

# Economic expansions, recessions and wars – how these historic events can affect presidents’ inaugural speeches

*Ziyu Chen*

Historic events have profound impacts on today’s world politics, including US politics. World War I decimated Europe and contributed to the rise of the United States. The Great Depression crippled Germany’s economy and thus led Nazi to come to its power which in turn led to World War II. WWII shaped today’s world vastly and the Great Recession gave rise to a new setup of the world which is still ongoing.

This project aims at discovering how these historic events can affect presidents’ inaugural speeches made during those events. In this project, I focus on three types of historic events in modern history – wars, major economic recessions and economic expansions. Specifically, those events include World War I, World War II, the Great Depression, the Great Recession, the longest and the second longest economic expansions in US history.

## Step 0 - Install and load libraries

[Hide](#)[Hide](#)

```

packages.used=c("rvest", "tibble", "qdap", "ggplot2",
                "sentimentr", "gplots", "dplyr",
                "tm", "syuzhet", "factoextra",
                "scales", "RColorBrewer",
                "RANN", "tm", "topicmodels", "lubridate")
# check packages that need to be installed.
packages.needed=setdiff(packages.used,
                        intersect(installed.packages()[,1],
                                packages.used))

# install additional packages
if(length(packages.needed)>0){
  install.packages(packages.needed, dependencies = TRUE)
}
# load packages
library("rvest")
library("tibble")
library("qdap")
library("ggplot2")
library("sentimentr")
library("gplots")
library("dplyr")
library("tm")
library("syuzhet")
library("factoextra")
library("scales")
library("RColorBrewer")
library("RANN")
library("tm")
library("topicmodels")
library("lubridate")
# Set working directiry
setwd("/Users/zailchen/Documents/RWorkplace/ADS_Project_1_Ziyu_Chen")
path <- getwd()
# Source functions
source(paste0(path,"/lib/plotstacked.R"))
source(paste0(path,"/lib/speechFuncs.R"))

```

This notebook was prepared with the following environmental settings.

Hide

Hide

```
print(R.version)
```

```
platform      _
arch          x86_64-apple-darwin15.6.0
os            x86_64
system        darwin15.6.0
status        x86_64, darwin15.6.0
major         3
minor         4.3
year          2017
month         11
day           30
svn rev       73796
language      R
version.string R version 3.4.3 (2017-11-30)
nickname      Kite-Eating Tree
```

## Step 1: Data harvest and read in the speeches

Hide

Hide

```

#### Inaugural speeches
main.page <- read_html(x = "http://www.presidency.ucsb.edu/inaugurals.php")
# Get link URLs
inaug=f.speechlinks(main.page)
inaug[,1] <- as.Date(inaug[,1], format="%B %e, %Y")
inaug=inaug[-nrow(inaug),] # remove the last line, irrelevant due to error.
inaug.list=read.csv(paste0(path,"/data/inauglist.csv"), stringsAsFactors = FALSE)
inaug.list$type=c(rep("inaug", nrow(inaug.list)))
inaug.url=inaug
speech.list=cbind(inaug.list, inaugur.url)
speech.list = speech.list[ , - which(colnames(speech.list) == "Date")]
colnames(speech.list)[8] <- "Date"
# Loop over each row in speech.list
speech.list$fulltext=NA
for(i in seq(nrow(speech.list))) {
  text <- read_html(speech.list$urls[i]) %>% # load the page
    html_nodes(".displaytext") %>% # isolate the text
    html_text() # get the text
  speech.list$fulltext[i]=text
  # Create the file name
  filename <- paste0(path,"/data/fulltext/",
                    speech.list$type[i],
                    speech.list$File[i], "-",
                    speech.list$Term[i], ".txt")
  sink(file = filename) %>% # open file to write
  cat(text) # write the file
  sink() # close the file
}

```

## Step 2: Data Processing — generate list of sentences

Hide

Hide

```

sentence.list=NULL
for(i in 1:nrow(speech.list)){
  sentences=sent_detect(speech.list$fulltext[i],
                        endmarks = c("?", ".", "!", "|", ";"))
  if(length(sentences)>0){
    emotions=get_nrc_sentiment(sentences)
    word.count=word_count(sentences)
    # colnames(emotions)=paste0("emo.", colnames(emotions))
    # in case the word counts are zeros?
    emotions=diag(1/(word.count+0.01))%%as.matrix(emotions)
    sentence.list=rbind(sentence.list,
                        cbind(speech.list[i,-ncol(speech.list)],
                              sentences=as.character(sentences),
                              word.count,
                              emotions,
                              sent.id=1:length(sentences)))
  }
}
sentence.list = sentence.list %>% filter(!is.na(word.count))

```

## Step 3: Identify inaugural speeches that are made during those historic events

There are altogether 10 of those speeches.

Hide

Hide

```

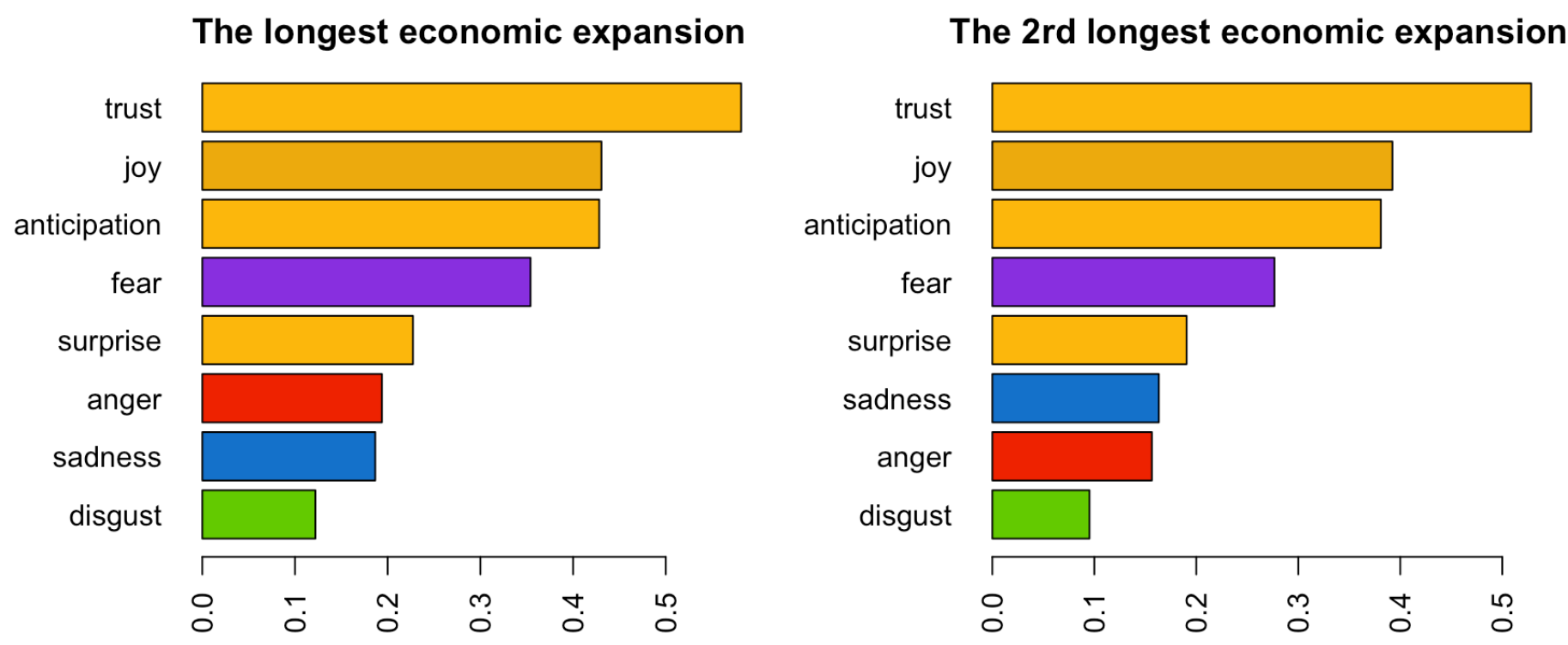
# President(s) who assumed the office during the longest period of economic expansion
in US history.
presi.expan1 <- speech.list %>%
  filter(Date %within% interval("1991-03-01", "2001-03-30")) %>%
  select(President, File, Term) %>%
  mutate(Event = "Expansion")
# President(s) who assumed the office during the second longest period of economic ex
pansion.
presi.expan2 <- speech.list %>%
  filter(Date %within% interval("1982-12-01", "1990-07-30")) %>%
  select(President, File, Term) %>%
  mutate(Event = "Expansion")
# President(s) who assumed the office during the Great Depression 1929 - 1933.
presi.recel <- speech.list %>%
  filter(Date %within% interval("1929-08-01", "1933-03-30")) %>%
  select(President, File, Term) %>%
  mutate(Event = "Recession")
# President(s) who assumed the office during the Great Recession 2007 - 2009.
presi.rece2 <- speech.list %>%
  filter(Date %within% interval("2007-12-01", "2009-06-30")) %>%
  select(President, File, Term) %>%
  mutate(Event = "Recession")
# President(s) who assumed the office during WWI.
presi.war1 <- speech.list %>%
  filter(Date %within% interval("1914-07-28", "1918-11-11")) %>%
  select(President, File, Term) %>%
  mutate(Event = "War")
# President(s) who assumed the office during WWII.
presi.war2 <- speech.list %>%
  filter(Date %within% interval("1939-09-01", "1945-09-02")) %>%
  select(President, File, Term) %>%
  mutate(Event = "War")

```

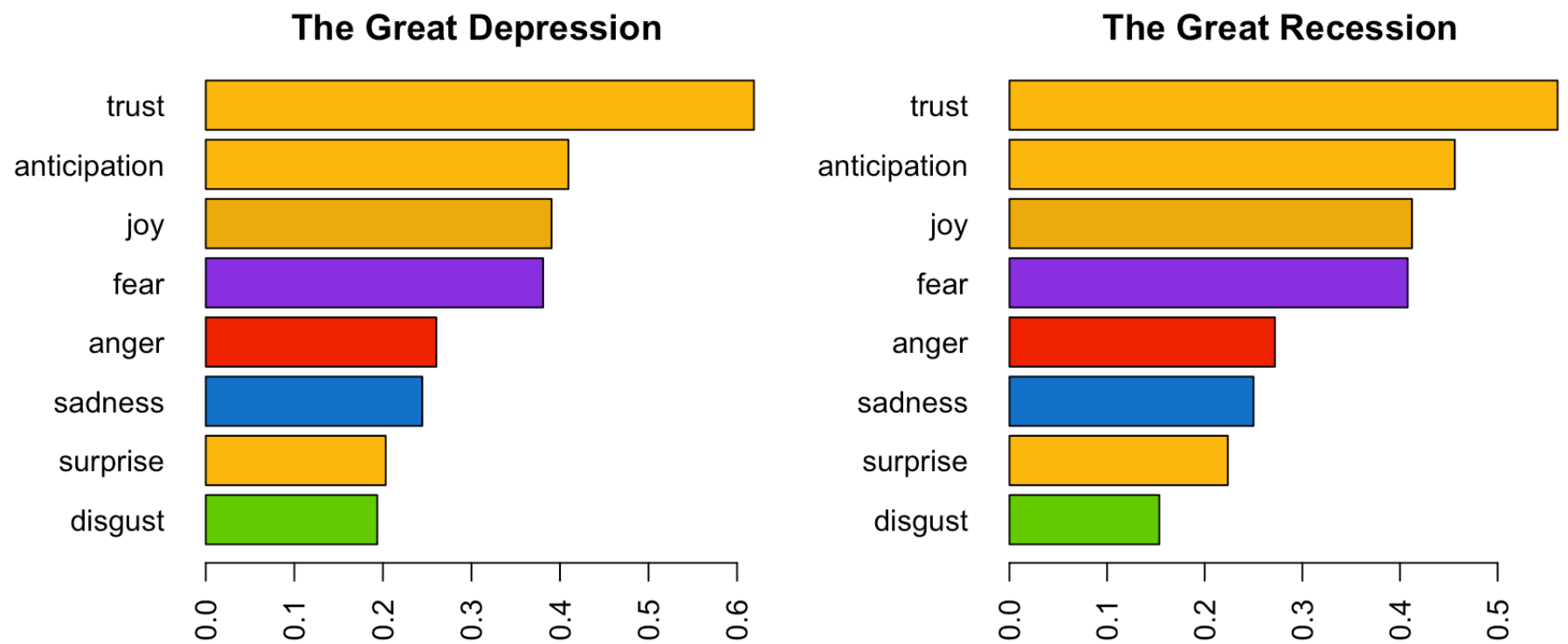
## Step 4: Sentiment analysis: Clustering of emotions

For each sentence we will apply sentiment analysis using NRC sentiment lexicon. Eight emotions will be displayed: trust, surprise, sadness, joy, fear, disgust, anticipation and anger. Those emotions are mapped with specific words which can be found in NRC sentiment lexicon. The goal of this section is to determine whether the emotions expressed from speeches made during those distinct historic events differ from each other.

```
col.use=c("red2", "darkgoldenrod1",
          "chartreuse3", "blueviolet",
          "darkgoldenrod2", "dodgerblue3",
          "darkgoldenrod1", "darkgoldenrod1")
par(mfrow=c(1,2),mar=c(4, 6, 2, 1))
# Emotion clustering of inaugural speeches made during the longest economic expansion
in history.
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.expan1[ , "President"])) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="The longest economic expansion")
# During the second longest economic expansion.
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.expan2[ , "President"])) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="The 2rd longest economic expansion")
```



```
# During the Great Depression 1929 - 1933.
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.recel[ , "President"]) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="The Great Depression")
# During the Great Recession 2007 - 2009.
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.rece2[ , "President"]) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="The Great Recession")
```

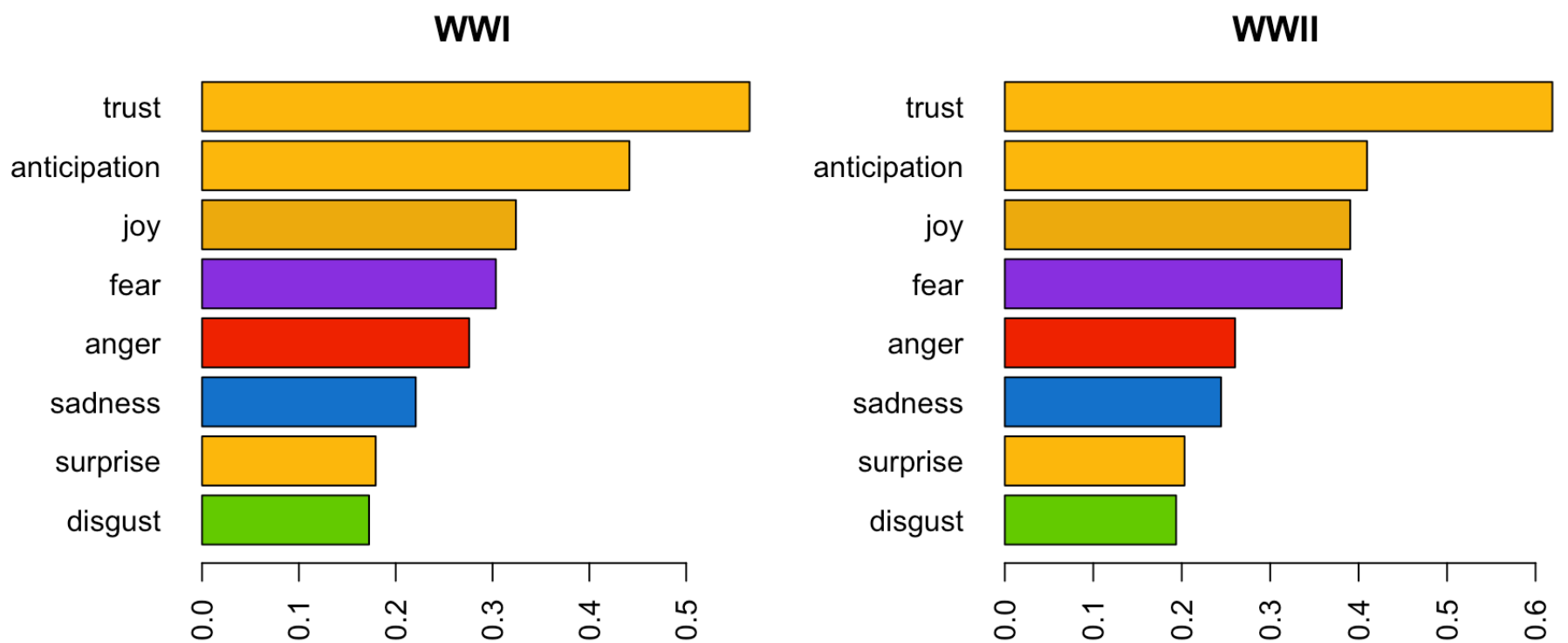


Hide

Hide

```
# During the World War I
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.war1[ , "President"]) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="WWI")
# During the World War II
emo.means=colMeans((sentence.list %>%
                    filter(President %in% presi.war2[ , "President"]) %>%
                    select(anger:trust))>0.01)
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, m
ain="WWII")
```





From the barplots above, we can easily tell that the relative importance of different emotions across all 10 speeches is quite consistent. “Trust” was significantly expressed more than any other emotions. During those major economic expansion periods, we see a lot more “joy” and “surprise” and less “fear” and “anger” overall compared to recession and war periods, which is quite reasonable.

## Step 5: Cluster speeches based on the emotion scores

Here we use K-means method to cluster speeches based on their emotion scores. The object of this step is to see whether speeches made during the same type of event can be clustered together, which may indicate a homogeneity in terms of the emotions they express. In order to obtain better results, the clustering was performed under two circumstances: across all three types of historic events and only across economic expansion and recession.

```

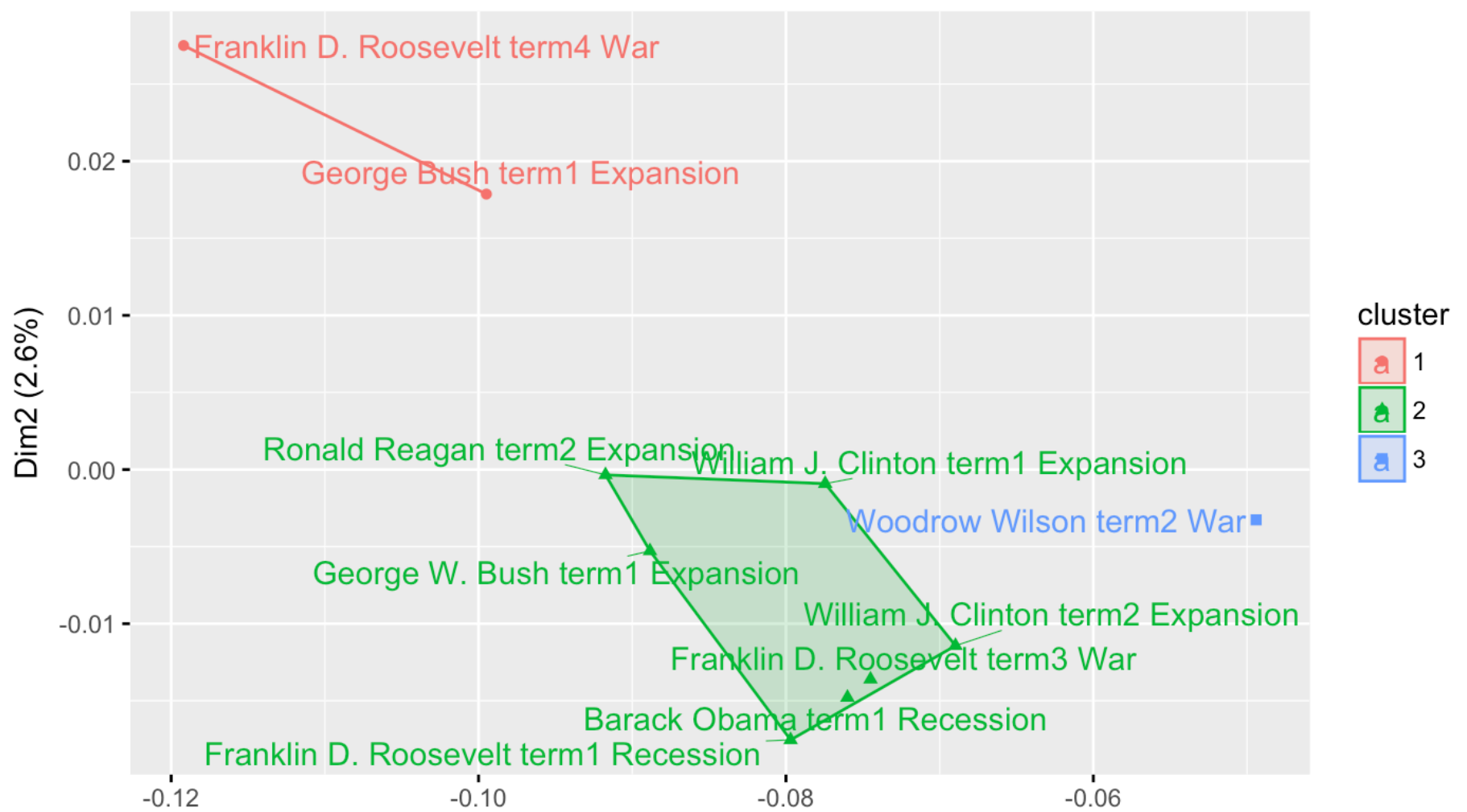
presidents <- rbind(presi.expan1, presi.expan2,
                    presi.recel, presi.rece2,
                    presi.war1, presi.war2) %>%
  mutate(President.term = paste0(President, " term " ,Term),
         President.event = paste0(President, " term",Term, " ",Event))

presid.summary=tbl_df(sentence.list) %>%
  mutate(President.term = paste0(President, " term " ,Term)) %>%
  filter(President.term %in% presidents$President.term) %>%
  group_by(President.term)%>%
  summarise(
    anger=mean(anger),
    anticipation=mean(anticipation),
    disgust=mean(disgust),
    fear=mean(fear),
    joy=mean(joy),
    sadness=mean(sadness),
    surprise=mean(surprise),
    trust=mean(trust)) %>%
  left_join(presidents, by = "President.term") %>%
  select(anger:trust, President.event)

presid.summary=as.data.frame(presid.summary)
rownames(presid.summary)=as.character((presid.summary$President.event))
r = which(colnames(presid.summary)=="President.event")
# Cluster under all three types of events: Expansion, recession, and war
km.res=kmeans(presid.summary[,-r], iter.max=200, 3)
fviz_cluster(km.res, stand=F, repel= TRUE,
              data = presid.summary[,-r], xlab="", xaxt="n",
              show.clust.cent=FALSE)

```

# Cluster plot

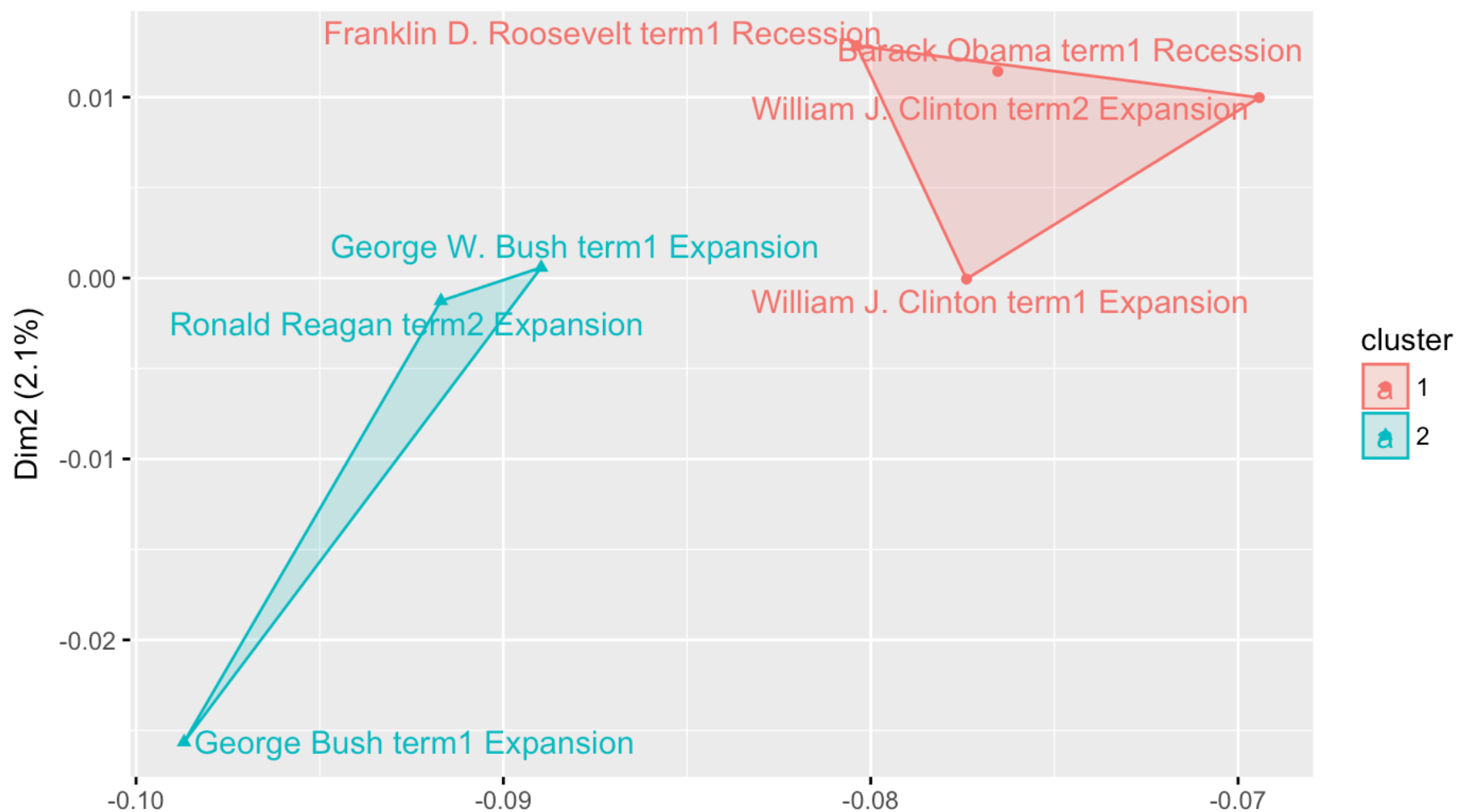


Hide

Hide

```
# Cluster only under economic expansion and recession
km.res=kmeans(presid.summary[-c(3, 4, 10), -r], iter.max=200, 2)
fviz_cluster(km.res, stand=F, repel= TRUE,
              data = presid.summary[-c(3, 4, 10), -r], xlab="", xaxt="n",
              show.clust.cent=FALSE)
```

Cluster plot



The result obtained from using  $k = 3$  and all three types of events is not informative. Obviously speeches are clustered together not according to the type of event during which they were made. However, when using  $k = 2$  and only expansions and recessions, the result seems quite promising. All speeches made during recessions are clustered together and most of speeches made during expansions are clustered together.

## Step 6: Topic modeling

Next we would like to analyze those speeches based on the differences in their topics. Here we use topic modeling. The goal is to determine if the speeches made during the same type of event are similar in their topics.

Hide

Hide

```
sentence.list.presi <- tbl_df(sentence.list) %>%
  mutate(President.term = paste0(President, " term ", Term)) %>%
  filter(President.term %in% presidents$President.term) %>%
  left_join(presidents, by = c("President", "File", "Term",
                             "President.term"))
```

# Build my own LDA function

To avoid running those extremely long LDA codes repeatedly, I created my own LDA function and write.file function.

Hide

Hide

```

LDA_Out <- function(event = NULL, df = sentence.list.presi, burnin = 4000,
                    iter = 2000, thin = 500, seed =list(2003,5,63,100001,765),
                    nstart = 5, best = TRUE, k = 10) {
  if (!is.null(event)) { df <- df %>% filter(tolower(Event) == tolower(event)) }
  corpus.list=df[2:(nrow(df)-1), ]
  sentence.pre=df$sentences[1:(nrow(df)-2)]
  sentence.post=df$sentences[3:(nrow(df)-1)]
  corpus.list$snippets=paste(sentence.pre, corpus.list$sentences, sentence.post, sep="
")
  rm.rows=(1:nrow(corpus.list))[corpus.list$sent.id==1]
  rm.rows=c(rm.rows, rm.rows-1)
  corpus.list=corpus.list[-rm.rows, ]
  ## Text mining
  docs <- Corpus(VectorSource(corpus.list$snippets))
  #writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
  ### Text basic processing
  #remove potentially problematic symbols
  docs <-tm_map(docs,content_transformer(tolower))
  #remove punctuation
  docs <- tm_map(docs, removePunctuation)
  #Strip digits
  docs <- tm_map(docs, removeNumbers)
  #remove stopwords
  docs <- tm_map(docs, removeWords, stopwords("english"))
  #remove whitespace
  docs <- tm_map(docs, stripWhitespace)
  #Stem document
  docs <- tm_map(docs,stemDocument)
  ## Topic modeling
  ###Gengerate document-term matrices.
  dtm <- DocumentTermMatrix(docs)
  #convert rownames to filenames#convert rownames to filenames
  rownames(dtm) <- paste(corpus.list$type, corpus.list$File,
                        corpus.list$Term, corpus.list$sent.id, sep="_")
  rowTotals <- apply(dtm , 1, sum) #Find the sum of words in each Document
  dtm <- dtm[rowTotals> 0, ]
  corpus.list=corpus.list[rowTotals>0, ]
  #Run LDA using Gibbs sampling
  ldaOut <-LDA(dtm, k, method="Gibbs", control=list(nstart=nstart,
                                                    seed = seed, best=best,
                                                    burnin = burnin, iter = iter,
                                                    thin=thin))

  return(ldaOut)
}

```

## Build function to write LDA files

Hide

Hide

```

write_lda_files <- function(event = NULL, Path = path) {

  ldaOut <- LDA_Out(event = event)
  #write out results
  #docs to topics
  ldaOut.topics <- as.matrix(topics(ldaOut))
  # table(c(1:k, ldaOut.topics.expan))
  write.csv(ldaOut.topics, file=paste0(Path, "/output/LDAGibbs", k, "DocsToTopics", "_", event, ".csv"))
  #top 6 terms in each topic
  ldaOut.terms <- as.matrix(terms(ldaOut, 20))
  write.csv(ldaOut.terms, file=paste0(Path, "/output/LDAGibbs", k, "TopicsToTerms", "_", event, ".csv"))
  #probabilities associated with each topic assignment
  topicProbabilities <- as.data.frame(ldaOut@gamma)
  write.csv(topicProbabilities, file=paste0(Path, "/output/LDAGibbs", k, "TopicProbabilities",
                                          "_", event, ".csv"))
}

```

## Run LDA and write files

In order to determine the differences in topics, here I run LDA separately on speeches made during different types of historic event (expansion, recession and war).

Hide

Hide

```

# Run LDA under 3 different events
ldaOut.expan <- LDA_Out(event = "expansion")
ldaOut.reces <- LDA_Out(event = "recession")
ldaOut.war <- LDA_Out(event = "war")
# Write LDA files
sapply(c("expansion", "recession", "war"), write_lda_files)

```

\$expansion

NULL

\$recession

NULL

\$war

NULL

## Step 7: LDA Results Visilization

In order to better visilize and understand the LDA results, I used ggplots. Each barplot represents a specific topic and each topic contains 20 terms. Based on the most popular terms and the most salient terms for each topic, we conclude a topic name to each topic.

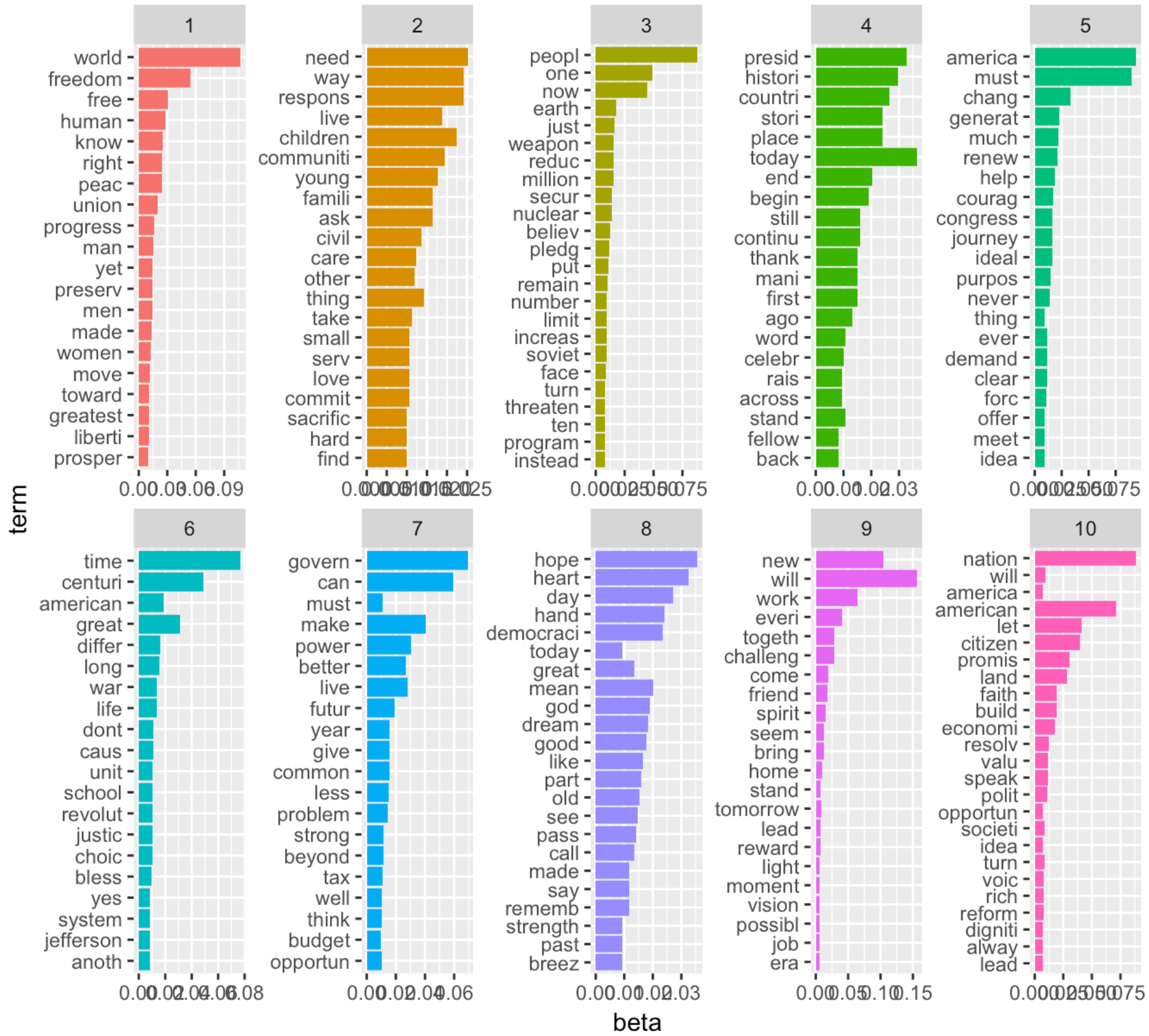
Hide

Hide

```
# During economic expansion
ldaOut.tidy.expan <- tidy(ldaOut.expan)
ldaOut.terms.expan <- as_tibble(terms(ldaOut.expan,20))
top.terms.expan <- ldaOut.tidy.expan %>%
  group_by(topic) %>%
  top_n(20, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top.terms.expan %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol=5, nrow=2) +
  ggtitle("During Economic Expansions")+
  coord_flip()
```



During Economic Expansions

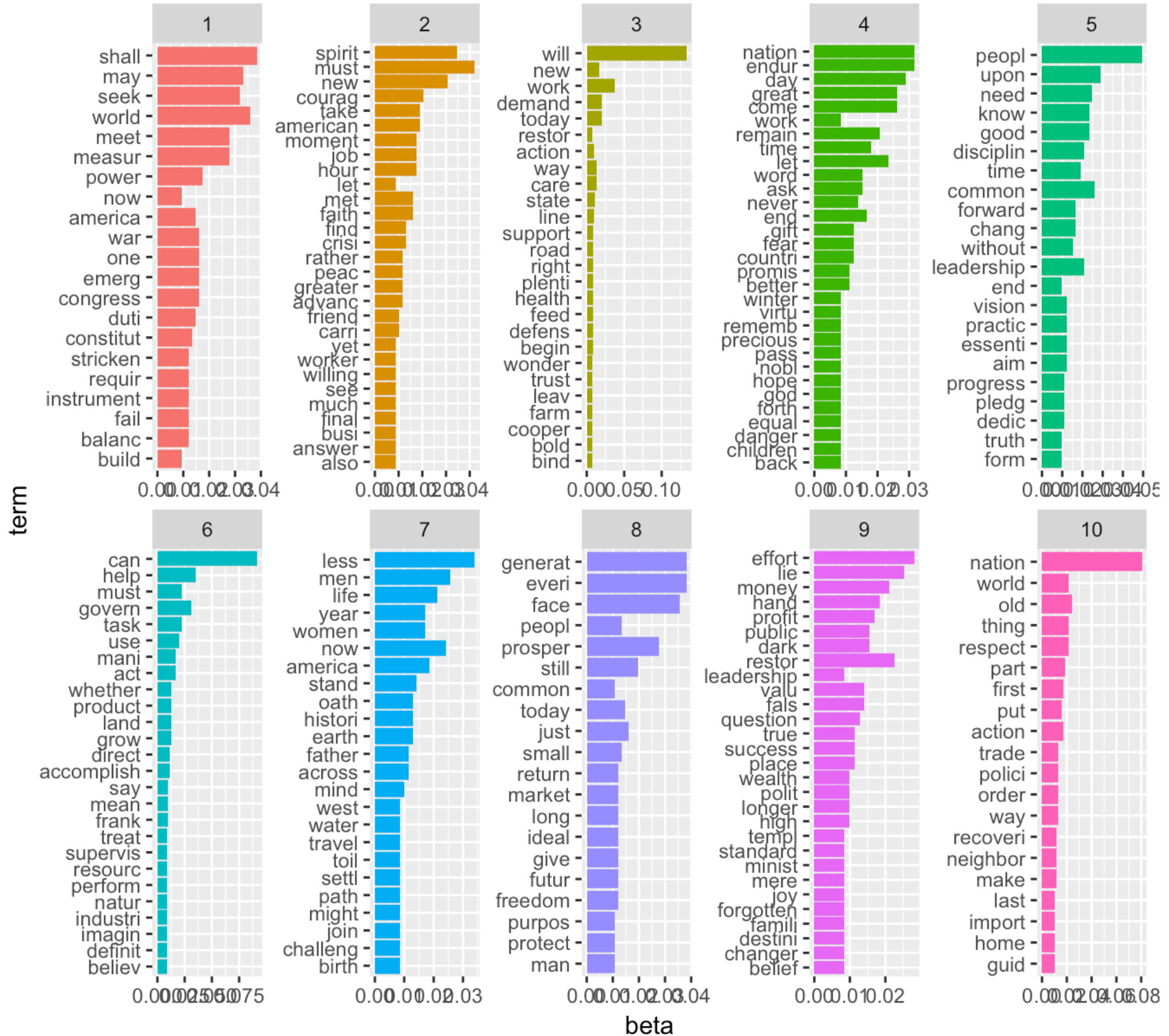


Hide

Hide

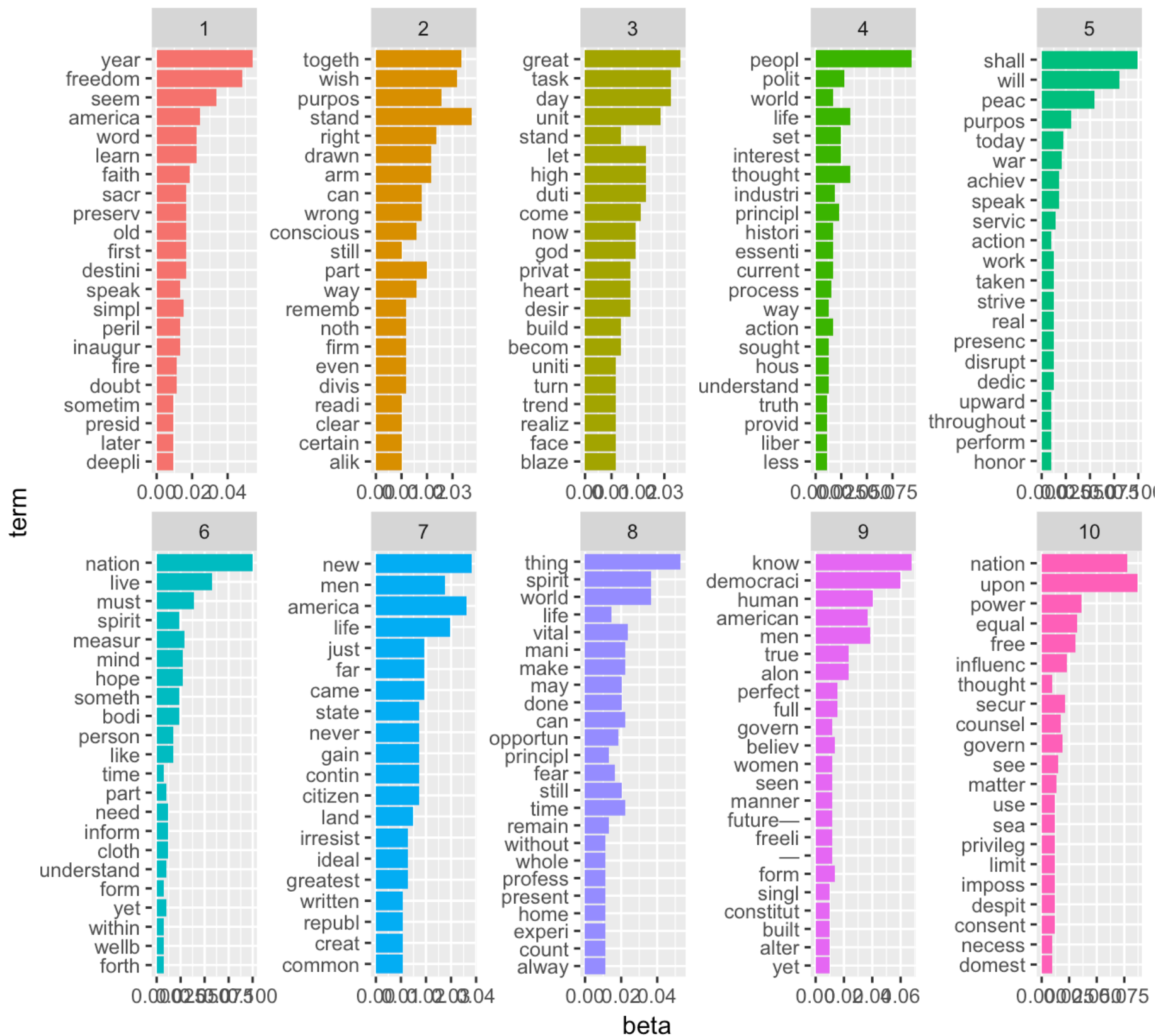
```
# During economic recession
ldaOut.tidy.reces <- tidy(ldaOut.reces)
ldaOut.terms.reces <- as_tibble(terms(ldaOut.reces,20))
top.terms.reces <- ldaOut.tidy.reces %>%
  group_by(topic) %>%
  top_n(20, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top.terms.reces %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol=5, nrow=2) +
  ggtitle("During Economic Recessions")+
  coord_flip()
```

# During Economic Recessions



```
# During wars
ldaOut.tidy.war <- tidy(ldaOut.war)
ldaOut.terms.war <- as_tibble(terms(ldaOut.war,20))
top.terms.war <- ldaOut.tidy.war %>%
  group_by(topic) %>%
  top_n(20, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top.terms.war %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol=5, nrow=2) +
  ggtitle("During Wars")+
  coord_flip()
```

## During Wars



Those LDA barplots are very informative and the differences in topics are quite salient between different types of historic events. For instance, during economic expansion, presidents like to talk about “freedom” (topic 1), “family” (topic 2), and “national security” (topic 3). Since those two major economic expansions are all during the Cold War, it does make sense presidents then like to talk about those things.

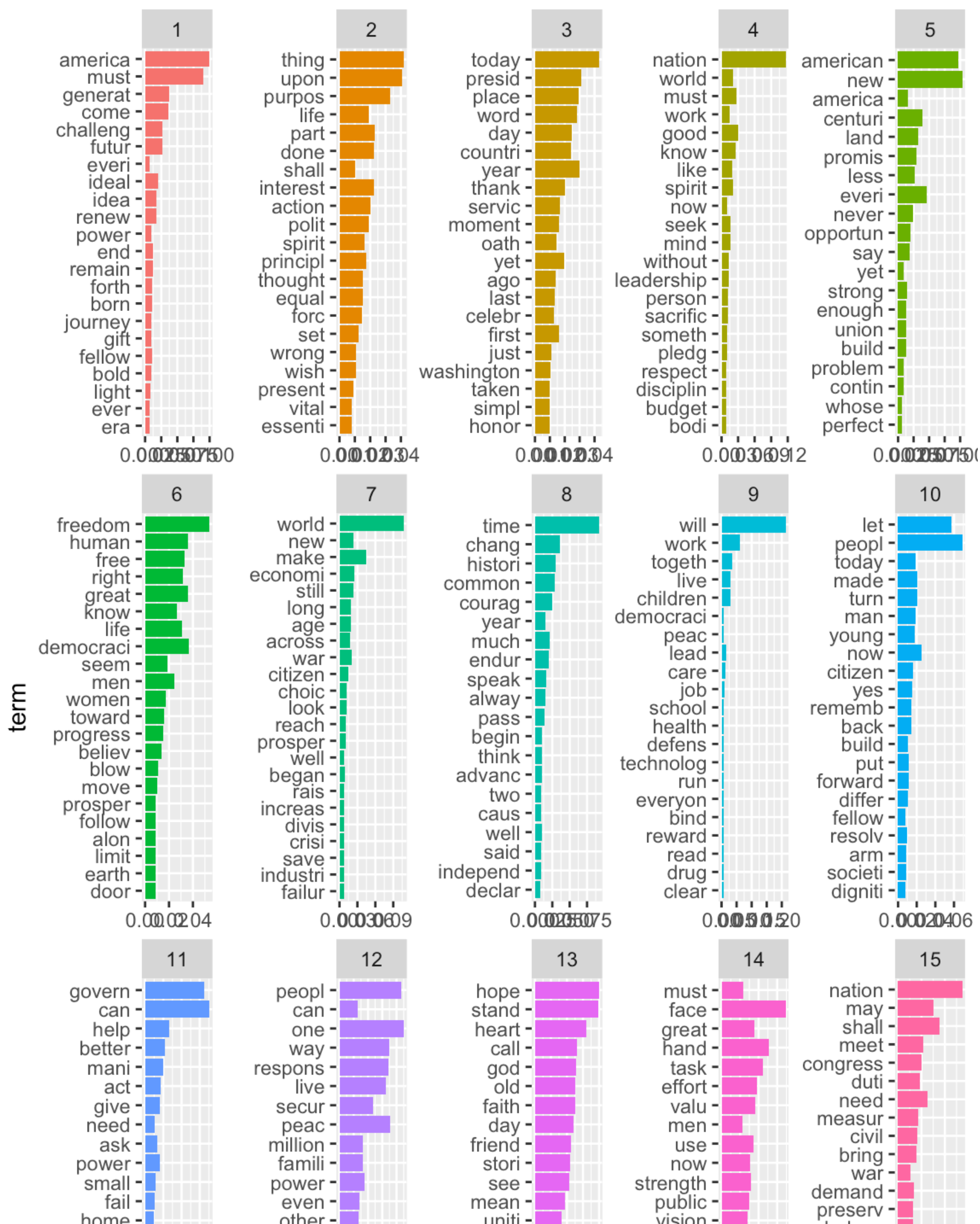
During recessions, they like to talk about “job” (topic 2), “reform” (topic 5), etc.

During wars, they like to talk about “cohesion” (topic 2), “responsibility” (topic 3), etc.

## Step 8: Perform LDA across all types of events



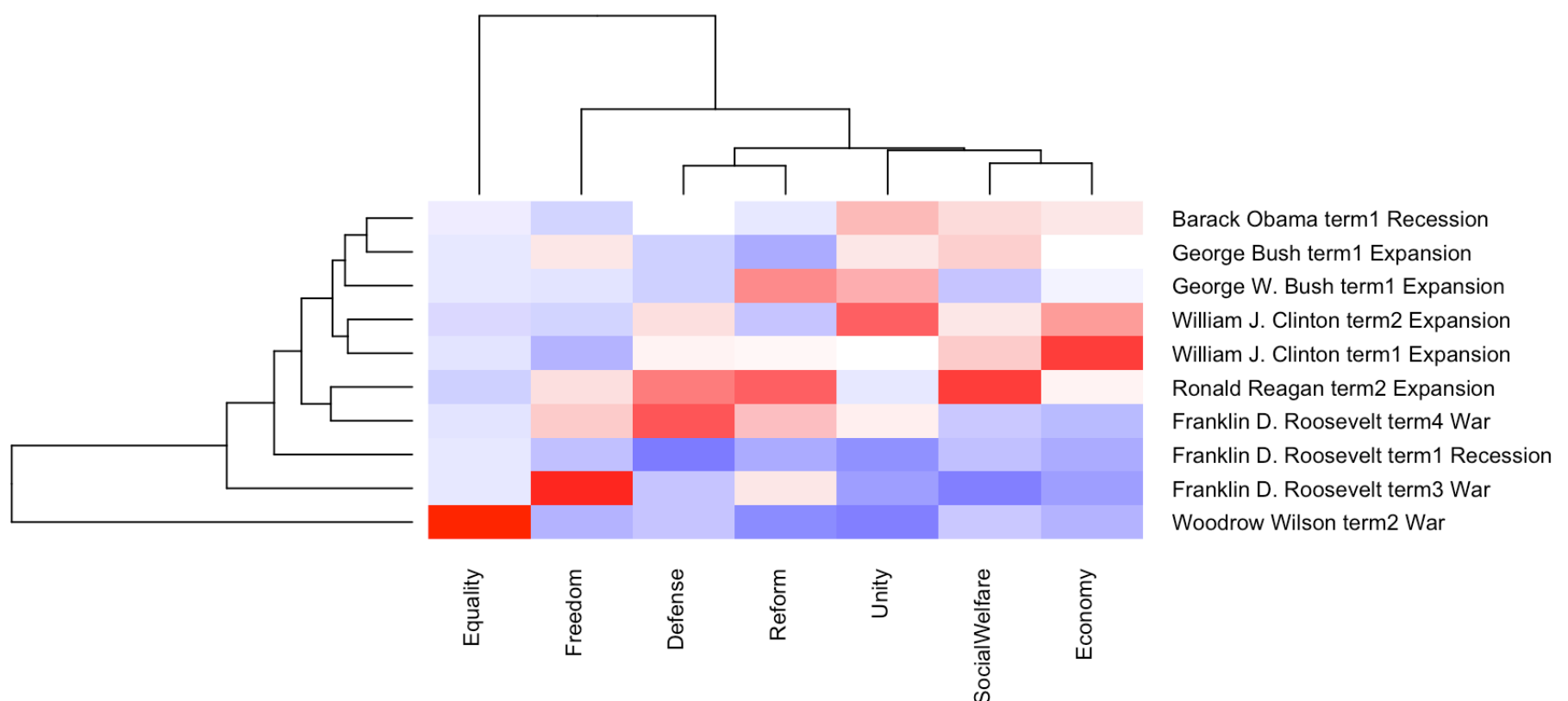
## Across all events





## Step 9: Topic clustering

Based on the LDA results, I gave all 15 topics a meaningful tag. Then I cluster the topics of those speeches together to find a pattern.



It shows that presidents who assumed the office during economic expansions like to talk about “social welfare”, “unity” and “economy” more. Presidents who assumed the office during wars preferred talking about “defense”, “equality”, and “freedom”. There is no recognizable favored topics for presidents assumed the office during recessions.

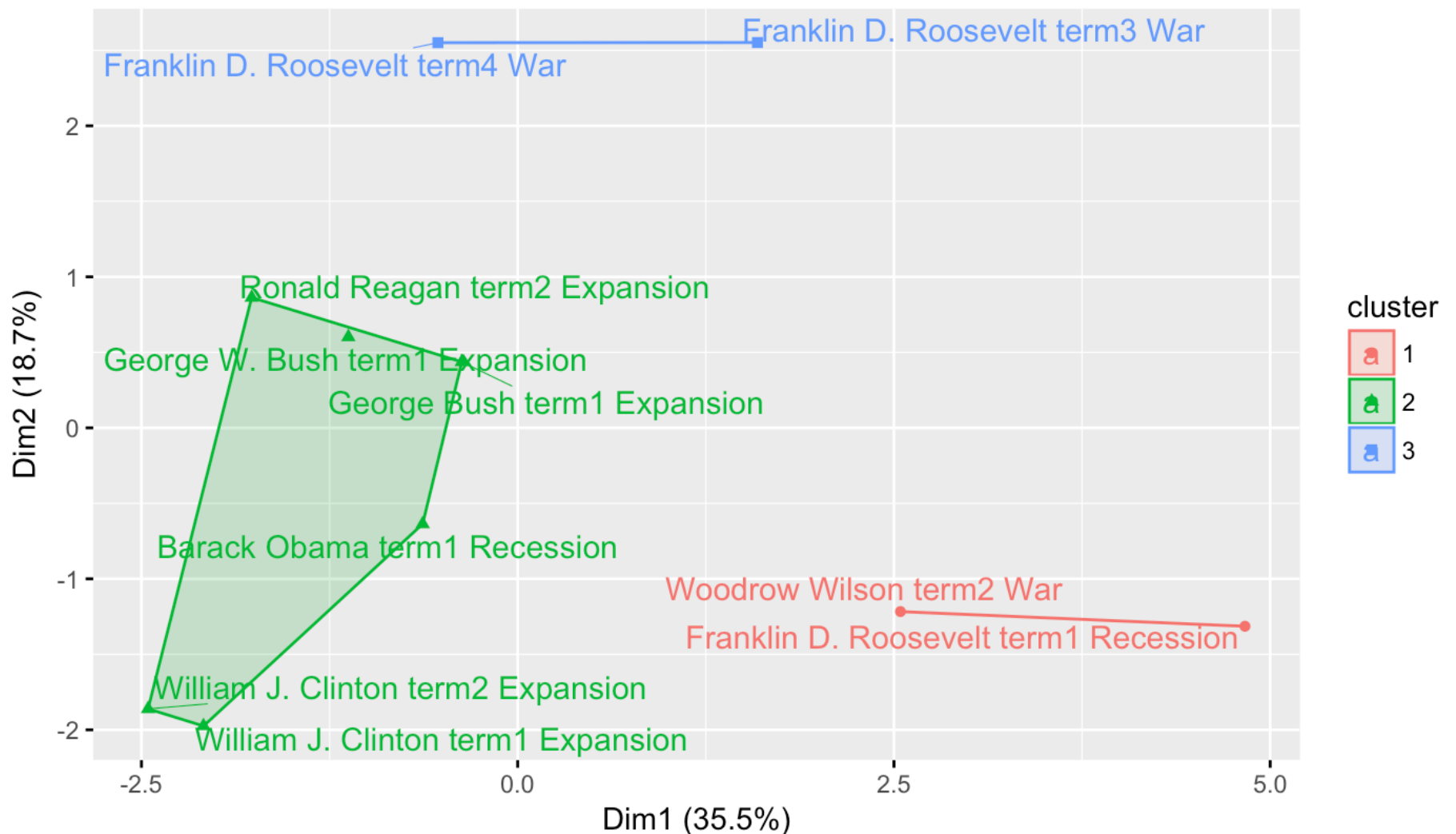
## Step 10: Cluster speeches using K-means

Lastly I want to cluster those speeches based on their topics. Here K-means method is used again. The purpose of this step is to see whether speeches made during the same type of event can be clustered together, which may indicate a homogeneity in their topics. In order to obtain better results, the clustering was performed under two circumstances: across all three types of historic events and only across economic expansion and recession.

[Hide](#)[Hide](#)

```
presid.summary=tbl_df(corpus.list.df)%>%
  mutate(President.term = paste0(President, " term " ,Term)) %>%
  select(President.term, American:Legislation) %>%
  group_by(President.term)%>%
  summarise_each(funs(mean)) %>%
  ungroup() %>%
  left_join(presidents, by = "President.term") %>%
  select(President.event, American:Legislation)
# Cluster on all three events: Expansions, recessions, and wars
presid.summary=as.data.frame(presid.summary)
rownames(presid.summary)=as.character((presid.summary[,1]))
km.res=kmeans(scale(presid.summary[,-1]), iter.max=200, 3)
fviz_cluster(km.res,
             stand=T, repel= TRUE,
             data = presid.summary[,-1],
             show.clust.cent=FALSE)
```

Cluster plot



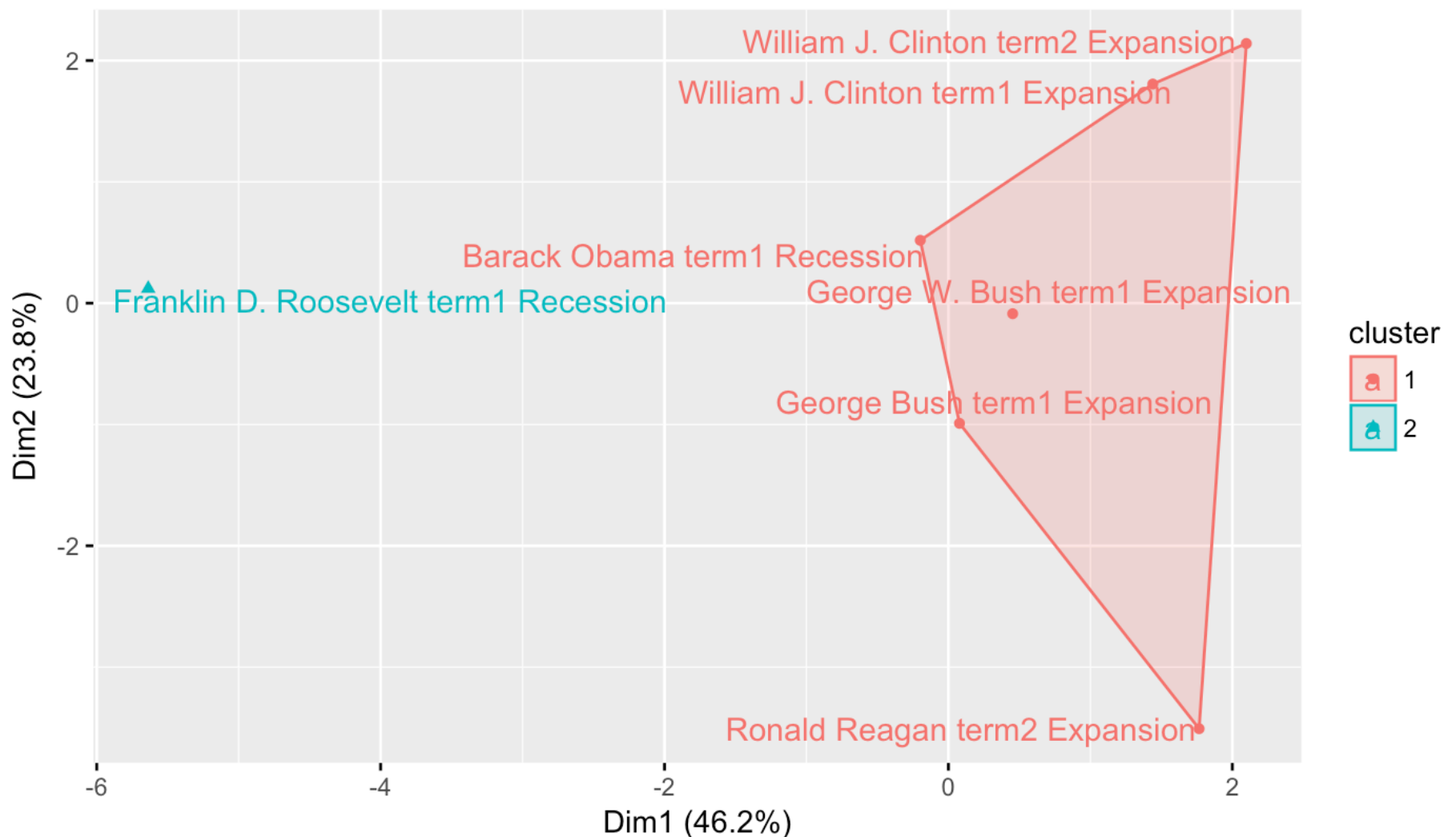


Hide

Hide

```
# Cluster on only expansions and recessions
km.res=kmeans(scale(presid.summary[-c(3,4,10),-1]), iter.max=200, 2)
fviz_cluster(km.res,
              stand=T, repel= TRUE,
              data = presid.summary[-c(3,4,10),-1],
              show.clust.cent=FALSE)
```

Cluster plot



Both cluster plots are informative. When clustering all 3 types of events, all speeches made during expansions are clustered together and most of the speech made during wars are clustered together. When clustering only on expansion and recession, there is only one mistake.

This means that the topics covered in the speeches made during economic expansions, recessions and wars are different from each other.

## Conclusion

Regardless of the types of events, the most expressed emotion in all speeches is “trust”. During economic expansion periods, there are a lot more “joy” and “surprise” and less “fear” and “anger” expressed in the inaugural speeches overall, compared to recession and war periods.

From the emotion cluster plots, we may conclude that the national economic status can have an influence on the emotions expressed in the inaugural speeches. Emotions expressed in speeches made during economic expansions and economic recessions are different. Presidents tend to talk about different topics in different historical backgrounds. For example, during economic expansion, they like to talk about “freedom”, “family”, “national security” and etc. During recessions, they like to talk about “jobs”, “reforms” (topic 5) and etc to show their future plans to revitalize economy. During wars, they like to talk about “cohesion”, “responsibility” and etc. to encourage the nation to stick together. Topic clustering plots also give the same conclusion. All speeches made during expansions are clustered together and most of the speech made during wars are clustered together. This indicates that types of events during which the speeches are made potentially can determine what topics the presidents are going to talk about in their inaugural speeches.