My Notebook

Code

check and install needed packages. Load the libraries and functions.

Hide

```
packages.used=c("rvest", "tibble", "qdap",
                "sentimentr", "gplots", "dplyr",
                "tm", "syuzhet", "factoextra",
                "beeswarm", "scales", "RColorBrewer",
                "RANN", "tm", "topicmodels")
# 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("sentimentr")
library("gplots")
library("dplyr")
library("tm")
library("syuzhet")
library("factoextra")
library("beeswarm")
library("scales")
library("RColorBrewer")
library("RANN")
library("tm")
library("topicmodels")
source("../lib/plotstacked.R")
source("../lib/speechFuncs.R")
```

Scrap speech URLs from http://www.presidency.ucsb.edu/ (http://www.presidency.ucsb.edu/).

part1:data preprocessing

What we do the first part is to grab data and then combine the data into one data.frame for future use.

Following the example of Jerid Francom (http://francojc.github.io/web-scraping-with-rvest/), we used Selectorgadget (http://selectorgadget.com/) to choose the links we would like to scrap. For this project, we selected all inaugural addresses of past presidents, nomination speeches of major party candidates and farewell addresses. We also included several public speeches from Donald Trump for our textual analysis of presidential speeches.

Hide

```
### Inauguaral speeches
main.page <- read_html(x = "http://www.presidency.ucsb.edu/inaugurals.php")
# Get link URLs
# f.speechlinks is a function for extracting links from the list of speeches.
inaug=f.speechlinks(main.page)
#head(inaug)
as.Date(inaug[,1], format="%B %e, %Y")
```

```
 [1] "1789-04-30" "1793-03-04" "1797-03-04" "1801-03-04" "1805-03-04" "1809-03-04" "1813
-03-04"
 [8] "1817-03-04" "1821-03-04" "1825-03-04" "1829-03-04" "1833-03-04" "1837-03-04" "1841
-03-04"
[15] "1845-03-04" "1849-03-05" "1853-03-04" "1857-03-04" "1861-03-04" "1865-03-04" "1869
-03-04"
[22] "1873-03-04" "1877-03-05" "1881-03-04" "1885-03-04" "1889-03-04" "1893-03-04" "1897
-03-04"
[29] "1901-03-04" "1905-03-04" "1909-03-04" "1913-03-04" "1917-03-04" "1921-03-04" "1925
-03-04"
[36] "1929-03-04" "1933-03-04" "1937-01-20" "1941-01-20" "1945-01-20" "1949-01-20" "1953
-01-20"
[43] "1957-01-21" "1961-01-20" "1965-01-20" "1969-01-20" "1973-01-20" "1977-01-20" "1981
-01-20"
[50] "1985-01-21" "1989-01-20" "1993-01-20" "1997-01-20" "2001-01-20" "2005-01-20" "2009
-01-20"
[57] "2013-01-21" "2017-01-20" NA
```

Hide

```
inaug=inaug[-nrow(inaug),] # remove the last line, irrelevant due to error.
```

Read data and then we assemble all scrapped speeches into one list.

Hide

```
inaug.list=read.csv("../data/inauglist.csv", stringsAsFactors = FALSE)
```

We assemble all scrapped speeches into one list. Note here that we don't have the full text yet, only the links to full text transcripts.

Hide

```
speech.list=inaug.list
speech.list$type=rep("inaug", nrow(inaug.list))
speech.url=inaug
speech.list=cbind(speech.list, speech.url)
```

Based on the list of speeches, we scrap the main text part of the transcript's html page. For simple html pages of this kind, Selectorgadget (http://selectorgadget.com/) is very convenient for identifying the html node that `rvest` can use to scrap its content. For reproducibility, we also save our scrapped speeches into our local folder as individual speech files.

Hide

```
# 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") %>% # isloate the text
    html_text() # get the text
  speech.list$fulltext[i]=text
  # Create the file name
  filename <- paste0("../data/fulltext/",
                     speech.list$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
}
```

Trump, as president-elect that has not been a politician, do not have a lot of formal speeches yet. For our textual analysis, we manually add several public transcripts from Trump: + [Transcript: Donald Trump's full immigration speech, annotated. LA Times, 08/31/2016] (http://www.latimes.com/politics/la-na-pol-donald-trump-immigration-speech-transcript-20160831-snap-htmlstory.html (http://www.latimes.com/politics/la-na-pol-donald-trump-immigration-speech-transcript-20160831-snap-htmlstory.html)) + Transcript of Donald Trump's speech on national security in Philadelphia - The Hill, 09/07/16 (http://thehill.com/blogs/pundits-blog/campaign/294817-transcript-of-donald-trumps-speech-on-national-security-in) + Transcript of President-elect Trump's news conference CNBC, 01/11/2017 (http://www.cnbc.com/2017/01/11/transcript-of-president-elect-donald-j-trumps-news-conference.html)

Hide

```
speech1=paste(readLines("../data/fulltext/SpeechDonaldTrump-NA.txt",
                  n=-1, skipNul=TRUE),
              collapse=" ")
speech2=paste(readLines("../data/fulltext/SpeechDonaldTrump-NA2.txt",
                  n=-1, skipNul=TRUE),
              collapse=" ")
speech3=paste(readLines("../data/fulltext/PressDonaldTrump-NA.txt",
                  n=-1, skipNul=TRUE),
              collapse=" ")
Trump.speeches=data.frame(
  President=rep("Donald J. Trump", 3),
  File=rep("DonaldJTrump", 3),
  Term=rep(0, 3),
  Party=rep("Republican", 3),
  Date=c("August 31, 2016", "September 7, 2016", "January 11, 2017"),
  Words=c(word_count(speech1), word_count(speech2), word_count(speech3)),
  Win=rep("yes", 3),
  type=rep("speeches", 3),
  links=rep(NA, 3),
  urls=rep(NA, 3),
  fulltext=c(speech1, speech2, speech3)
)
speech.list=rbind(speech.list, Trump.speeches)
```

We seperate the “fulltext” column into seperated sentences and we assign an sequential id to each sentence in a speech (`sent.id`) and also calculated the number of words in each sentence as *sentence length* (`word.count`).

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)
                              )
    )
  }
}
```

Some non-sentences exist in raw data due to erroneous extra end-of sentence marks.

Hide

```
sentence.list=
  sentence.list%>%
  filter(!is.na(word.count))
```

part2: Topic modeling

For topic modeling, we prepare a corpus of sentence snipets as follows. For each speech, we start with sentences and prepare a snipet with a given sentence with the flanking sentences.

Hide

```
corpus.list=sentence.list[2:(nrow(sentence.list)-1), ]
sentence.pre=sentence.list$sentences[1:(nrow(sentence.list)-2)]
sentence.post=sentence.list$sentences[3:(nrow(sentence.list)-1)]
corpus.list$snipets=paste(sentence.pre, corpus.list$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

Hide

```
docs <- Corpus(VectorSource(corpus.list$snipets))
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
Each moment in history is a fleeting time, precious and unique. But some stand out as mo
ments of beginning, in which courses are set that shape decades or centuries. This can b
e such a moment.
```

Text basic processing

Adapted from https://eight2late.wordpress.com/2015/09/29/a-gentle-introduction-to-topic-modeling-using-r/ (https://eight2late.wordpress.com/2015/09/29/a-gentle-introduction-to-topic-modeling-using-r/).

Hide

```
#remove potentially problematic symbols
#transform upper letter to lower letter
docs <-tm_map(docs,content_transformer(tolower))
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
not only those who are now making their tax returns, but those who meet the enhanced cos
t of existence in their monthly bills, know by hard experience what this great burden is
and what it does. no matter what others may want, these people want a drastic economy. t
hey are opposed to waste.
```

Hide

```
#remove punctuation
docs <- tm_map(docs, removePunctuation)
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
let us then fellowcitizens unite with one heart and one mind let us restore to social in
tercourse that harmony and affection without which liberty and even life itself are but
dreary things and let us reflect that having banished from our land that religious intol
erance under which mankind so long bled and suffered we have yet gained little if we cou
ntenance a political intolerance as despotic as wicked and capable of as bitter and bloo
dy persecutions
```

Hide

```
#Strip digits
docs <- tm_map(docs, removeNumbers)
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
this wont be a rally speech per se instead im going to deliver a detailed policy address
on one of the greatest challenges facing our country today illegal immigration ive just
landed having returned from a very important and special meeting with the president of m
exico a man i like and respect very much
```

Hide

```
#remove stopwords
docs <- tm_map(docs, removeWords, stopwords("english"))
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
 people   united states   failed   need   registered  mandate   want direct vigorous act
ion   asked  discipline  direction  leadership
```

Hide

```
#remove whitespace
docs <- tm_map(docs, stripWhitespace)
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
 will america uncrossed desert unclimbed ridge
```

Hide

```
#Stem document
#worked working work are the same meaning, this is what this step is doing
docs <- tm_map(docs,stemDocument)
writeLines(as.character(docs[[sample(1:nrow(corpus.list), 1)]]))
```

```
sustain educ advanc knowledg growth religi spirit toler faith strengthen home
```

Topic modeling

Gengerate document-term matrices.

Hide

```
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

Hide

```
#Set parameters for Gibbs sampling
burnin <- 4000 # numbers we do not need
iter <- 2000
thin <- 500
seed <-list(2003,5,63,100001,765) # seed to start mcmc,  reproducible
nstart <- 5
best <- TRUE
#Number of topics
k <- 20 #number of topics in anaylsis(may be a guess)
#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))
#write out results
#docs to topics
ldaOut.topics <- as.matrix(topics(ldaOut))
table(c(1:k, ldaOut.topics))
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
319 783 313 292 280 378 483 472 307 297 399 304 413 211 305 315 251 210 199 245
```

Hide

```
write.csv(ldaOut.topics,file=paste("../out/LDAGibbs",k,"DocsToTopics.csv"))
#top 6 terms in each topic
ldaOut.terms <- as.matrix(terms(ldaOut,20))
write.csv(ldaOut.terms,file=paste("../out/LDAGibbs",k,"TopicsToTerms.csv"))
#probabilities associated with each topic assignment
topicProbabilities <- as.data.frame(ldaOut@gamma)
write.csv(topicProbabilities,file=paste("../out/LDAGibbs",k,"TopicProbabilities.csv"))
```

Hide

```
terms.beta=ldaOut@beta
terms.beta=scale(terms.beta)
topics.terms=NULL
for(i in 1:k){
  topics.terms=rbind(topics.terms, ldaOut@terms[order(terms.beta[i,], decreasing = TRUE)
[1:7]])
}
topics.terms
```

```
      [,1]        [,2]        [,3]         [,4]          [,5]           [,6]         [,
7]
 [1,] "deleg"     "concentr"  "junctur"    "overflow"    "forthsometim" "unadjust"   "im
bib"
 [2,] "happen"    "realli"    "russia"     "big"         "didnt"        "got"        "th
ere"
 [3,] "construct" "naval"     "rapid"      "canal"       "fortif"       "correspond" "in
terior"
 [4,] "aim"       "woe"       "prerog"     "cancer"      "waver"        "token"      "so
litari"
 [5,] "fals"      "bosom"     "profess"    "genuin"      "templ"        "inher"      "ar
istocraci"
 [6,] "thrive"    "asham"     "infirm"     "exil"        "comput"       "shift"      "sh
utter"
 [7,] "communism" "proven"    "hunger"     "unselfish"   "ideolog"      "steadfast"  "be
neath"
 [8,] "dream"     "truli"     "willing"    "bright"      "imag"         "edg"        "de
pth"
 [9,] "separ"     "texa"      "endang"     "speedili"    "dissolut"     "confin"     "th
irteen"
[10,] "promin"    "incompet"  "dilig"      "civilservic" "unreserv"     "falsehood"  "un
aid"
[11,] "immigr"    "border"    "clinton"    "crimin"      "alien"        "terror"     "—"

[12,] "celebr"    "planet"    "bush"       "weari"       "recit"        "miss"       "fl
eet"
[13,] "industri"  "revenu"    "expenditur" "tariff"      "surplus"      "farm"       "co
ntract"
[14,] "prefer"    "purchas"   "equit"      "pretext"     "discern"      "worship"    "id
l"
[15,] "gratitud"  "divin"     "ardent"     "sensibl"     "vicissitud"   "flatter"    "em
ot"
[16,] "jurisdict" "urg"       "obey"       "judiciari"   "veto"         "railway"    "su
premaci"
[17,] "technic"   "inexhaust" "accid"      "tumult"      "insidi"       "symptom"    "co
miti"
[18,] "south"     "suffrag"   "southern"   "color"       "voter"        "perish"     "fu
rther"
[19,] "stake"     "cure"      "bar"        "rage"        "savag"        "dissent"    "di
ssatisfi"
[20,] "neutral"   "attitud"   "britain"    "undisturb"   "diplomat"     "inflex"     "fa
roff"
```

Hide

```
ldaOut.terms
```

```
      Topic 1      Topic 2  Topic 3     Topic 4      Topic 5   Topic 6    Topic 7     To
pic 8
 [1,] "govern"     "want"   "war"       "will"       "spirit"  "new"      "world"     "a
merica"
 [2,] "power"      "mani"   "great"     "make"       "liberti" "work"     "freedom"   "l
et"
 [3,] "general"    "know"   "made"      "now"        "peopl"   "live"     "peac"      "a
merican"
 [4,] "limit"      "say"    "forc"      "good"       "much"    "american" "free"      "t
ogeth"
 [5,] "feder"      "think"  "maintain"  "believ"     "mean"    "old"      "human"     "g
od"
 [6,] "peopl"      "thing"  "necessari" "futur"      "life"    "home"     "men"       "f
aith"
 [7,] "within"     "dont"   "resourc"   "past"       "still"   "better"   "seek"      "g
enerat"
 [8,] "upon"       "back"   "might"     "lead"       "success" "must"     "help"      "h
eart"
 [9,] "system"     "like"   "may"       "bring"      "everi"   "find"     "know"      "h
ope"
[10,] "experi"     "just"   "commerc"   "understand" "prosper" "land"     "nation"    "e
nd"
[11,] "exercis"    "get"    "progress"  "follow"     "republ"  "opportun" "strength"  "w
ord"
[12,] "instrument" "great"  "armi"      "greatest"   "continu" "children" "hope"      "c
all"
[13,] "object"     "way"    "proper"    "requir"     "yet"     "need"     "man"       "c
ourag"
[14,] "concern"    "look"   "prepar"    "sacrific"   "caus"    "way"      "long"      "d
ream"
[15,] "grant"      "happen" "import"    "hard"       "true"    "make"     "democraci" "t
oday"
[16,] "observ"     "theyr"  "improv"    "effort"     "love"    "everi"    "forc"      "v
oic"
[17,] "establish"  "meet"   "defens"    "rather"     "chang"   "respons"  "mankind"   "p
romis"
[18,] "respect"    "lot"    "without"   "save"       "done"    "reach"    "achiev"    "b
less"
[19,] "author"     "thank"  "civil"     "possibl"    "remain"  "famili"   "progress"  "a
cross"
[20,] "direct"     "deal"   "militari"  "step"       "fail"    "see"      "earth"     "f
orward"
      Topic 9     Topic 10   Topic 11   Topic 12  Topic 13     Topic 14   Topic 15
  Topic 16
 [1,] "state"     "shall"    "countri"  "time"    "industri"   "right"    "countri"
  "law"
 [2,] "unit"      "duti"     "immigr"   "year"    "public"     "citizen"  "may"
  "constitut"
 [3,] "union"     "public"   "also"     "day"     "busi"       "upon"     "confid"
  "congress"
 [4,] "constitut" "offic"    "take"     "presid"  "product"    "equal"    "honor"
  "execut"
 [5,] "whole"     "servic"   "million"  "now"     "revenu"     "everi"    "patriot"
```

```
  "may"
 [6,] "territori" "respons"  "number"   "first"   "may"        "just"     "fellowcitize
n" "legisl"
 [7,] "form"      "best"     "defens"   "today"   "demand"     "secur"    "circumst"
  "elect"
 [8,] "exist"     "call"     "illeg"    "come"    "tax"        "individu" "high"
  "upon"
 [9,] "institut"  "faith"    "border"   "centuri" "economi"    "principl" "whose"
  "effect"
[10,] "extend"    "oblig"    "mani"     "histori" "secur"      "well"     "happi"
  "enforc"
[11,] "part"      "execut"   "plan"     "stand"   "money"      "act"      "experi"
  "author"
[12,] "preserv"   "support"  "system"   "chang"   "condit"     "found"    "conduct"
  "pass"
[13,] "domest"    "principl" "hillari"  "moment"  "protect"    "support"  "exampl"
  "case"
[14,] "import"    "give"     "clinton"  "last"    "expenditur" "protect"  "great"
  "rule"
[15,] "object"    "civil"    "american" "ago"     "practic"    "rest"     "occas"
  "decis"
[16,] "independ"  "upon"     "ask"      "begin"   "trade"      "whether"  "intellig"
  "subject"
[17,] "term"      "trust"    "open"     "fellow"  "pay"        "other"    "devot"
  "depart"
[18,] "popul"     "purpos"   "crimin"   "oath"    "increas"    "fair"     "express"
  "major"
[19,] "perfect"   "pledg"    "remov"    "sinc"    "labor"      "enjoy"    "degre"
  "recommend"
[20,] "consequ"   "perform"  "includ"   "justic"  "debt"       "privileg" "everi"
  "amend"
      Topic 17   Topic 18   Topic 19  Topic 20
 [1,] "must"     "one"      "can"     "nation"
 [2,] "peopl"    "countri"  "peopl"   "polici"
 [3,] "need"     "parti"    "never"   "peac"
 [4,] "use"      "polit"    "nation"  "interest"
 [5,] "nation"   "question" "great"   "foreign"
 [6,] "purpos"   "differ"   "hand"    "respect"
 [7,] "common"   "anoth"    "without" "relat"
 [8,] "order"    "interest" "strong"  "countri"
 [9,] "problem"  "good"     "ever"    "among"
[10,] "moral"    "member"   "place"   "cours"
[11,] "justic"   "well"     "even"    "intern"
[12,] "task"     "opinion"  "fear"    "friend"
[13,] "knowledg" "feel"     "alway"   "desir"
[14,] "high"     "race"     "less"    "honor"
[15,] "sens"     "mere"     "reason"  "justic"
[16,] "among"    "may"      "present" "determin"
[17,] "econom"   "can"      "side"    "wish"
[18,] "develop"  "seem"     "natur"   "promot"
[19,] "hold"     "section"  "littl"   "independ"
[20,] "realiz"   "south"    "weak"    "best"
```

Based on the most popular terms and the most salient terms for each topic, we assign a hashtag to each topic. It is easy to find that the topic 13 contains so many words about economy, such as tax, debt, pay, etc. And the topic 3 contains many words about military, such as war, force, military, etc. So we can definitely define the topic of 13 and 3 are economy and military, and then we use the first common words of other topics as their hashtag. And this is what we do next.

Hide

```
topics.hash=as.vector(ldaOut.terms[1, ])
topics.hash[13] <- "economy"
topics.hash[15] <- "lucky"
topics.hash[3] <- "military"
corpus.list$ldatopic=as.vector(ldaOut.topics)
corpus.list$ldahash=topics.hash[ldaOut.topics]
colnames(topicProbabilities)=topics.hash
corpus.list.df=cbind(corpus.list, topicProbabilities)
```

Clustering of topics

Hide

```
par(mar=c(1,1,1,1))
topic.summary=tbl_df(corpus.list.df)%>%
              select(File, govern:nation)%>%
              group_by(File)%>%
              summarise_each(funs(mean))
```

```
`summarise_each()` is deprecated.
Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
To map `funs` over all variables, use `summarise_all()`
```

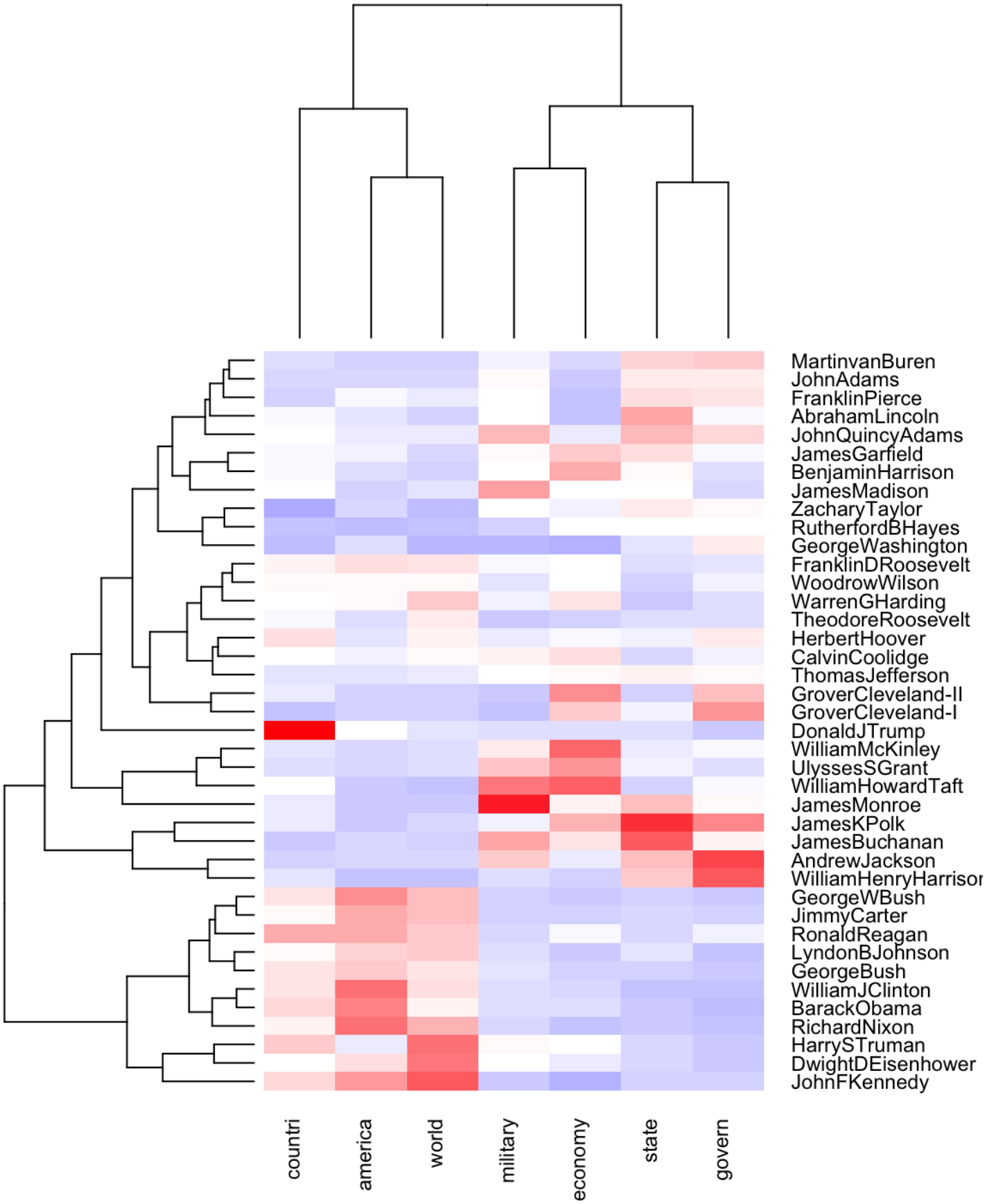
Hide

```
topic.summary=as.data.frame(topic.summary)
rownames(topic.summary)=topic.summary[,1]
topic.plot=c(1, 13, 9, 11, 8, 3, 7)
print(topics.hash[topic.plot])
```

```
[1] "govern"   "economy"  "state"    "countri"  "america"  "military" "world"
```

Hide

```
heatmap.2(as.matrix(topic.summary[,topic.plot+1]),
          scale = "column", key=F,
          col = bluered(100),
          cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
          trace = "none", density.info = "none")
```


MartinvanBuren
JohnAdams
FranklinPierce
AbrahamLincoln
JohnQuincyAdams
JamesGarfield
BenjaminHarrison
JamesMadison
ZacharyTaylor
RutherfordBHayes
GeorgeWashington
FranklinDRoosevelt
WoodrowWilson
WarrenGHarding
TheodoreRoosevelt
HerbertHoover
CalvinCoolidge
ThomasJefferson
GroverCleveland-II
GroverCleveland-I
DonaldJTrump
WilliamMcKinley
UlyssesSGrant
WilliamHowardTaft
JamesMonroe
JamesKPolk
JamesBuchanan
AndrewJackson
WilliamHenryHarrisor
GeorgeWBush
JimmyCarter
RonaldReagan
LyndonBJohnson
GeorgeBush
WilliamJClinton
BarackObama
RichardNixon
HarrySTruman
DwightDEisenhower
JohnFKennedy
countri
america
world
military
economy
state
govern

Now we take five recent presidents to make this question simpler. The most recent 5 presidents are GeorgeBush, WilliamJClinton, GeorgeWBush, BarackObama, DonaldJTrump. Now we make a plot to see which topic they pay attention to more.

Hide

```
par(mfrow=c(5, 1), mar=c(1,1,2,0), bty="n", xaxt="n", yaxt="n")
topic.plot=c(3, 13, 14, 15, 8, 9, 12)
print(topics.hash[topic.plot])
```

```
[1] "military" "economy"  "right"    "lucky"    "america"  "state"    "time"
```

Hide

```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="GeorgeBush", type=="inaug", Term==1)%>%
select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="George Bush, inaugural Speeche
s")
```

```
[1] 0.04472720 0.08945439 0.13418159 0.17890878 0.22363598 0.26836318 0.31309037
```

Hide

```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="WilliamJClinton", type=="inaug", Term==
1)%>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="William J Clinton, inaugural Sp
eeches")
```

```
[1] 0.04653387 0.09306773 0.13960160 0.18613547 0.23266933 0.27920320 0.32573706
```

Hide

```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="GeorgeWBush", type=="inaug", Term==1)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="George W. Bush, inaugural Speec
hes")
```

```
[1] 0.04136933 0.08273866 0.12410799 0.16547732 0.20684666 0.24821599 0.28958532
```

Hide

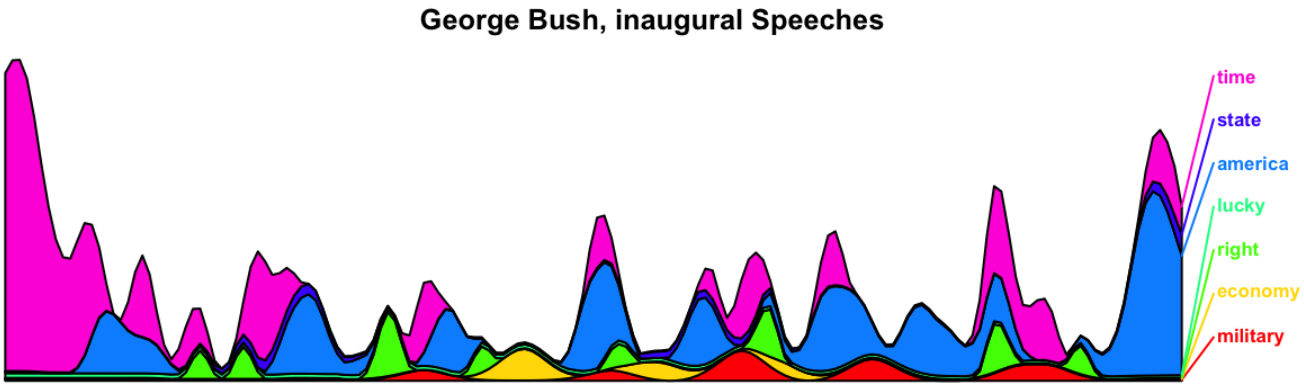
```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="BarackObama", type=="inaug", Term==1)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="Barack Obama, inaugural Speeche
s")
```

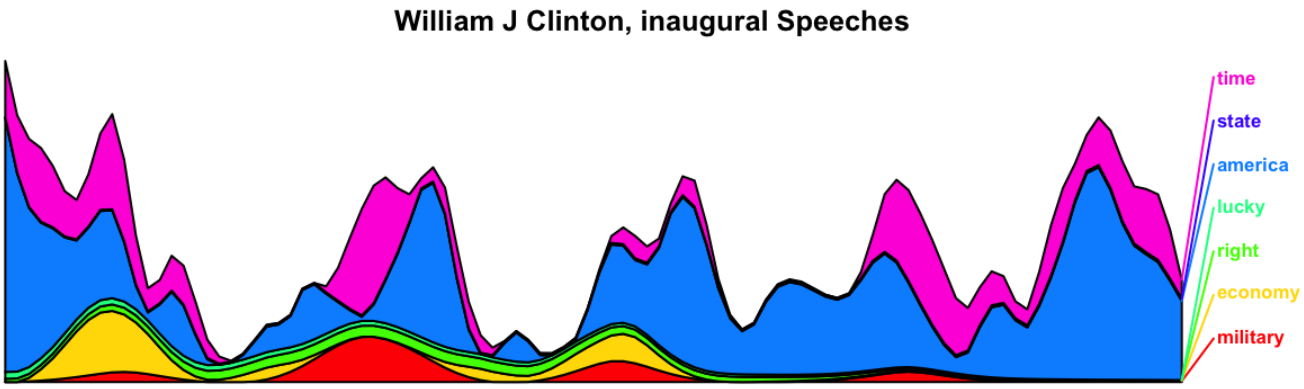
```
[1] 0.04448455 0.08896909 0.13345364 0.17793819 0.22242274 0.26690728 0.31139183
```

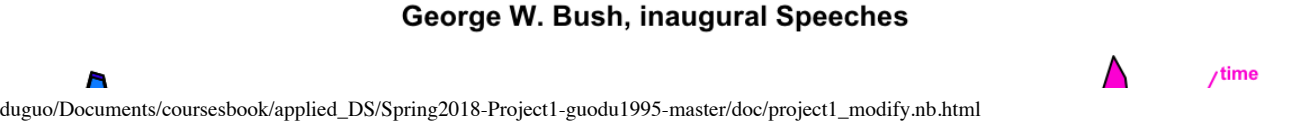
Hide

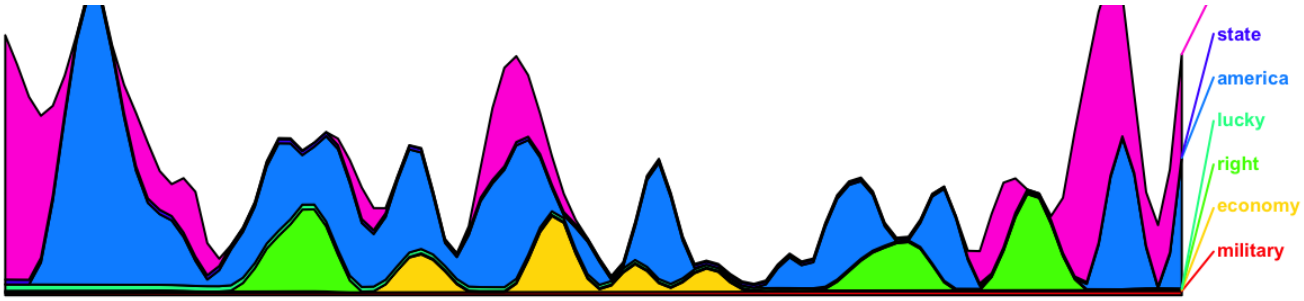
```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="DonaldJTrump", type=="inaug", Term==1)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="Donald J. Trump, inaugural Spee
ches")
```

```
[1] 0.04306823 0.08613646 0.12920469 0.17227292 0.21534115 0.25840938 0.30147761
```

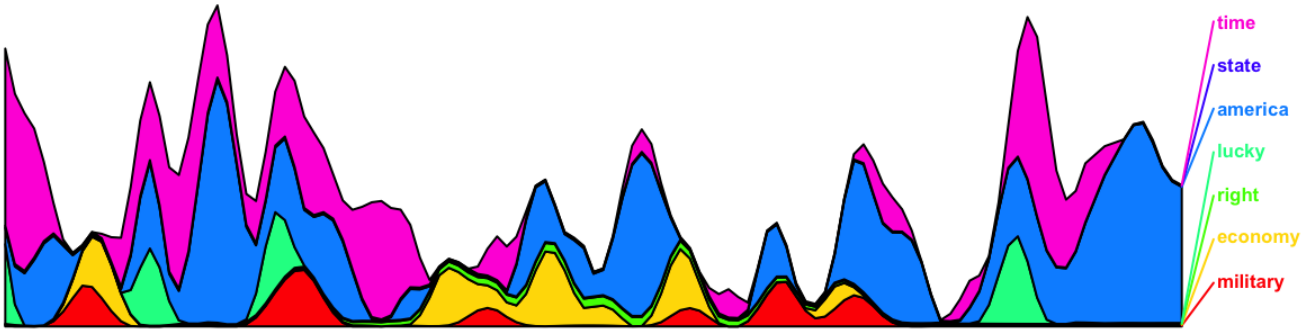




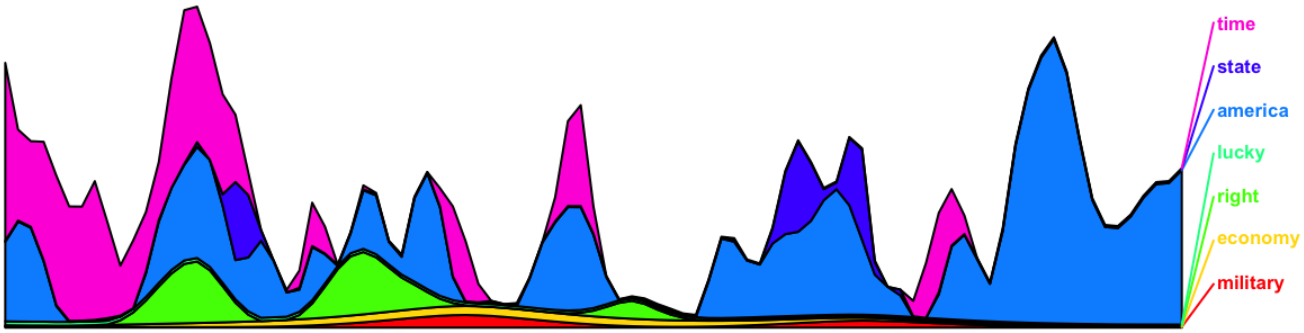




Barack Obama, inaugural Speeches



Donald J. Trump, inaugural Speeches



From the plot, we can find that Barack Obama's fist inaugural topic contains much more "economy" than other presidents, but why? As we know, Global financial crisis happend in 2008, and Obama became the president in 2009. So he needed to take some economical strategies to revive American's economy. Someone may ask that Barack Obama talked more about economy beacause he prefered to pay attention to American economy, but not because the 2008 Global financial crisis happened. We can make a further plot to compare the 1st term inaugural speech and 2nd term inaugural speech of Barack Obama to see the specific reason.

Hide

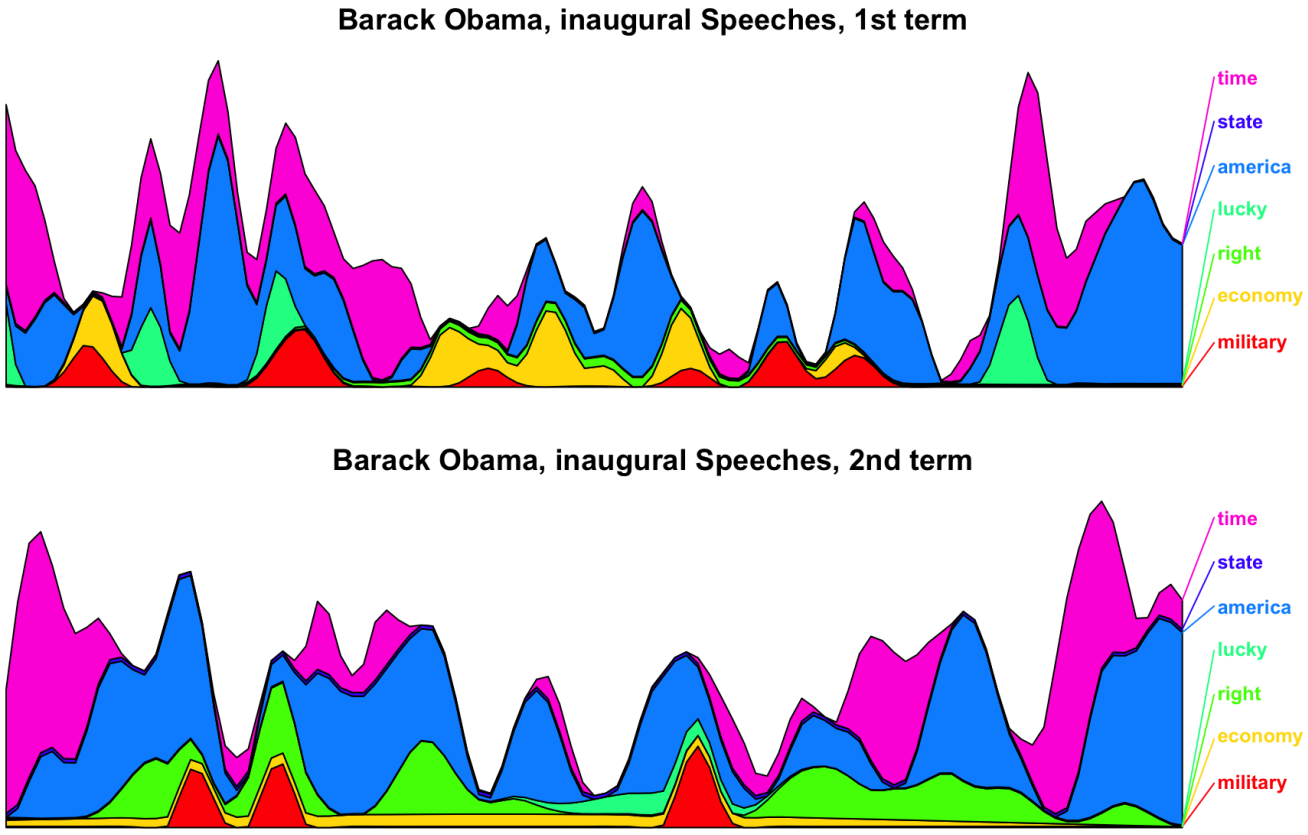
```
par(mfrow=c(2, 1), mar=c(1,1,2,0), bty="n", xaxt="n", yaxt="n")
speech.df=tbl_df(corpus.list.df)%>%filter(File=="BarackObama", type=="inaug", Term==1)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="Barack Obama, inaugural Speeche
s, 1st term")
```

```
[1] 0.04448455 0.08896909 0.13345364 0.17793819 0.22242274 0.26690728 0.31139183
```

Hide

```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="BarackObama", type=="inaug", Term==2)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="Barack Obama, inaugural Speeche
s, 2nd term")
```

```
[1] 0.03867174 0.07734348 0.11601523 0.15468697 0.19335871 0.23203045 0.27070220
```



From this plot, we find that compared to the first inaugural speech, Barack Obama talked much less about economy. So we know that the reason Barack Obama talked much about economy is that he needed to take some positive economy strategy in response of the global financial crisis. So, In other words, can we say that some big events can affect the topic of presidents' inaugural speech? Now, let's make a graph compared the first and second inaugural speech of GeorgeWBush.

Hide

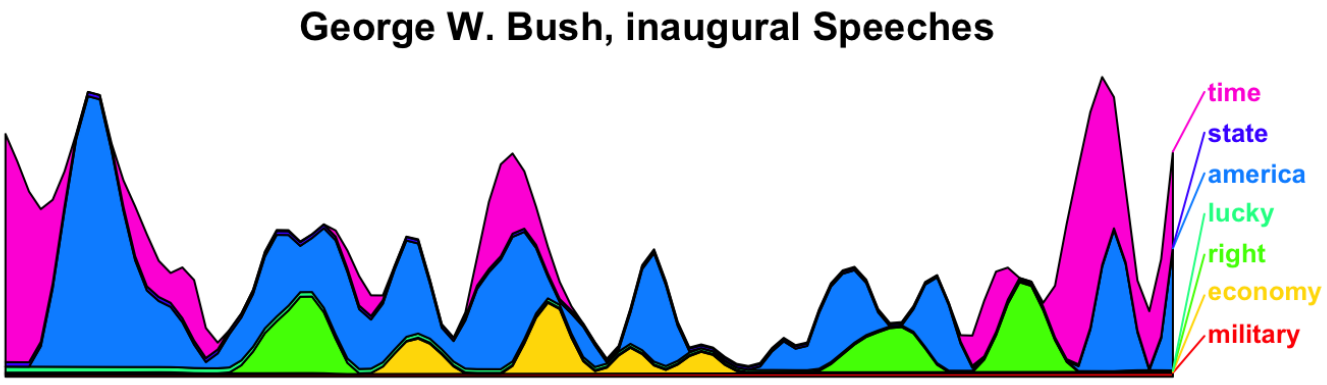
```
par(mfrow=c(2, 1), mar=c(1,1,2,0), bty="n", xaxt="n", yaxt="n")
speech.df=tbl_df(corpus.list.df)%>%filter(File=="GeorgeWBush", type=="inaug", Term==1)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="George W. Bush, inaugural Speec
hes")
```

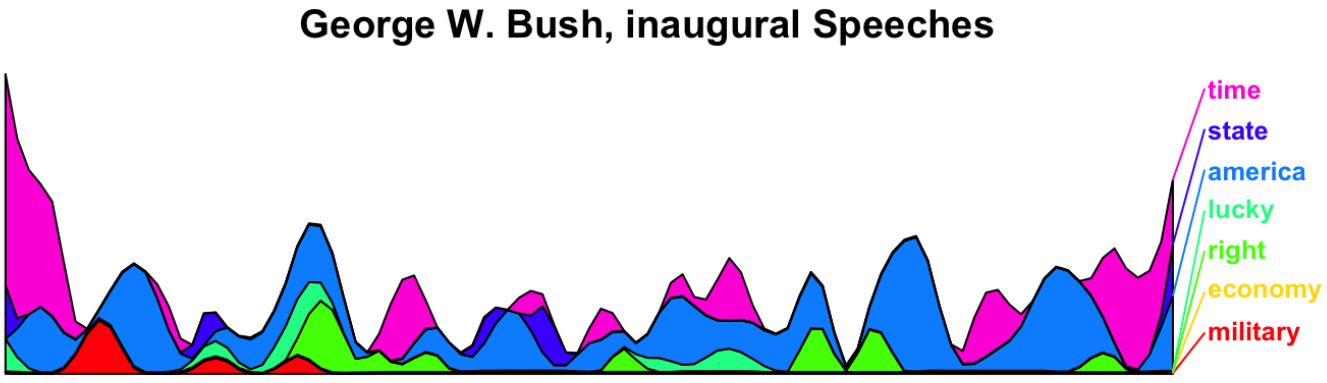
```
[1] 0.04136933 0.08273866 0.12410799 0.16547732 0.20684666 0.24821599 0.28958532
```

Hide

```
speech.df=tbl_df(corpus.list.df)%>%filter(File=="GeorgeWBush", type=="inaug", Term==2)%
>%select(sent.id, govern:nation)
speech.df=as.matrix(speech.df)
speech.df[,-1]=replace(speech.df[,-1], speech.df[,-1]<1/15, 0.001)
speech.df[,-1]=f.smooth.topic(x=speech.df[,1], y=speech.df[,-1])
plot.stacked(speech.df[,1], speech.df[,topic.plot+1],
             xlab="Sentences", ylab="Topic share", main="George W. Bush, inaugural Speec
hes")
```

```
[1] 0.06058238 0.12116475 0.18174713 0.24232951 0.30291189 0.36349426 0.42407664
```





From this graph, we can find GeorgeWBush talked much more about military at the beginning of his topic. As we know, “September 11 attacks” happened in 2001 when GeorgeWBush was being the president of his first term. So if he wanted to be the president for the second term, he must say something about solving terrorist attack. So we can hold the opinion that big events can affect the inaugural speeches of presidents more or less.

Now we make five clusters to classify inaugural speeches of all US presidents.

Hide

```
presid.summary=tbl_df(corpus.list.df)%>%
  filter(type=="inaug")%>%
  select(File, govern:nation)%>%
  group_by(File)%>%
  summarise_each(funs(mean))
```

```
`summarise_each()` is deprecated.
Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
To map `funs` over all variables, use `summarise_all()`
```

Hide

```
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,
              5)
fviz_cluster(km.res,
             stand=T, repel= TRUE,
             data = presid.summary[,-1],
             show.clust.cent=FALSE)
```

From this graph, we find that the five most recent presidents, GeorgeBush, WilliamJClinton, GeorgeWBush, BarackObama, DonaldJTrump are in the left cluster of the graph above. It seems that different era will lead to different topic(in other words, they may face the same problem during the same era.)

part3:word cloud

First, let make a word cloud to show the most common 50 words of five presidents (GeorgeBush, WilliamJClinton, GeorgeWBush, BarackObama, DonaldJTrump) 'inaugural speeches.

Hide

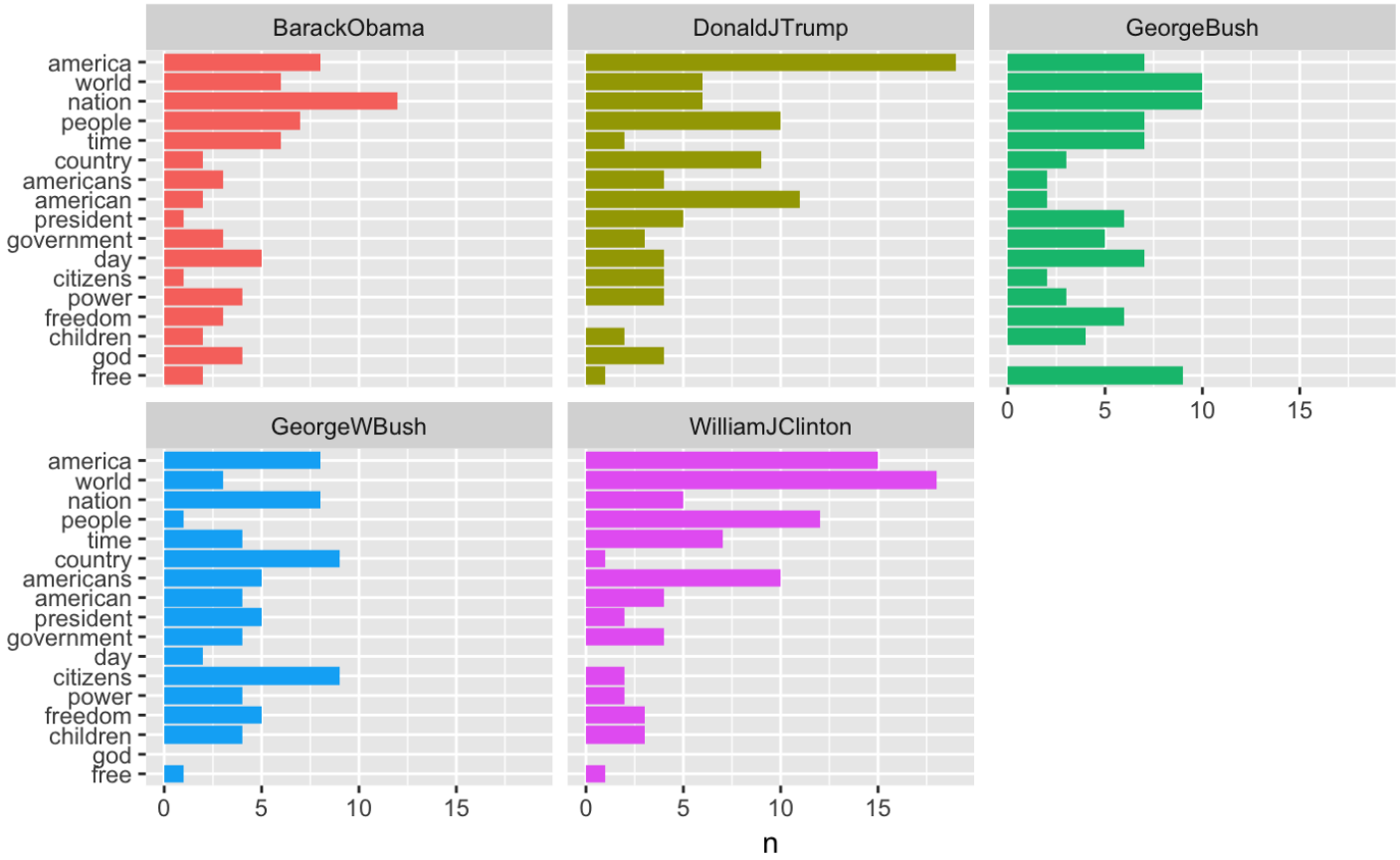
```
td1 <- sentence.list %>% filter(File %in% c("BarackObama", "DonaldJTrump", "GeorgeWBush"
, "WilliamJClinton", "GeorgeBush"), Term == 1) %>% select(File, sentences)
td1$sentences <- as.character(td1$sentences)
td1 <- td1 %>% unnest_tokens(word, sentences)
td1 <- td1 %>%
  anti_join(stop_words, by = "word")
td1 %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 50, color = c("purple4", "red4", "black")))
```

We can find the top 8 most common words are america, world, nation, americans, people, americans, american, time, country, people. It is obvious that the inaugural speeches should concentrate on what they can achieve for “America”.

Now we have a look at which president like to use these common words most.

Hide

```
foo <- td1 %>%
  group_by(word, File) %>%
  count()
bar <- td1 %>%
  group_by(word) %>%
  count() %>%
  rename(all = n)
foo %>%
  left_join(bar, by = "word") %>%
  arrange(desc(all)) %>%
  head(80) %>%
  ungroup() %>%
  ggplot(aes(reorder(word, all, FUN = min), n, fill = File)) +
  #ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ File) +
  theme(legend.position = "none")
```

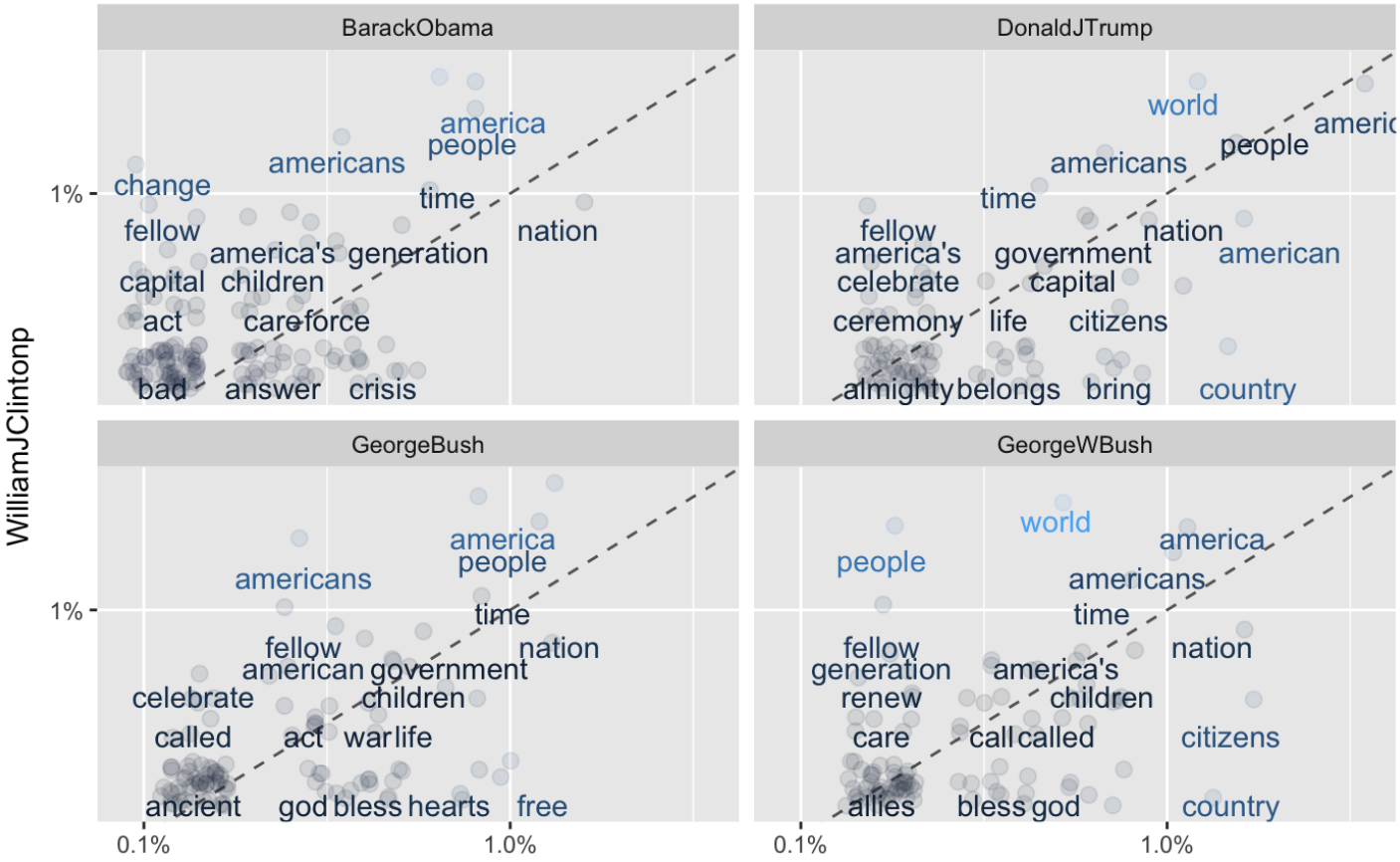


Barack Obama used “nation” most, DonaldJTrump used “american” most, and WilliamJClinton used “world” most.

We can dive deeper into this fundamental question by directly comparing the relative frequency of word use between presidents. In this example, we will compare WilliamJClinton to the other presidents.

Hide

```
frequency1 <- td1 %>%
  count(File, word) %>%
  group_by(File) %>%
  mutate(freq = n / sum(n)) %>%
  select(-n) %>%
  spread(File, freq) %>%
  gather(File, freq, BarackObama : GeorgeWBush) %>%
  filter(!is.na(freq) & !is.na(File))
ggplot(frequency1, aes(freq, WilliamJClinton , color = abs(WilliamJClinton - freq))) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.1, height = 0.1) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  #scale_color_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray95") +
  facet_wrap(~File, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "WilliamJClintonp", x = NULL)
```



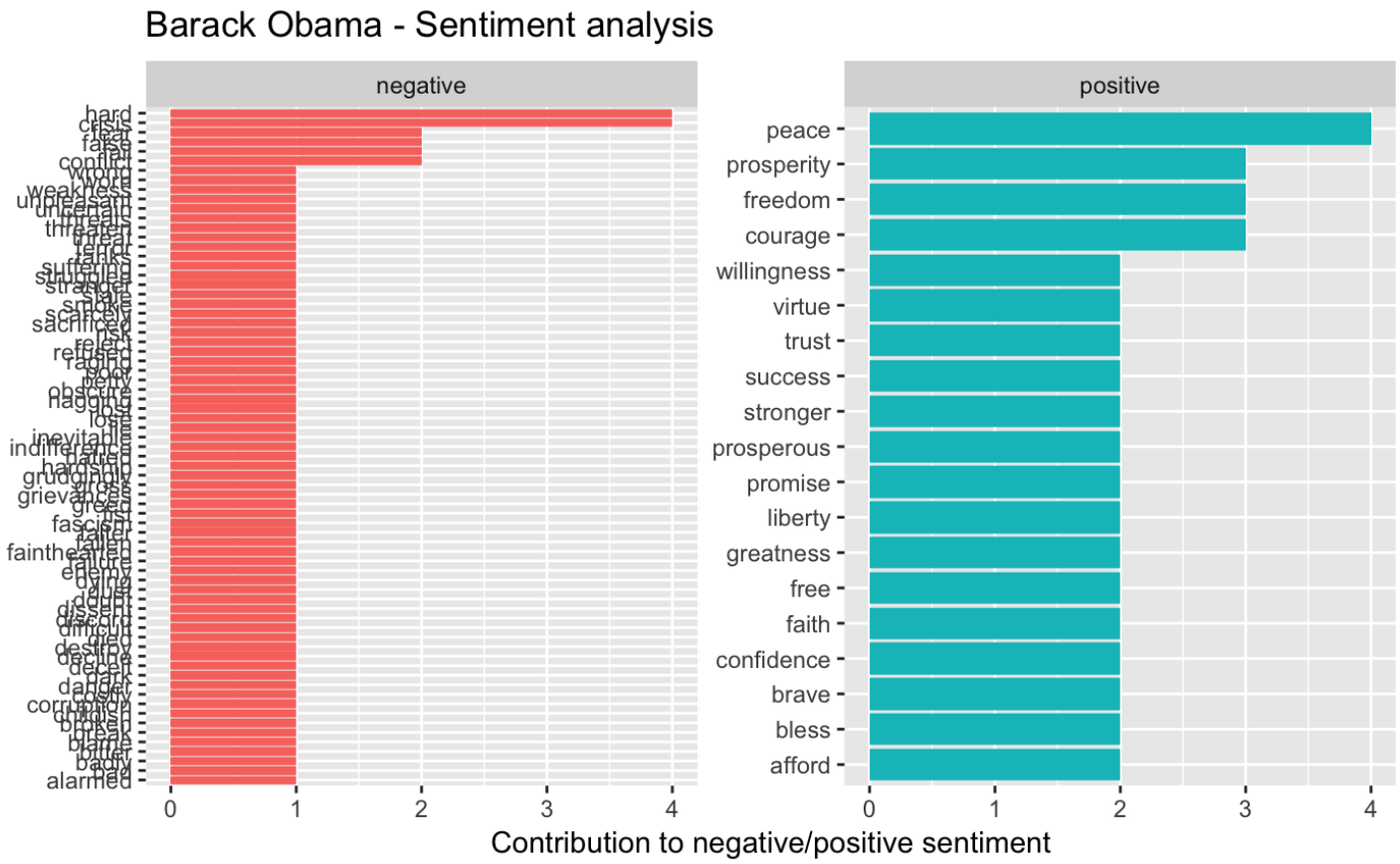
In these plots, words that are close to the dashed line (of equal frequency) have similar frequencies in the corresponding presidents. Words that are further along a particular president axis (such as “change” for William J Clinton vs Barack Obama) are more frequent for that president. The blue-gray scale indicates how different the WilliamJClinton frequency is from the overall frequency (with higher relative frequencies being lighter). The (slightly jittered) points in the background represent the complete set of (high-frequency) words, whereas the displayed words have been chosen to avoid overlap.

The plots give us another useful overview angle. For instance, they suggest that WIlliamJClinton uses “fellow” more than other four presidents, but he uses “americans” as more often than Barack Obama and George Bush but equally often as Donald Trump and George W Bush.

part4:sentiment analysis

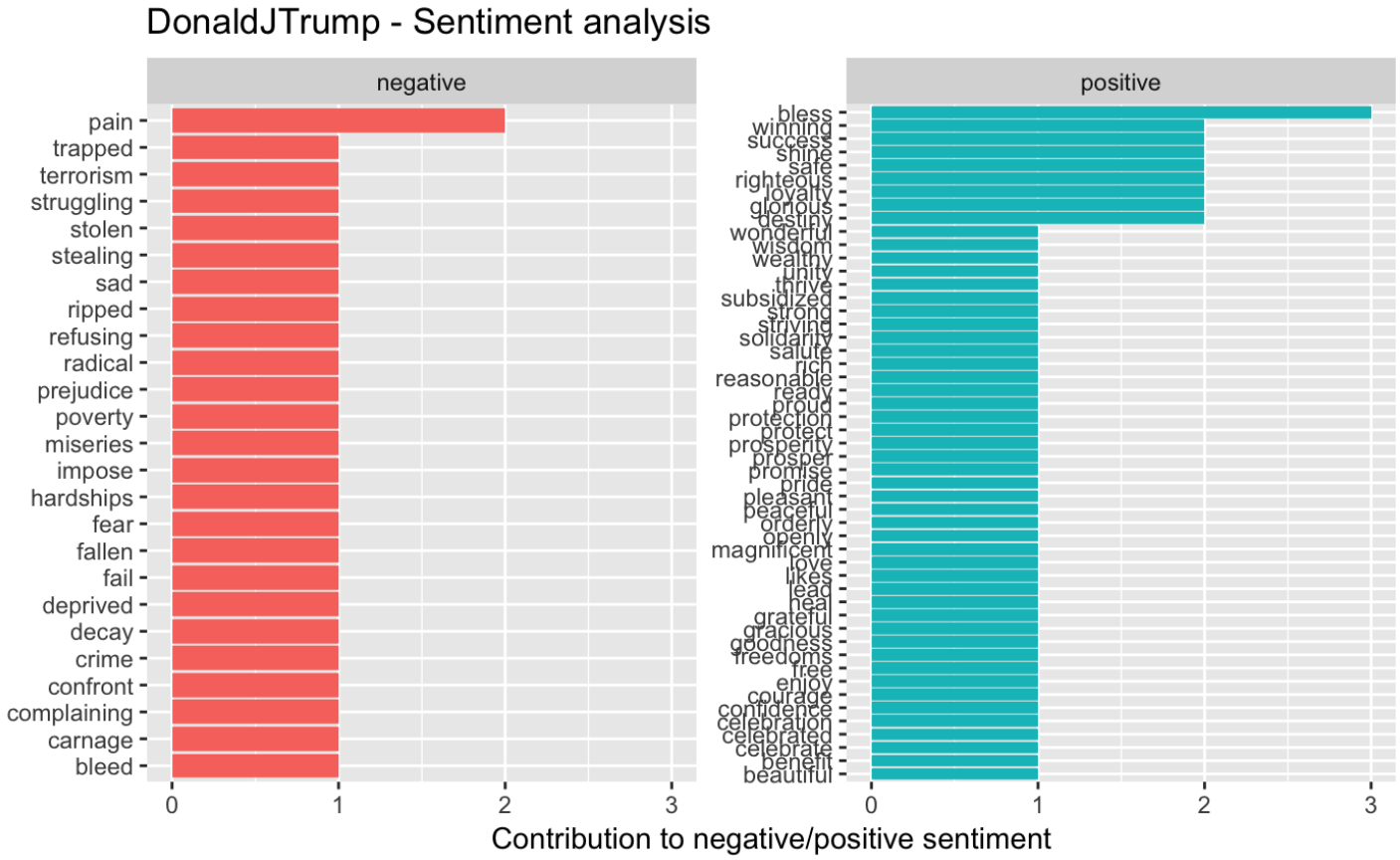
Hide

```
td1 %>%
  filter(File == "BarackObama") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("Barack Obama - Sentiment analysis")
```



Hide

```
td1 %>%
  filter(File == "DonaldJTrump") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("DonaldJTrump - Sentiment analysis")
```



Comparison wordcloud

An alternative way of portraying the most important words for each sentiment is through our favourite word-clouds. This uses the reshape library. Here are all the presidents combined as an example:

Hide

```
td1 %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("#F8766D", "#00BFC4"), max.words = 50)
```

We can see “hard” and “freedom” squaring up against one another; locked in an eternal battle in which “success” counters “gail” and “unity” rages against “break”.