Lab 5

Thomas Haase

October 6, 2025

Table of contents

Part 1

In this lab, we will use a data set containing a random sample of public Facebook posts by members of the U.S. Congress from 2017.1 Our broad objective in this first part of the lab is to explore what topics were discussed, and possible variation by party membership.

```
library(quanteda)
library(topicmodels)
library(slam)

library(word2vec)
```

```
library(data.table)
library(kableExtra)

library(easystats)

library(tibble)
library(ggplot2)
library(tidytext)

setwd("~/Github/ML-Labs/5")
```

Task 1

Begin by importing `fb-congress-data3.csv`. Report basic information about the data set; how many rows and column it has, as well as the name of the variables.

```
d <- read.csv("fb-congress-data3.csv")

tribble(
  ~Name,     ~Value,
  "Rows",    nrow(d) |> as.character(),
  "Columns", ncol(d) |> as.character()) |>
  kable()
```

Name	Value
Rows	6752
Columns	4

```
varnames <- names(d)
```

Variables contained in the dataset: doc_id, screen_name, party, message

Task 2

As you may have noticed from your inspection in #1, this data set has yet to be pre-processed (it contains punctuation, etc.). Hence, that is what you shall do now. More specifically, perform the following steps:

```
# i
corp <- d |>
  corpus(docid_field = "doc_id",
         text_field = "message",
         meta = list("screen_name",
                     "party"))
```

```r
# ii
toks <- corp |>
  tokens(remove_punct = T,
         remove_numbers = T,
         remove_symbols = T,
         remove_url = T) |>
# iii
  tokens_remove(stopwords("english"))

# iv
toks[1:3]
```

```
Tokens consisting of 3 documents and 2 docvars.
1 :
 [1] "President"     "Trump"         "backs"         "Paris"
 [5] "Agreement"     "economic"      "environmental" "national"
 [9] "security"      "moral"         "disaster"      "United"
[ ... and 6 more ]

2 :
 [1] "Many"       "thanks"     "first"      "class"      "summer"
 [6] "interns"    "Washington" "hard"       "work"       "folks"
[11] "#GA03"

3 :
 [1] "economy"    "needs"      "shot"       "arm"        "spur"
 [6] "growth"     "Co-Chair"   "bipartisan" "Problem"    "Solvers"
[11] "Caucus"     "I've"
[ ... and 27 more ]
```

```r
# v
dfm <- toks |> dfm() |>

# vi
  dfm_trim(min_termfreq = 5)
# only keep documents with more then 10 features
dfm <- dfm[which(rowSums(dfm)>=10), ]
```

Task 3

Now we are ready to do some topic modeling! To do so, we will use the topicmodels package, and the function LDA(). Set x to your document-term-matrix and specify method="Gibbs" (note: Gibbs is the name of a particular estimation procedure; see the Appendix of the lecture for more details). Set the number of iterations to 1000, and specify a seed number to ensure replicability (hint: to specify iterations and seed number, use the control argument). Finally, set the number of topics, K=50 With these settings specified, start the estimation. This could take a minute or two.

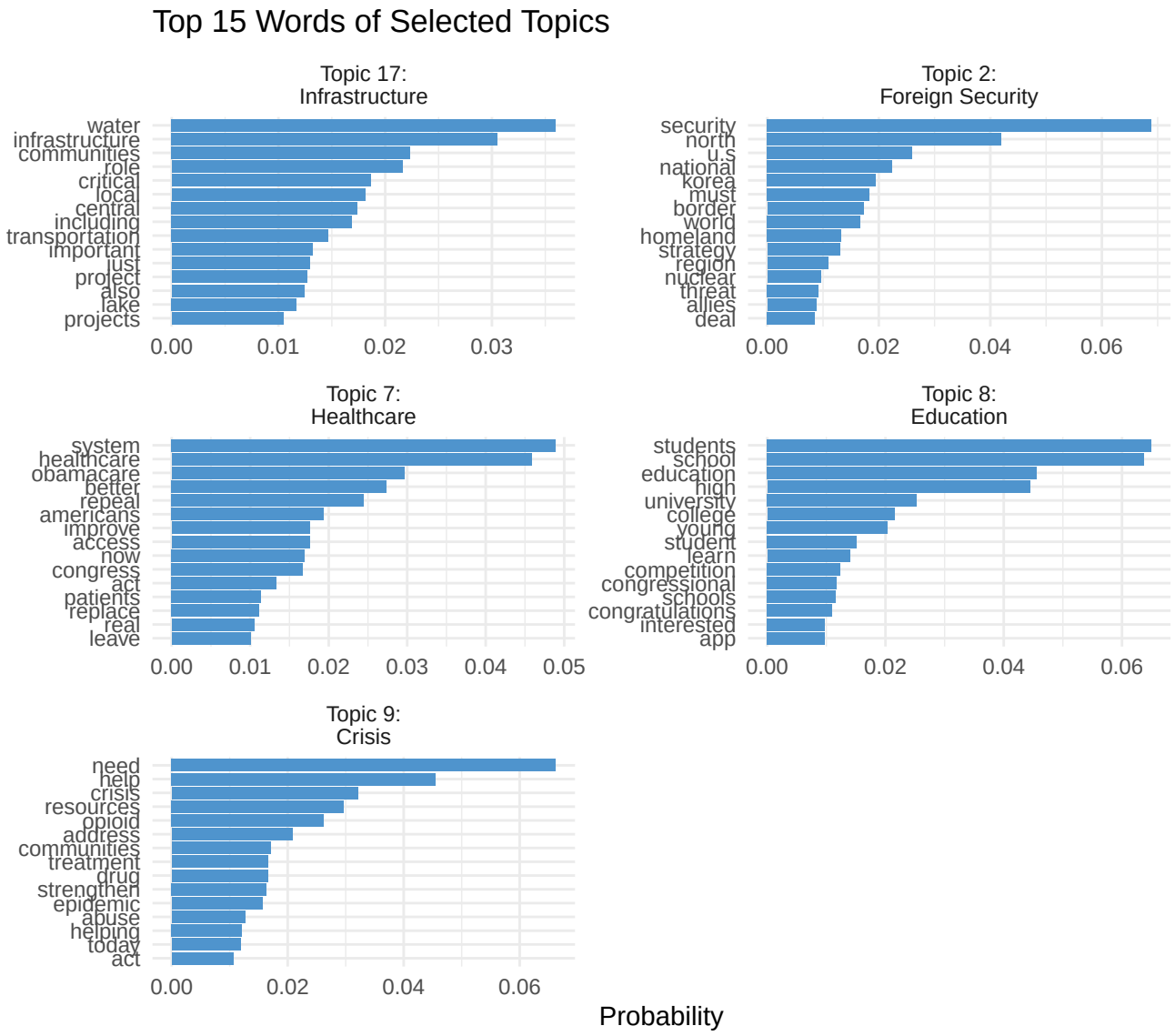
```r
set.seed(5)

K <- 50

if(!file.exists("lda.rds")){
  lda <- LDA(x = dfm, k = K,
             method="Gibbs",
             control=list(iter = 1000,
                          seed = 5,
                          verbose = 1))
  saveRDS(lda, file = "lda.rds")
} else {
  lda <- readRDS("lda.rds")
}
```

Task 4

Once the estimation is finished, use the `get_terms()` function to extract the 15 words with the highest probability in each topic. In a real research setting, we would carefully examine each of the topics. Here, I only ask you to briefly skim them, and then focus on 5 that (i) you think are interesting, (ii) has a clear theme, and (iii) are clearly distinct from the other topics. Provide a label to each of those based on the top 15 words. Complementing your label, please also provide a bar chart displaying on the y-axis the top 15 words, and on the x-axis their topic probabilities. Hint: you can retrieve each topic's distribution over words using topicmodels's function posterior.3 Lastly, please also report a general assessment—based on your skim—about the general quality of the topics; do most of them appear clearly themed and distinct, or are there a lot of "junk" topics?

```r
# get_terms(lda,15)

words <- data.table(topic = 1:K,
                    posterior(object = lda)$terms) |>
  melt.data.table(id.vars = 'topic')
words <- words[order(value,decreasing = T)]

t15 <- words[,head(.SD,15),by='topic']
t15 <- t15[topic %in% c(2,7,8,9,17)]

# plot
topic_labels <- c(
  "2"  = "Topic 2:\nForeign Security",
  "7"  = "Topic 7:\nHealthcare",
  "8"  = "Topic 8:\nEducation",
  "9"  = "Topic 9:\nCrisis",
  "17" = "Topic 17:\nInfrastructure"
)
```

```
# Apply topic labels
t15[, topic_label := topic_labels[as.character(t15$topic)]][!is.na(topic_label)] |>
  ggplot(aes(x = value, y = reorder_within(variable, value, topic_label))) +
  geom_bar(stat = 'identity', position = 'dodge', fill = "steelblue3") +
  facet_wrap(~topic_label, scales = 'free', ncol = 2) +
  scale_y_reordered() +
  labs(title = "Top 15 Words of Selected Topics",
       x = "Probability", y = "") +
  theme_minimal()
```



There are a lot of junk topics, but a few of them are very distinct.

Task 5

Out of the 5 topics that you labeled, select two which you think are particularly interesting. For these two, identify the three documents which have the highest proportion assigned of this topic (hint 1: use topicmodels's `posterior()` to extract documents' distribution over topics | hint 2: to identify the document ids which correspond to each row of what you extract from posterior(), you can use `ldaobject@documents`. See help file for more details.), and do a qualitative inspection ($= 2 \times 3$ documents to read). Does your readings corroborate your labels? Are they about what you expected?

```
documents <- data.table(doc_id = lda@documents,
                        posterior(object = lda)$topics)
# Assign topics as column names
colnames(documents)[2:ncol(documents)] <- paste0('Topic',colnames(documents)[2:ncol(documents)])



top_docs <- documents[
  order(Topic8,decreasing = T)][
    1:3,doc_id] |>
  as.integer()

tibble(ID = as.character(top_docs),
       Education = as.character(corp)[top_docs]) |>
  kable() |>
  kable_styling(bootstrap_options = c("striped", "hover",
                                      "condensed", "responsive")) |>
  column_spec(1, bold = TRUE, width = "3em") |>
  column_spec(2, width = "40em")
```

ID	Education
3937	Thompson Valley High School is a local example of Career and Technical Education in action. The school combines traditional classroom education with applied learning. Yesterday, I got to attend a cooking, entrepreneurship, and agriculture class alongside students. As the top-Democratic member of the Early Childhood and K-12 subcommittee on the Education and Workforce Committee, I've worked hard to support Career and Technical Education programs. This spring, the Education and Workforce committee debated and the House of Representatives approved a reauthorization of the Perkins Career and Technical Education program that will support innovation and bring our CTE programs into the 21st century. I want to thank the educators who are on the front lines teaching Career and Technical Education, helping to prepare students for the real-world and good jobs. http://www.reporterherald.com/news/education/ci_31312475/polis-visits-career-classes-thompson

3933	I visited Lake Minneola High School, Weeki Wachee High School, Crystal River High School, Lecanto High School, Wildwood Middle High School, Tavares High School and Lake Weir High School to congratulate the winning students. This is the biggest turnout we've had so far - 86 students participated in this year's competition. The spectacular artwork and record turnout is a reflection of the emphasis that our schools are placing on the arts.
274	Announcing the 2017 Congressional Art Competition! Your students are invited to participate in my annual Congressional Art Competition–now open to all high school students who reside in the 11th District. Entries can include paintings, drawings, prints, and more. We have had a blast with this competition in the past and are really looking forward to the entries this year! Winners will be chosen by a panel of art professionals, and there will be a reception for all the students who enter–as well as their teachers and families–on April 29, 2017. The over-all winner of our district's competition will receive two round-trip tickets to the National Reception in Washington, DC, a $3,000 scholarship to a prestigious Southeastern art college, and their art will be displayed for one year in the U.S. Capitol! You can find more information on my website below. Note that the DEADLINE to enter this year is Friday April 21, 2017.

```
top_docs <- documents[
  order(Topic17,decreasing = T)][
    1:3,doc_id] |>
  as.integer()

tibble(ID = as.character(top_docs),
       Infrastructure = as.character(corp)[top_docs]) |>
  kable() |>
  kable_styling(bootstrap_options = c("striped", "hover",
                                      "condensed", "responsive")) |>
  column_spec(1, bold = TRUE, width = "3em") |>
  column_spec(2, width = "40em")
```

ID	Infrastructure
6440	From harmful algal blooms in Owasco Lake, to bursting pipes in Syracuse, to sewer problems in Oswego, to aging filtration systems in communities around the district – our region faces water infrastructure challenges like never before. Today I stood in at the northern shore of Owasco Lake with a bipartisan group of local leaders to highlight my commitment to strengthening water infrastructure for CNY. I was proud to lead bipartisan letters in Congress to double the Clean & Drinking Water Funds, invest in Harmful Algal Bloom research, and to prioritize programs for small and rural communities to develop wastewater and drinking water infrastructure.

2068	I was pleased to join Transportation and Infrastructure Chairman Shuster, Subcommittee Chairman Graves, and my Florida delegation colleagues to discuss the upcoming WRDA bill. This important piece of legislation has authorized critical Everglades restoration projects, including the Central Everglades Planning Project and the Picayune Strand. It has also authorized much needed upgrades and expansions to PortMiami and Port Everglades. Because of these projects, we are able to restore and protect the Everglades for future generations, as well as provide our ports with adequate resources to fit the needs of our growing economy. This roundtable is just the beginning of the process, but I appreciate that the Transportation and Infrastructure Committee recognizes how important this bill is to Florida. I thank Chairman Shuster for his leadership throughout the years on ensuring WRDA properly served our community. I remain committed to working with Speaker Ryan, Chairman Shuster, and Subcommittee Chairman Graves to ensure the 2018 WRDA bill reflects Florida's infrastructure needs.
2037	This is great news in our efforts to keep Lake Erie healthy and safe! Lake Erie is one of Ohio's most precious and important natural resources. Many lakeshore communities rely on it for clean and affordable drinking water and the lake plays a key role in Ohio's economy. For the last few years, I have been working with the Ohio EPA to prevent the U.S. Army Corps of Engineers from dumping dredged sediment from the Cuyahoga River that contains contaminants called polychlorinated biphenyls, or PCBs, into Lake Erie. Elevated PCB levels can lead to contamination among the lake's fish populations. I'm very happy to see this result!

For both topics the three selected prototypical-tweets adress gunviolence and infrasstructure. The model seems very convincing, since the content of the inspected texts meet my expectation.

Task 6

Now, estimate a topic model—as in #3—but with K=3 instead. Extract the top 15 words from each topic, (try to) label each, and then make an assessment of the overall quality of them. To further explore the quality of this topic model, reconsider the documents you read in #5: extract the distribution over topics for these documents (from your new K=3 model). How well does this topic model capture the theme of these documents? Based on your analysis, which of the two K's do you prefer? Motivate.

```
# estimate LDA
set.seed(5)
K <- 3

if(!file.exists("lda2.rds")){
  lda <- LDA(x = dfm, k = K,
             method="Gibbs",
             control=list(iter = 1000,
                          seed = 5,
                          verbose = 1))
  saveRDS(lda, file = "lda2.rds")
} else {
  lda <- readRDS("lda2.rds")
```

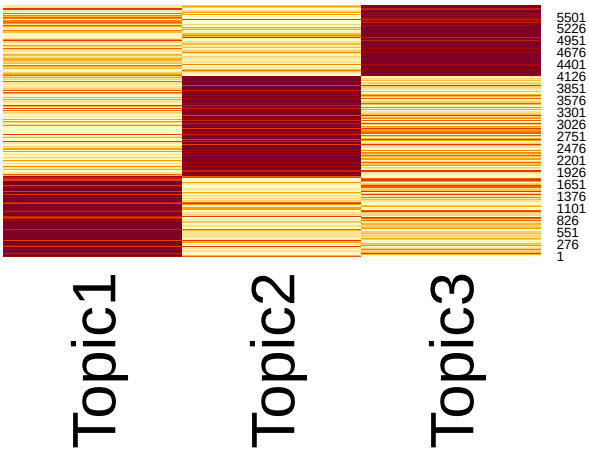
```
}


# get top words
get_terms(lda,15) |>
  kable(col.names = c("Healthcare & Tax",
                      "Veterans & Community",
                      "National Government"),
        caption = "Top 15 Terms of Topic Model (K = 3)")
```

Healthcare & Tax	Veterans & Community	National Government
health	today	president
care	veterans	trump
can	great	u.s
bill	day	federal
act	community	national
tax	week	must
americans	office	congress
families	thank	years
help	service	law
new	district	house
make	washington	american
people	congressional	continue
work	local	security
need	one	states
house	students	committee

Table 2: Top 15 Terms of Topic Model (K = 3)

It seems like multiple topics got recognized as one from the model. There are probably more then 3 larger topics adressed in the texts.

```
documents <- data.table(doc_id = lda@documents,
                        posterior(object = lda)$topics)
# Assign topics as column names
colnames(documents)[2:ncol(documents)] <- paste0('Topic',colnames(documents)[2:ncol(documents)])

set.seed(5)

documents[,-"doc_id"] |>
  as.matrix() -> mat

par(mar=c(5,10,4,2))
mat[order(apply(mat, 1, which.max)), ] |>
  heatmap(Rowv = NA, Colv = NA, margins = c(10, 1))
```



The heatmap, that ordered the documents by maximum probability, shows, that the distribution over Topics is quite distinct. Only few texts have high probability in all topics. The K= 50 model captured unnecessary topics, while the K = 3 one still shows many stripes in its heatmap. These could probably make up new topics! The optimum lies somewhere inbetween 3 and 50. Just from the Interpretation the 50 topic model is more powerful because it captures more distinctions.

Task 7

Continuing with the topic model you concluded the most appropriate, perform the following sets of analyses:

- i. Compute the prevalence of each topic, across all documents. Report which is the most prevalent topic, overall, and then report—in the form of a single plot; e.g., a bar chart—the prevalence of the topics you labeled.

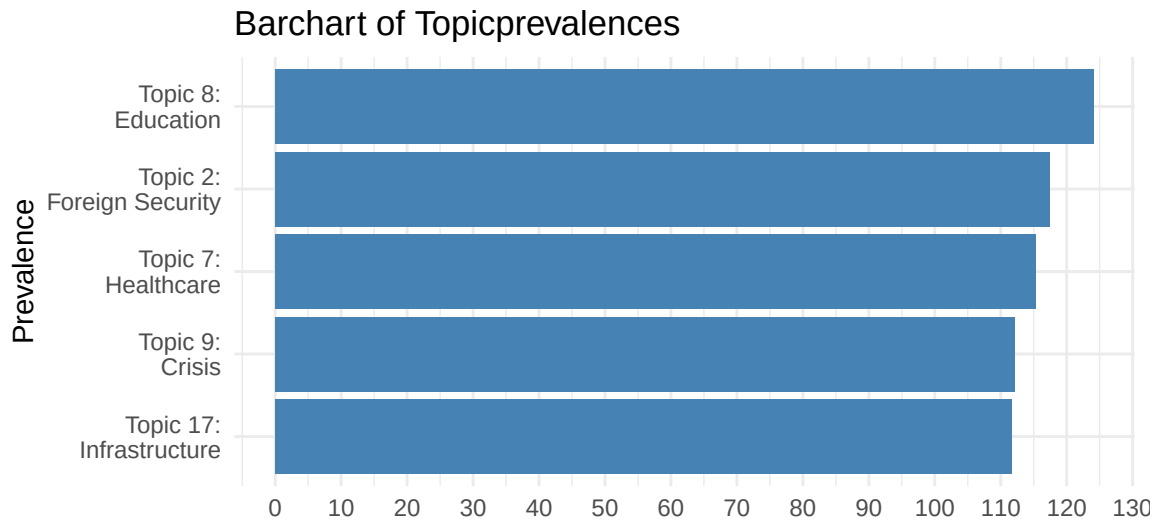
-ii. Compare the prevalence on your labeled topics between democrats and republicans. You can for example fit a fractional regression model using glm(family="quasibinomial") or using t-tests of difference in means. Interpret.

```
# load 50 topic lda
lda <- readRDS("lda.rds")
K <- 50


documents <- data.table(doc_id = lda@documents,
                        posterior(object = lda)$topics)
colnames(documents)[2:ncol(documents)] <- paste0('Topic',colnames(documents)[2:ncol(documents)])

# query for most prevalent topic
top_topic <- documents[, -"doc_id"] |>
  colSums() |>
  which.max() |>
  names()
```

```r
# report
cat("Most prevalent topic:", top_topic, "\n")
```

```
Most prevalent topic: Topic11
```

```r
# plot labled topics
data.table(Topic = documents[, -"doc_id"] |>
             colSums() |>
             names(),
           Prevalence = documents[, -"doc_id"] |>
             colSums())[
               sub("Topic", "", Topic) %in% names(topic_labels)
               ][
                 , Topic := topic_labels[sub("Topic", "", Topic)]
                 ] |>
ggplot(aes(x = reorder(Topic, Prevalence),
           y = Prevalence)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    x = "Prevalence", y = "",
    title = "Barchart of Topicprevalences"
  ) +
  scale_y_continuous(breaks = seq(0, 150, 10)) +
  theme_minimal()
```

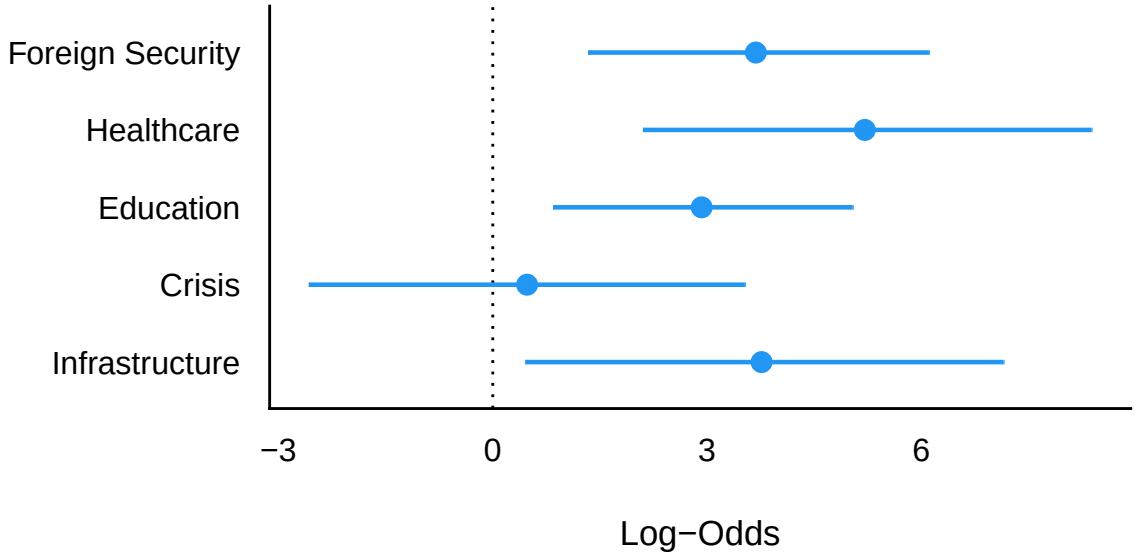



```r
# prepare data
documents <- data.table(
  doc_id = lda@documents,
  posterior(lda)$topics
)
```

```r
colnames(documents)[2:ncol(documents)] <- paste0('Topic',colnames(documents)[2:ncol(documents)])
# add partyinformation
documents <- documents[, party := docvars(dfm)$party |>
                         factor(levels = c("Democrat",
                                           "Republican"))]
# filter data
documents <- documents[, .(doc_id,party,
                           Topic2,Topic7,
                           Topic8,Topic9,
                           Topic17)]
setnames(documents,
         c("Topic2","Topic7","Topic8",
           "Topic9","Topic17"),
         c("Foreign Security",
           "Healthcare",
           "Education",
           "Crisis",
           "Infrastructure"))

# calculate model
glm <- glm(party ~ `Foreign Security` + Healthcare +
             Education + Crisis + Infrastructure,
           family = "binomial",
           data = documents)

# because quasibinomial was suggested but
# Im using binomial lets check overdispersion
check_overdispersion(glm)
```

```
# Overdispersion test

 dispersion ratio = 1.000
          p-value =  0.88
```

```r
glm |>
  parameters(exponentiate = F) |>
  plot()
```




```
glm |> summary()
```

```
Call:
glm(formula = party ~ `Foreign Security` + Healthcare + Education +
    Crisis + Infrastructure, family = "binomial", data = documents)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -0.3688     0.0742  -4.970 6.68e-07 ***
`Foreign Security`   3.6819     1.2136   3.034  0.00242 **
Healthcare           5.2070     1.5988   3.257  0.00113 **
Education            2.9235     1.0677   2.738  0.00618 **
Crisis               0.4806     1.5508   0.310  0.75662
Infrastructure       3.7617     1.7024   2.210  0.02713 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7965.7  on 5747  degrees of freedom
Residual deviance: 7937.4  on 5742  degrees of freedom
AIC: 7949.4

Number of Fisher Scoring iterations: 4
```

The topics Foreign Security, Healthcare, Education and Infrastructure can distinguish between both parties statistically significant. Adressing the topic "Crisis" can not be used to distinguish between parties.

Task 8

BONUS (not obligatory; suggestion; do after you have completed the rest of the lab). As a bonus exercise—to expose you to the traditional computer sciency way of selecting the number of topics, K-you shall consider a data-driven approach, relying on the measure of hold-out likelihood (or, perplexity as its also called). To do so, do the following:

- i. Split your document term matrix into two (a training and test set); 80/20 division.
- ii. Write a loop which in each iteration (a) estimates a topic model using a particular K, and then (b) computes (and stores) its perplexity using the topicmodels function perplexity(), which takes as input the model object and the test document-term-matrix (note: the document-term-matrix needs to be transformed into a particular format: use … for this).
- iii. Consider the following range of K: $3, 10, 25, 50, 75, 100, 200$, and run the loop. This may take a few minutes. Once the loop has finished, plot your results (x-axis: K, y-axis: perplexity).

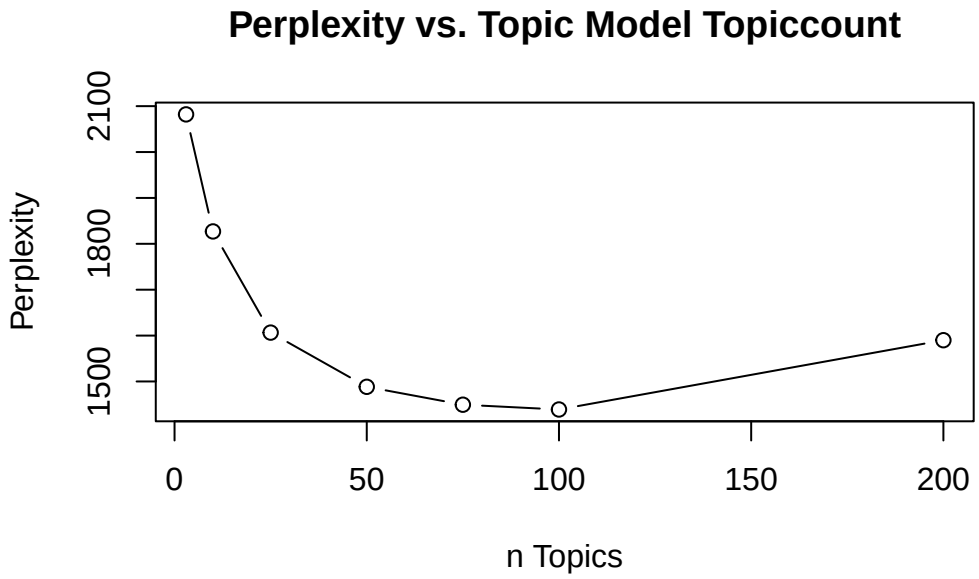
Interpret. Based on this, what is a reasonable K?

```
rm(documents, lda, mat, t15, words, toks, corp,
   K, top_docs, top_topic, topic_labels, varnames)

# Split data
set.seed(5)
train_ids <- sample(seq_len(nrow(dfm)), size = floor(0.8 * nrow(dfm)))
train_data <- dfm[train_ids, ]
test_data <- dfm[-train_ids, ]
```

```
ks <- c(3, 10, 25, 50, 75, 100, 200)
perp <- c()

if(!file.exists("perplexity.rds")){

  for(i in 1:length(ks)){

    # Train
    current_tm <- LDA(x = train_data,
                      k = ks[i],
                      method="Gibbs",
                      control=list(iter = 500,
                                   seed = 1,
                                   verbose = 100))
    # Compute perplexity
    perp[i] <- perplexity(object = current_tm,
                          newdata = as.simple_triplet_matrix(test_data)) # For some reason, perpl
    print(perp)
  }

  saveRDS(perp, file = "perplexity.rds")
} else {
```

```
  perp <- readRDS("perplexity.rds")
}
```

```
par(mar=c(5,4,4,2)+0.1)
plot(x=ks,y=perp,
     type="b",
     xlab="n Topics",
     ylab="Perplexity",
     main="Perplexity vs. Topic Model Topiccount")
```



Perplexity measures how well LDA models predict unseen documents, with lower scores indicating better performance and topic understanding. It measures how well the model predicts unseen or held-out documents. Since perplexity is lowest at 100 topics the facebookposts can statistically optimal be clustered into 100 topics. In a previous task it was already detected, that a K of 50 topics already contain a lot of uninterpretable ones, which is why 100 topics might be statistically optimal but useless to human interpreters.

Part 2

In this second part of the lab, we will continue with the data U.S. Congress–Facebook posts data set. However, now with a different focus: a focus on the word-level, using word embeddings instead of topic models.

```
rm(list = ls())
```

Task 1

Because word embeddings are not negatively affected by stop words or other highly frequent terms, your first task is to reimport the `fb-congress-data3.csv` file, and re-process the data; performing step i–ii in task #2, but skipping #3. Here, we also do not want to transform our documents into a document-term matrix. Instead, after having tokenized and cleaned the documents, paste each back into a single string per document. Hint: for this, you could for example write: sapply(mytokens,function(x)paste(x,collapse = " ")). As a last pre-processing step, transform all your text into lowercase (hint: you can use the function tolower() for this).

```
d <- read.csv("fb-congress-data3.csv")

# i
corp <- d |>
  corpus(docid_field = "doc_id",
         text_field = "message",
         meta = list("screen_name",
                     "party"))
# ii
toks <- corp |>
  tokens(remove_punct = T,
         remove_numbers = T,
         remove_symbols = T,
         remove_url = T)

s <- sapply(toks,function(x)paste(x,collapse = " ")) |>
  tolower()
```

Task 2

Now we are set to fit word embeddings! To begin, let us fit one word embedding model to all documents—not separating posts by democrats and republicans. Use word2vec's word2vec() function to fit a cbow model (type="cbow") using 15 negative samples per real context/observation (negative=15), and setting dim=50, the number of dimensions of the word vetors/embeddings. This will take a minute or two.

```
set.seed(5)

we <- word2vec(x = s, type = "cbow",
               iter = 50, hs = T,
               negative = 15,
               threads = 4)
```

Task 3

When the estimation in #2 is finished, identify the 10 nearest terms to 3 focal words of your choice/interest. Make sure to select words which occur frequently in your data. Hint: to

retrieve the closest words in embedding/word vector space, you may use the following code: `predict(w2v,c("word2","word2","word3"),type="nearest",top_n = 10)`, where wv2 is the object storing the fitted model of the word2vec function. Does the results you find makes sense? Why/why not?

```
nearw <- predict(we, c("trump",
                       "food",
                       "russia"),
                 type = "nearest",
                 top_n = 10)



nearw$trump |> kable()
```

term1	term2	similarity	rank
trump	obama	0.9036403	1
trump	trump's	0.8998323	2
trump	obama's	0.8004857	3
trump	vice	0.7599276	4
trump	elect	0.7353007	5
trump	tone	0.7083130	6
trump	omaha	0.7071037	7
trump	healthier	0.6882860	8
trump	companies	0.6847061	9
trump	kennedy	0.6831337	10

The term "trump" has names of other leaders and leadingpositions in its nearby area. This makes intuitive sense, because trump was president in the year when the facebookposts were collected.

```
nearw$food |> kable()
```

term1	term2	similarity	rank
food	addiction	0.6854993	1
food	shoreline	0.6809211	2
food	maintenance	0.6733322	3
food	island	0.6649883	4
food	port	0.6615895	5
food	standards	0.6594132	6
food	tampa	0.6586419	7
food	bright	0.6465849	8
food	aids	0.6464297	9
food	procedures	0.6442885	10

The term "food" is surrounded by other terms that are crisis, infrastructure and ressource related. This makes intuitive sense, because food just as infrastructure is something that should be made always available by the politics in its society-steering role. Also all terms are crisis related since a crisis is often related with situations were food, ports, stores, … is not available. Some nonsensical numbers are also in there, aswell as "comment".

```
nearw$russia |> kable()
```

term1	term2	similarity	rank
russia	russian	0.8283573	1
russia	russia's	0.7625262	2
russia	collusion	0.7168145	3
russia	integrity	0.6978391	4
russia	clinton	0.6974280	5
russia	undue	0.6957203	6
russia	broader	0.6924533	7
russia	associates	0.6908404	8
russia	campaign	0.6823993	9
russia	relationship	0.6743895	10

The term "russia" is surrounded by synonyms/similar terms, aswell as "campaign","mueller", "elections", "extend" and "meddling". Since russian interference in the US votings was a big topic in 2017 these associations are no s urprise. The resulting FBI investigation put pressure on trump, which is why he fired the FBI director and Robert Mueller was appointed to oversee the investigation.

Task 4

What initially made people so excited about word embeddings was their surprising ability to solve seemingly complex analogy tasks. Your task now is to attempt to replicate one such classical analogy result, first with the embedding vectors that you have already estimated, and second using a pre-trained embedding model. To do so, please perform the following steps:

- i. Extract the whole embedding matrix: embedding <- as.matrix(w2v).
- ii. Identify the rows in the embedding matrix which correspond to king, man, woman, and create a new R object kingtowoman which is equal to the vector for king, minus the vector for man, plus the vector for woman. Hint: to extract the row corresponding to a particular word (e.g., "king"), you may use w2v[rownames(w2v)=="king",].
- iii. Use word2vec's function word2vec_similarity() to identify the 20 most similar words to kingtowoman. Do you find "queen" in the top 20? Why do you think you get the result you do?
- iv. Next, we will consider a pre-trained embedding model (trained on all Wikipedia articles that existed in 2014 and about 5 million news articles). The embedding vectors from this model are stored in the file `glove6B200d.rds` Note: this file is large; more than 300MB. Use readRDS() to import it, and stored it in an R object called pretrained. Each row stores the embedding vector for a particular word. With this info in mind, report how many embedding dimensions were used for this model, and how many words we have embedding vectors for.

- v. Repeat steps ii–iii for pretrained. Does "queen" appear in the top 20 here? What do you think explains this difference/similarity to the self-trained result?

```
# i.
e <- as.matrix(we)

# ii.
KING  <- e[rownames(e)=="king",]
MAN   <- e[rownames(e)=="man",]
WOMAN <- e[rownames(e)=="woman",]

kingtowoman <- KING - MAN + WOMAN

# iii
kingtowoman <- matrix(kingtowoman, nrow = 1)
rownames(kingtowoman) <- "kingtowoman"

word2vec_similarity(kingtowoman, e,
                    top_n = 20,
                    type = "cosine") |>
  kable()
```

term1	term2	similarity	rank
kingtowoman	king	0.7485505	1
kingtowoman	united	0.5414166	2
kingtowoman	rev	0.5140675	3
kingtowoman	nra	0.4926765	4
kingtowoman	lives	0.4851092	5
kingtowoman	michelle	0.4833211	6
kingtowoman	survey	0.4831932	7
kingtowoman	sanctuary	0.4829294	8
kingtowoman	launched	0.4825294	9
kingtowoman	counterparts	0.4816345	10
kingtowoman	lower	0.4698601	11
kingtowoman	york	0.4495962	12
kingtowoman	kansas	0.4345347	13
kingtowoman	completed	0.4325152	14
kingtowoman	modernize	0.4269441	15
kingtowoman	thin	0.4254866	16
kingtowoman	magazine	0.4254506	17
kingtowoman	demonstrates	0.4210394	18
kingtowoman	bipartisan	0.4170111	19
kingtowoman	groundbreaking	0.4147043	20

Looks like "queen" is not among the closest 20 terms!

```
e[rownames(e)=="queen",]
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
     [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
     [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
     [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
```

The reason is because queen is not part of the embeddingspace! Therefore we can only approximate where it should be but will never find it as an observation in the space.

```
# iv.
pretrained <- readRDS("glove6B200d.rds")

nrow(pretrained)
```

```
[1] 400000
```

```
ncol(pretrained)
```

```
[1] 200
```

```
#pretrained[50:53,1:3]
```

The embeddingmodel contains 400000 words each with a vector of size 200.

```
# v.

KING  <- pretrained[rownames(pretrained)=="king",]
MAN   <- pretrained[rownames(pretrained)=="man",]
WOMAN <- pretrained[rownames(pretrained)=="woman",]

kingtowoman <- KING - MAN + WOMAN

kingtowoman <- matrix(kingtowoman, nrow = 1)
rownames(kingtowoman) <- "kingtowoman"

word2vec_similarity(kingtowoman,
                    pretrained,
                    top_n = 20,
                    type = "cosine") |>
  kable()
```

term1	term2	similarity	rank
kingtowoman	king	0.8209068	1
kingtowoman	queen	0.7119166	2
kingtowoman	princess	0.6121214	3
kingtowoman	monarch	0.6024806	4
kingtowoman	prince	0.5960041	5
kingtowoman	throne	0.5915314	6
kingtowoman	daughter	0.5588056	7
kingtowoman	elizabeth	0.5547403	8
kingtowoman	kingdom	0.5494517	9
kingtowoman	mother	0.5419817	10
kingtowoman	crown	0.5347209	11
kingtowoman	wife	0.5149215	12
kingtowoman	royal	0.5115056	13
kingtowoman	marry	0.5111473	14
kingtowoman	woman	0.5082911	15
kingtowoman	marriage	0.5049194	16
kingtowoman	sister	0.4976907	17
kingtowoman	margaret	0.4923242	18
kingtowoman	husband	0.4798765	19
kingtowoman	her	0.4793502	20

Queen does indeed appear as the second term in the embeddingspace. Most of the terms appearing in the list have something to do with "king". The terms in the self-trained list deviated more since the corpus of the self trained embeddings does not contain so many words related to "king".

- vi. Given the result in (v), what do you expect, if you were to construct a measure of occupational gender bias along the lines of Garg et al. (2018), that is by comparing the distance between different occupations and gendered words, for example: occupationalbias = dist(statistician, man) – dist(statistician, woman), would this score be "more correct" than the one you would obtain from the same calculation on your facebook/congress model? Why/why not?

What is correct depends a lot on the question and context. In general the wikipediacorpus is a larger and more general dataset which could be interesting to research because of that. The genderbias vector in this wikipediadataset would then be a "very correct" measure of itself. On the other hand in our own facebook dataset a similar question would be possible to answer in the same way. Since the data is more sparse/smaller its harder to trust the results. During the calculation the words place themselves in relation to all other words. If there are less words the uncertainty of its position is generally higher.

Task 5

Now we shall make a comparison between democrats and republicans. Split the data from step #1 into two based on party affiliation. Then, repeat 2–3, but now separately for republicans and democrats. For #3, select words which you expect might be used differently between the two political camps (but still are frequently used by both; for example "abortion", "obamacare"). Do you find any differences? Do they align with your expectations?

```r
# Split data
toks_dem <- toks[which(toks$party == "Democrat"), ]
toks_rep <- toks[which(toks$party == "Republican"), ]

s_dem <- sapply(toks_dem, function(x)paste(x,collapse = " ")) |>
  tolower()
s_rep <- sapply(toks_rep, function(x)paste(x,collapse = " ")) |>
  tolower()


# word2vec
set.seed(5)

we_dem <- word2vec(x = s_dem,
                   type = "cbow",
                   iter = 50, hs = T,
                   negative = 15,
                   threads = 4)

we_rep <- word2vec(x = s_rep,
                   type = "cbow",
                   iter = 50, hs = T,
                   negative = 15,
                   threads = 4)

# get nearby words
nw_dem <- predict(we_dem,
                  c("trump",
                    "food",
                    "russia"),
                  type = "nearest",
                  top_n = 10)

nw_rep <- predict(we_rep,
                  c("trump",
                    "food",
                    "russia"),
                  type = "nearest",
                  top_n = 10)


nw_dem$trump |> kable()
```

term1	term2	similarity	rank
trump	obama	0.8823119	1
trump	trump's	0.8626608	2

term1	term2	similarity	rank
trump	vice	0.8085989	3
trump	drug	0.7147294	4
trump	kennedy	0.6839656	5
trump	elect	0.6763733	6
trump	spanish	0.6691946	7
trump	relations	0.6611997	8
trump	betsy	0.6381862	9
trump	roosevelt	0.6301155	10

```
nw_rep$trump |> kable()
```

term1	term2	similarity	rank
trump	trump's	0.8485833	1
trump	obama's	0.8372140	2
trump	vice	0.8041630	3
trump	obama	0.7993160	4
trump	george	0.7501575	5
trump	korea	0.7403826	6
trump	prosperity	0.7158450	7
trump	lights	0.7116018	8
trump	carolina	0.7100822	9
trump	mccarthy	0.6910413	10

Both in the republicans and democrat wordembedding "trump" appears in the recion of obama, vice and other leaders.

```
nw_dem$food |> kable()
```

term1	term2	similarity	rank
food	artists	0.7238600	1
food	funding	0.7147017	2
food	scott	0.7110178	3
food	marijuana	0.6967536	4
food	rhode	0.6682764	5
food	write	0.6494653	6
food	small	0.6461626	7
food	fallen	0.6412873	8
food	several	0.6284564	9
food	prices	0.6268474	10

```
nw_rep$food |> kable()
```

term1	term2	similarity	rank
food	ten	0.6959761	1
food	related	0.6799689	2
food	played	0.6705300	3
food	politicians	0.6569324	4
food	successful	0.6522213	5
food	double	0.6472754	6
food	vital	0.6453699	7
food	salt	0.6366376	8
food	of	0.6324419	9
food	fits	0.6275907	10

Food is connected to crisis and life related terms for both parties.

```
nw_dem$russia |> kable()
```

term1	term2	similarity	rank
russia	russian	0.8086202	1
russia	russia's	0.7030754	2
russia	presidential	0.7028244	3
russia	extent	0.6891156	4
russia	campaign	0.6806622	5
russia	sadly	0.6664057	6
russia	an	0.6564488	7
russia	obstruction	0.6548241	8
russia	credibility	0.6524557	9
russia	slash	0.6516288	10

```
nw_rep$russia |> kable()
```

term1	term2	similarity	rank
russia	directors	0.7045166	1
russia	reagan	0.6669981	2
russia	virgin	0.6520987	3
russia	virginians	0.6281113	4
russia	ambassador	0.6250147	5
russia	undo	0.6218809	6
russia	impose	0.6204441	7
russia	terror	0.6203799	8
russia	assets	0.6133589	9

term1	term2	similarity	rank
russia	days	0.6088092	10

For the term "Russia" a meaningful difference in framing is observable. Democrats associate russia with more negative terms like obstruction, or question credibility. Also the "mueller" incident discussed in a previous task is appearing here. Republicans associate russia with terms regarding the war, but not necesarry negativly. imposing, unleaching, efficiency is a neutral and almost a careful positive framing.

Task 6

```
rm(d,pretrained,corp,KING,MAN,WOMAN)

# read wordlists
pos <- readLines("positive.txt")
neg <- readLines("negative.txt")

# get embedding matrices
e_dem <- as.matrix(we_dem)
e_rep <- as.matrix(we_rep)


# Create the projection:
# 1) Extract the relevant word vectors from the "embedding" matrix and
# compute averages
posv_dem <- e_dem[which(rownames(e_dem) %in% pos),]|>
  apply(2,mean) |>
  as.matrix() |>
  t()
negv_dem <- e_dem[which(rownames(e_dem) %in% neg),]|>
  apply(2,mean) |>
  as.matrix() |>
  t()

posv_rep <- e_rep[which(rownames(e_rep) %in% pos),] |>
  apply(2,mean) |>
  as.matrix() |>
  t()
negv_rep <- e_rep[which(rownames(e_rep) %in% neg),] |>
  apply(2,mean) |>
  as.matrix() |>
  t()

# 2) Compute the difference to get the dimension
neg_pos_dem <- negv_dem - posv_dem
neg_pos_rep <- negv_rep - posv_rep
```

```
neg_assoc_terms_dem <- word2vec::word2vec_similarity(x = neg_pos_dem,
                                                     y = e_dem[-which(rownames(e_dem) %in% pos),],
                                                     top_n = 10000) # get distance to all words
neg_assoc_terms_rep <- word2vec::word2vec_similarity(x = neg_pos_rep,
                                                     y = e_rep[-which(rownames(e_rep) %in% pos),],
                                                     top_n = 10000) # get distance to all words

neg_assoc_terms_dem[,-1] |>
  head(50) |>
  kable()
```

term2	similarity	rank
misguided	0.3190255	1
notice	0.3021212	2
republican's	0.2891667	3
damaging	0.2877578	4
proposal	0.2867447	5
arbitration	0.2866713	6
hate	0.2859513	7
weaken	0.2811909	8
trumpcare	0.2809032	9
anti	0.2770302	10
recent	0.2754134	11
gop's	0.2745423	12
rejected	0.2745396	13
provisions	0.2733255	14
reached	0.2731139	15
reports	0.2727242	16
disturbing	0.2707910	17
ignore	0.2700278	18
circuit	0.2691919	19
russia	0.2689459	20
streets	0.2684063	21
preventing	0.2676252	22
advisory	0.2669555	23
response	0.2669015	24
netneutrality	0.2662738	25
threats	0.2653219	26
isn't	0.2650489	27
withdraw	0.2641229	28
strip	0.2636187	29
slash	0.2634881	30
aren't	0.2627838	31
repeal	0.2625641	32
exist	0.2619988	33
threaten	0.2610693	34

term2	similarity	rank
muslims	0.2605116	35
alone	0.2585244	36
rip	0.2582572	37
illegal	0.2577870	38
gop	0.2575555	39
result	0.2568855	40
address	0.2562891	41
score	0.2561535	42
impeachment	0.2554149	43
heartless	0.2552983	44
extreme	0.2546433	45
sale	0.2540395	46
cigarettes	0.2533462	47
insurers	0.2528137	48
mar	0.2521968	49
contracts	0.2498043	50

```
neg_assoc_terms_rep[,-1] |>
  head(50) |>
  kable()
```

term2	similarity	rank
overly	0.3297485	1
gone	0.3094015	2
billions	0.3034466	3
drop	0.2993944	4
bad	0.2915669	5
regulations	0.2848778	6
prior	0.2836224	7
ruling	0.2832588	8
backs	0.2810076	9
complex	0.2772524	10
storms	0.2750113	11
achieving	0.2700140	12
risk	0.2687067	13
attack	0.2677646	14
harvey	0.2673441	15
least	0.2654885	16
burden	0.2653474	17
certain	0.2645363	18
deported	0.2639490	19
expansion	0.2635588	20
millions	0.2629986	21
devastation	0.2619164	22

term2	similarity	rank
identify	0.2616857	23
except	0.2615484	24
crooked	0.2589501	25
severe	0.2577281	26
costly	0.2576455	27
strike	0.2555048	28
protections	0.2554087	29
aliens	0.2552176	30
highlights	0.2547450	31
seven	0.2547251	32
fires	0.2540718	33
approximately	0.2535948	34
terrorists	0.2535659	35
expensive	0.2535047	36
penalty	0.2531493	37
struggling	0.2521343	38
dropped	0.2509475	39
bureaucratic	0.2506048	40
disaster	0.2501231	41
trillion	0.2498214	42
obamacare	0.2497065	43
threaten	0.2484015	44
grocery	0.2479160	45
probation	0.2479080	46
impose	0.2478580	47
occurred	0.2468906	48
number	0.2449965	49
saving	0.2442921	50

```
word_negativity <- function(df,word){
  df[which(df$term2 == word),]
}
```

```
word_negativity(neg_assoc_terms_dem, "obama")$rank
```

```
[1] 1219
```

```
word_negativity(neg_assoc_terms_rep, "obama")$rank
```

```
[1] 409
```

"Obama" is more negativly associated in the republican embedding space compared to the democrats embedding space ("obama" has a higher rank on the constructed negativity "scale").

```
word_negativity(neg_assoc_terms_dem, "russia")$rank
```

```
[1] 20
```

```
word_negativity(neg_assoc_terms_rep, "russia")$rank
```

```
[1] 95
```

"Russia"s rank is almmost similar for both parties, but the republicans are a bit more negative about it.

```
word_negativity(neg_assoc_terms_dem, "war")$rank
```

```
[1] 1533
```

```
word_negativity(neg_assoc_terms_rep, "war")$rank
```

```
[1] 221
```

"War" has a much higher negativity rank in the republican wordembedding model. Are democrats in favour of war? Probably not.

```
kwic(toks_dem, "war", window = 5) |> kable()
```

docname	from	to	pre	keyword	post	pattern
459	17	17	and Afghanistan and other Gulf	War	veterans were exposed to the	war
535	14	14	illegally annexed Crimea engaged in	war	crimes in their bombing of	war
656	8	8	our country honored Filipino World	War	II veterans for their heroism	war
998	46	46	call to serve in World	War	II and spoke little about	war
1052	71	71	from WASP pilots in World	War	II to women in combat	war
1075	35	35	are bringing us closer to	war	with Iran	war
1087	26	26	disrespectful assertion of the Civil	War		war
1141	7	7	President questioning why the Civil	War	occurred and suggesting Andrew Jackson	war
1157	15	15	very seriously the matters of	war	and peace President Trump's proposed	war
1176	12	12	futile and seemingly endless 17-year	war	in Afghanistan is deeply disappointing	war

docname	from	to	pre	keyword	post	pattern
1176	74	74	Afghanistan has become America's longest	war	It is bleeding us to	war
1605	42	42	Protected Status for immigrants fleeing	war	disease and natural disaster This	war
1696	31	31	later the United States declared	war	on Japan and we were	war
1696	40	40	we were thrust into World	War	II Today we remember the	war
1696	72	72	remember the enormous toll that	war	has taken on our country	war
1861	29	29	Japanese internment camps during World	War	II under Executive Order Mr	war
1906	54	54	50th Anniversary of the Vietnam	War	Many who served during this	war
2107	22	22	is quietly starting a new	war	that Congress has not declared	war
2145	23	23	today Senator Dole a World	War	II veteran attended Officer Candidate	war
2174	35	35	that led to the Cold	War	With every day President-elect Trump	war
2174	102	102	arms race and new Cold	War	My op-ed with Dr Ira	war
2232	33	33	before Bashar al-Assad is a	war	criminal who has starved barrel	war
2608	9	9	and remember our prisoners of	war	those missing in action and	war
2608	56	56	Underwood's airplane crashed in World	War	II his family will be	war
2972	60	60	of Nazi fighters During World	War	II my dad's ship was	war
2975	6	6	These are dark times Terrorism	war	threats and now white supremacists	war
3140	54	54	of the South Boston World	War	I Committee honoring the legacy	war
3140	65	65	of fallen heroes in World	War	I from South Boston to	war
3140	79	79	of the South Boston Vietnam	War	Memorial to the powerful Massachusetts	war
3140	202	202	at the South Boston World	War	I Service Excel High School	war
3140	222	222	display in honor of World	War	I	war
3395	9	9	is complicit in Saudi Arabian	war	crimes and in exclusion of	war
3493	74	74	keep America from going to	war	They are also on the	war
3504	77	77	whose father was a World	War	II veteran and who serves	war
3534	130	130	it happen He also shared	war	stories from his days as	war
3634	177	177	visible and invisible wounds of	war	long after they lay down	war
3637	77	77	including an year old World	War	II veteran Join me in	war
3673	3	3	After World	War	II my mother and grandparents	war
3772	103	103	including who served in World	War	II The hall was at	war
4012	27	27	left vacant by the Civil	War	Knowing the history of his	war

docname	from	to	pre	keyword	post	pattern
4095	60	60	begin removing these weapons of	war	from our streets The first	war
4095	136	136	murdered with these weapons of	war	To my colleagues in Congress	war
4308	23	23	on Pearl Harbor in World	War	II individuals of Japanese descent	war
4448	9	9	Central Coast WWII and Korean	War	veterans on their Honor Flight	war
4656	7	7	VA Health Care System Vietnam	War	Commemoration Thank you to RimaAnn	war
4656	46	46	our country in the Vietnam	War	Their service and sacrifice will	war
4678	68	68	that the Administration's march toward	war	with North Korea is nuts	war
4678	128	128	quickly escalate to an all-out	war	in the region resulting in	war
4678	169	169	to strike the U.S homeland	War	must always be the last	war
4705	14	14	of with veterans of World	War	II and other members of	war
4713	74	74	easily escalate into a nuclear	war	with catastrophic consequences A preemptive	war
4713	186	186	without a Congressional declaration of	war	as the Constitution requires As	war
4713	195	195	requires As always diplomacy not	war	is our best chance and	war
4714	75	75	and died in every American	war	yet have no vote on	war
4809	18	18	lost this battle but the	war	sure isn't over Millions of	war
4905	41	41	New Mexico's contribution to World	War	II and our ongoing nuclear	war
4991	54	54	when and if we declare	war	Presidential authority to use the	war
5051	35	35	was drafted into the Vietnam	War	and is involved with his	war
5126	65	65	his meaningless words and his	war	on women will reverse decades	war
5179	29	29	construed as an act of	war	that my Republican colleagues would	war
5448	14	14	to make unilateral decisions on	war	sets a dangerous precedent I've	war
5456	244	244	and suffrage throughout the Civil	War	and Reconstruction periods By the	war
5502	9	9	office held the eighth Vietnam	War	50th Commemoration ceremony To date	war
5570	8	8	outrageous and disgraceful that Allied	War	Veterans would decide to ban	war
5606	46	46	the important story of the	War	of in the Chesapeake Bay	war
5668	77	77	visitors like these two Civil	War	era generals	war
5674	29	29	at Pearl Harbor During World	War	II the beacon went dark	war
5768	48	48	with the unseen wounds of	war	after their service in World	war
5768	54	54	after their service in World	War	II and I appreciate the	war

docname	from	to	pre	keyword	post	pattern
5785	162	162	that purpose consistent with the	War	Powers Act The full letter	war
5833	81	81	during the El Salvador civil	war	I am grateful to all	war
5966	41	41	their country in a global	war	I am honored they could	war
5971	44	44	Worcester Paul was a decorated	war	hero who came home and	war
6081	44	44	and rages a social media	war	with athletes he continues to	war
6105	164	164	deaths in the entire Vietnam	War	Congress must get serious about	war
6137	44	44	them heal their wounds after	war	Thank you so much for	war
6166	49	49	FBI Director and decorated Vietnam	war	hero Robert Mueller who was	war
6214	128	128	have assumed the burden of	war	on our behalf Of course	war
6235	73	73	is a group of World	War	African-American military pilots navigators mechanics	war
6502	20	20	Works today named for Korean	War	hero Thomas Hudner who was	war

A quick keyword in context (kwic) analysis shows that democrats honor heroes and vctims of war, aswell as criticising republican war involvement.