?

When we condition on race, we find that the new topics are somewhat race-specific and is related to background of different races. For example, one category focuses on British and ukip, which are related to the UK. By conditioning on race, we are separating groups based on race and each race group may have different topics, which is what we see in the above topic models.