project1

jl4756

9/18/2017

```
library("rvest")
```

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

```
library("qdap")
```

```
## Loading required package: qdapDictionaries
## Loading required package: qdapRegex
## Loading required package: qdapTools
## Loading required package: RColorBrewer
##
## Attaching package: 'qdap'
## The following object is masked from 'package:rvest':
##
##     %>%
## The following object is masked from 'package:base':
##
##     Filter
```

```
library("syuzhet")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:qdap':
##
##     %>%
## The following object is masked from 'package:qdapTools':
##
##     id
## The following object is masked from 'package:qdapRegex':
##
##     explain
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("beeswarm")
library("tibble")
library("sentimentr")
```

```
##
## Attaching package: 'sentimentr'

## The following object is masked from 'package:syuzhet':
##
##     get_sentences
```

```
library("factoextra")
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:qdapRegex':
##
##     %+%

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library("scales")
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:syuzhet':
##
##     rescale
```

```
library("RColorBrewer")
library("RANN")
source("../lib/plotstacked.R")
source("../lib/speechFuncs.R")
```

read in url

```
### Inauguaral speeches
main.page <- read_html(x = "http://www.presidency.ucsb.edu/inaugurals.php")
inaug=f.speechlinks(main.page)
as.Date(inaug[,1], format="%B %e, %Y")
```

```
##  [1] "1789-04-30" "1793-03-04" "1797-03-04" "1801-03-04" "1805-03-04"
##  [6] "1809-03-04" "1813-03-04" "1817-03-04" "1821-03-04" "1825-03-04"
## [11] "1829-03-04" "1833-03-04" "1837-03-04" "1841-03-04" "1845-03-04"
## [16] "1849-03-05" "1853-03-04" "1857-03-04" "1861-03-04" "1865-03-04"
## [21] "1869-03-04" "1873-03-04" "1877-03-05" "1881-03-04" "1885-03-04"
## [26] "1889-03-04" "1893-03-04" "1897-03-04" "1901-03-04" "1905-03-04"
## [31] "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"
## [41] "1949-01-20" "1953-01-20" "1957-01-21" "1961-01-20" "1965-01-20"
## [46] "1969-01-20" "1973-01-20" "1977-01-20" "1981-01-20" "1985-01-21"
## [51] "1989-01-20" "1993-01-20" "1997-01-20" "2001-01-20" "2005-01-20"
## [56] "2009-01-20" "2013-01-21" "2017-01-20" NA
```

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

#### Nomination speeches
```

```
main.page=read_html("http://www.presidency.ucsb.edu/nomination.php")
nomin <- f.speechlinks(main.page)
nomin<-nomin[-47,] # remove the irrelevant line.

#### Farewell speeches
main.page=read_html("http://www.presidency.ucsb.edu/farewell_addresses.php")
farewell <- f.speechlinks(main.page)
```

read in list

```
inaug.list=read.csv("inauglist.csv", stringsAsFactors = FALSE)
nomin.list=read.csv("nominlist.csv", stringsAsFactors = FALSE)
farewell.list=read.csv("farewelllist.csv", stringsAsFactors = FALSE)
```

combine list and url

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

write in full text

```
# 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
}
```

write in Trump's speeches.

```
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(
  X...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)
```

```
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)
                              )
    )
  }
}
```

remove non-sentences

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

choose only "Democratic" and "Republican" party to compare.

```
newlist=sentence.list%>%filter(!is.na(Party))
newlist<-rbind(newlist[newlist$Party=="Democratic",],newlist[newlist$Party=="Republican",])
```

compare only parties

```
party.summary<-aggregate(newlist[,13:22],list(newlist$Party),mean)
party.summary=as.data.frame(party.summary)
party.summary$ratio=(party.summary$negative)/(party.summary$positive)
# negative:positive rate, the smaller the better
party.summary$ratio
```

```
## [1] 0.4650764 0.5031626
```

```
# the ratio of "Democratic" is smaller than the one of "Republican", which means, the speech of Democra
rownames(party.summary)<-party.summary[,1]
party.summary<-party.summary[,-1]
# create a plot to compare different emotions
{x<-c(1:8)
plot(x,party.summary[1,1:8],type="l",col=1,xaxt = "n")
lines(x,party.summary[2,1:8],type="l",col=2)
legend("topright",c("Democratic","Republican"),lty=1,col=c("black","red"))
axis(1,at=1:8,labels=c("anger","anticipation","disgust","fear","joy","sadness","surprise","trust"))}
```

compare through president, order by party

```
president.summary<-aggregate(newlist[,13:22],list(newlist$File,newlist$Party),mean)
president.summary=as.data.frame(president.summary)
president.summary$ratio=(president.summary$negative)/(president.summary$positive)
rownames(president.summary)<-president.summary[,1]
president.summary<-president.summary[,-1]
colnames(president.summary)[1]<-c("Party")
# compare ratio in numbers of presidents from different parties.
compare.ratio<-president.summary[order(president.summary$ratio),]
table(compare.ratio$Party[compare.ratio$ratio<0.5])
```

```
##
## Democratic Republican
##         14         13
```

```
# compare positive word rates in numbers of presidents from different parties.
compare.pos<-president.summary[order(president.summary$positive),]
table(compare.pos$Party[compare.pos$positive>0.075])
```

```
##
## Democratic Republican
##         12         14
```

```
# there isn't too much difference between the number of presidents who likes to use positive words
```

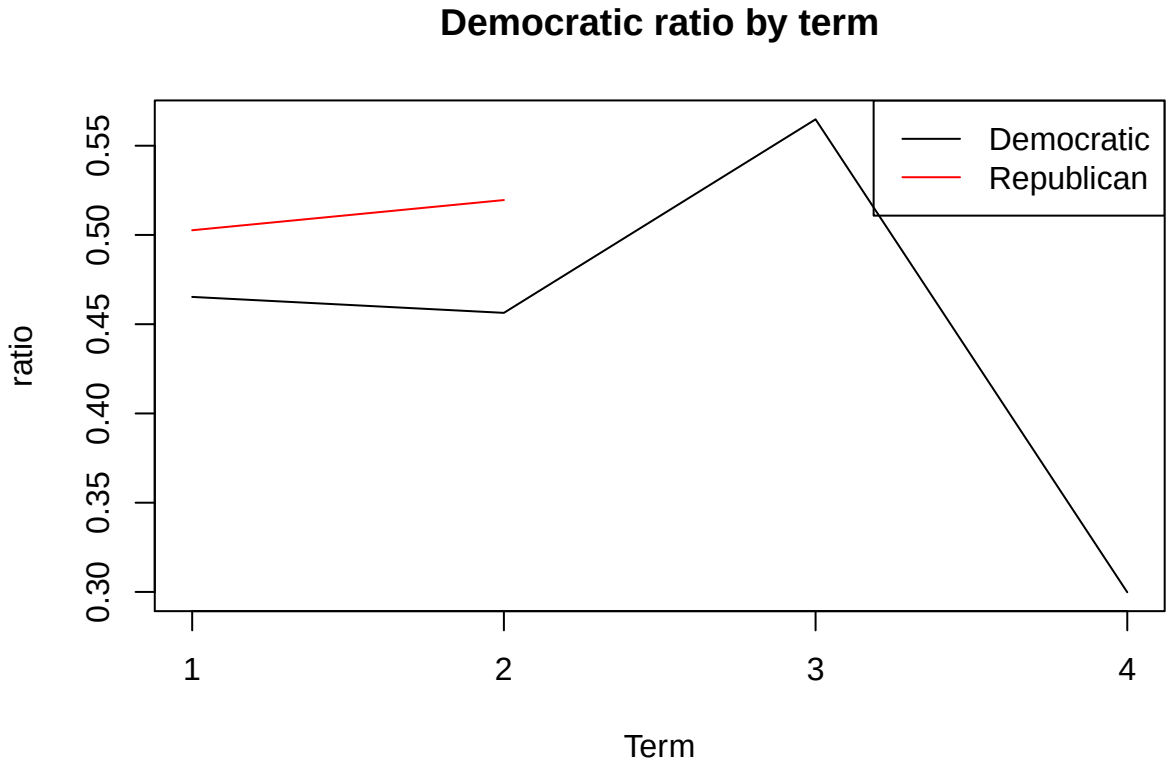
compare between Terms

```
term.summary<-aggregate(newlist[,13:22],list(newlist$Term),mean)
term.summary=as.data.frame(term.summary)
term.summary$ratio=(term.summary$negative)/(term.summary$positive)
rownames(term.summary)<-term.summary[,1]
term.summary<-term.summary[,-1]
term.summary$ratio # as we can see, presidents tend to use more possitive words when they become a pres
```

```
## [1] 0.5365998 0.4869489 0.4946730 0.5647641 0.2998310
```

compare between Terms an Parties

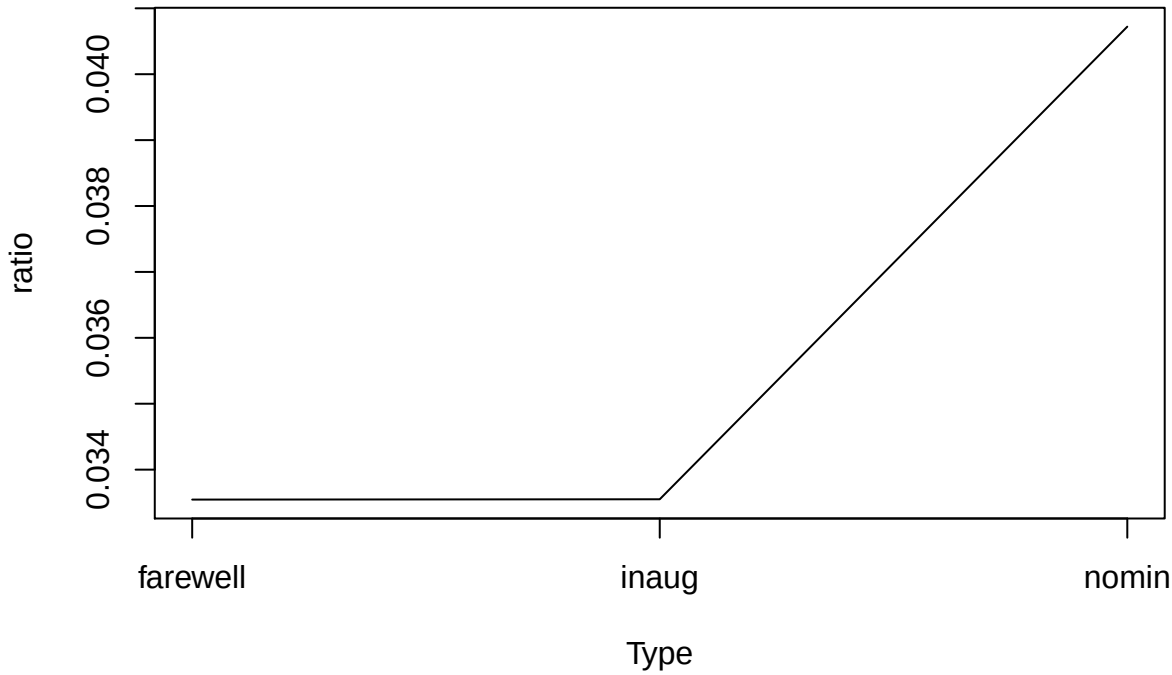
```
tp.summary<-aggregate(newlist[,13:22],list(newlist$Term,newlist$Party),mean)
tp.summary=as.data.frame(tp.summary)
tp.summary$ratio=(tp.summary$negative)/(tp.summary$positive)
colnames(tp.summary)[1:2]<-c("Term","Party")

{x<-c(1:4)
plot(x,tp.summary[1:4,13],type="l",col=1,xlab="Term",ylab="ratio",main="Democratic ratio by term",xaxt =
lines(tp.summary[6:7,13],type="l",col=2)
axis(1,at=1:4,labels=c(1,2,3,4))
legend("topright",c("Democratic","Republican"),lty=1,col=c("black","red"))
}
```



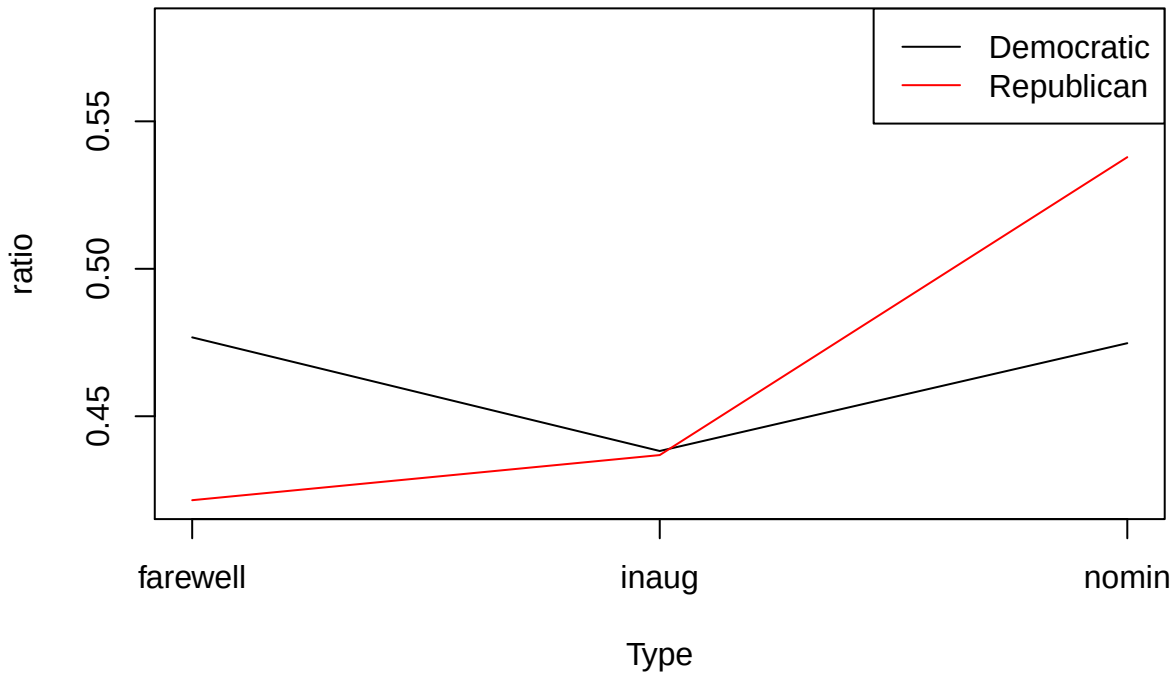

```
# we can see Democratic presidents tend to use more positive words from their Term1 to Term2, except fo
# However we can see that Republican President tends to use more negative words in their 2nd Term.
```

compare between type of files.

```
type.summary<-aggregate(newlist[,13:22],list(newlist$type),mean)
type.summary=as.data.frame(type.summary)
type.summary$ratio=(type.summary$negative)/(type.summary$positive)
rownames(type.summary)<-type.summary[,1]
type.summary<-type.summary[,-1]
{x<-c(1:3)
plot(x,tp.summary[1:3,11],type="l",col=1,xlab="Type",ylab="ratio",xaxt = "n")
axis(1,at=1:3,labels=rownames(type.summary)[1:3])
}
```




```
# we can see that there is a big decrease of negative words from nomin to inaug, I suppose this may bec
typ.summary<-aggregate(newlist[,13:22],list(newlist$type,newlist$Party),mean)
typ.summary=as.data.frame(typ.summary)
typ.summary$ratio=(typ.summary$negative)/(typ.summary$positive)
colnames(typ.summary)[1:2]<-c("Type","Party")
{x<-c(1:3)
plot(x,typ.summary[1:3,13],type="l",col=1,xlab="Type",ylab="ratio",xaxt = "n",ylim=c(min(typ.summary[,13
lines(typ.summary[4:6,13],type="l",col=2)
axis(1,at=1:3,labels=c("farewell","inaug","nomin"))
legend("topright",c("Democratic","Republican"),lty=1,col=c("black","red"))
}
```



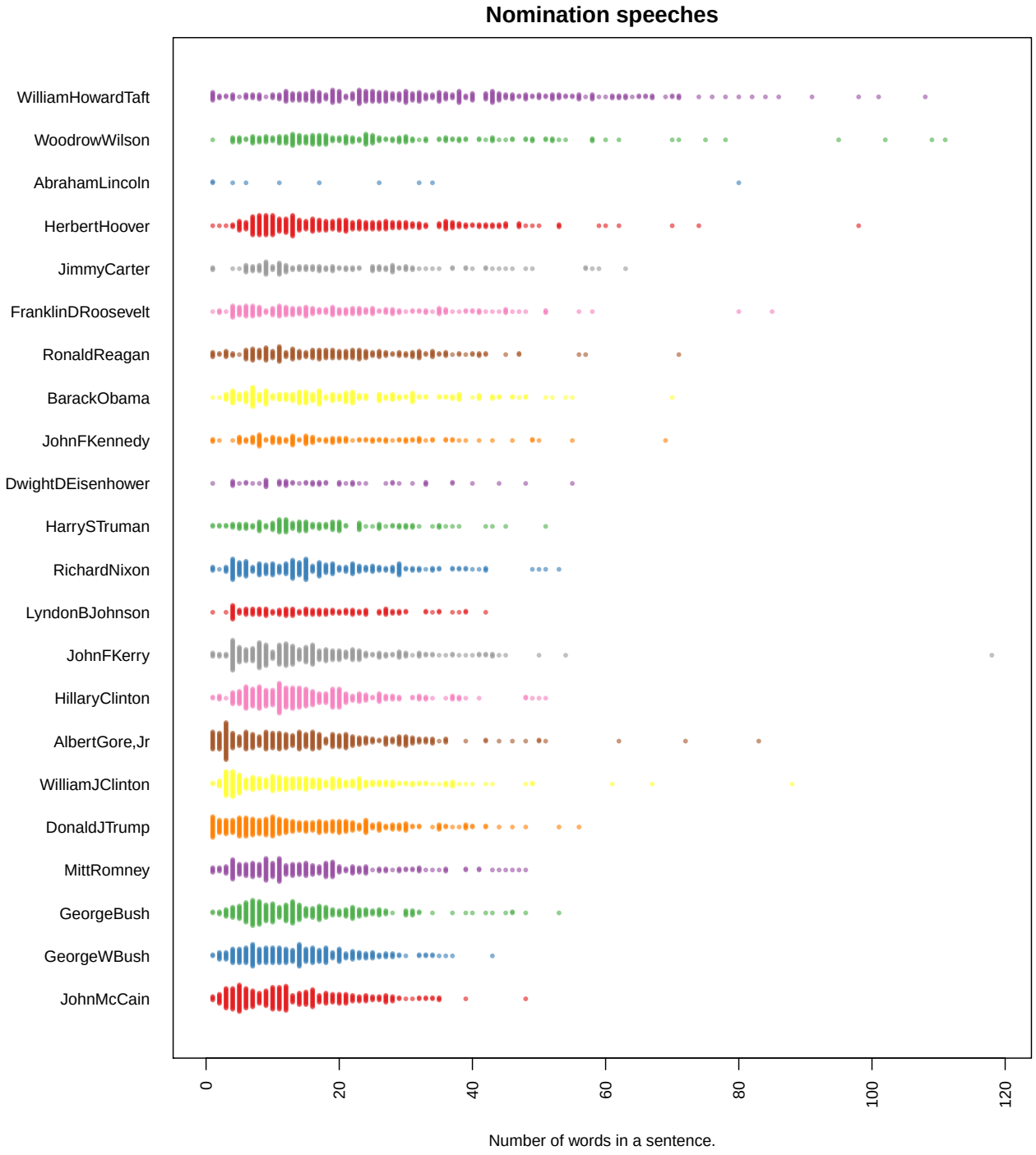
#

choose those that are famous presidents

```
sel.comparison=c("DonaldJTrump","JohnMcCain", "GeorgeBush", "MittRomney", "GeorgeWBush",
                 "RonaldReagan","AlbertGore,Jr", "HillaryClinton","JohnFKerry",
                 "WilliamJClinton","HarrySTruman", "BarackObama", "LyndonBJohnson",
                 "GeraldRFord", "JimmyCarter", "DwightDEisenhower", "FranklinDRoosevelt",
                 "HerbertHoover","JohnFKennedy","RichardNixon","WoodrowWilson",
                 "AbrahamLincoln", "TheodoreRoosevelt", "JamesGarfield",
                 "JohnQuincyAdams", "UlyssesSGrant", "ThomasJefferson",
                 "GeorgeWashington", "WilliamHowardTaft", "AndrewJackson",
                 "WilliamHenryHarrison", "JohnAdams")
```

First term

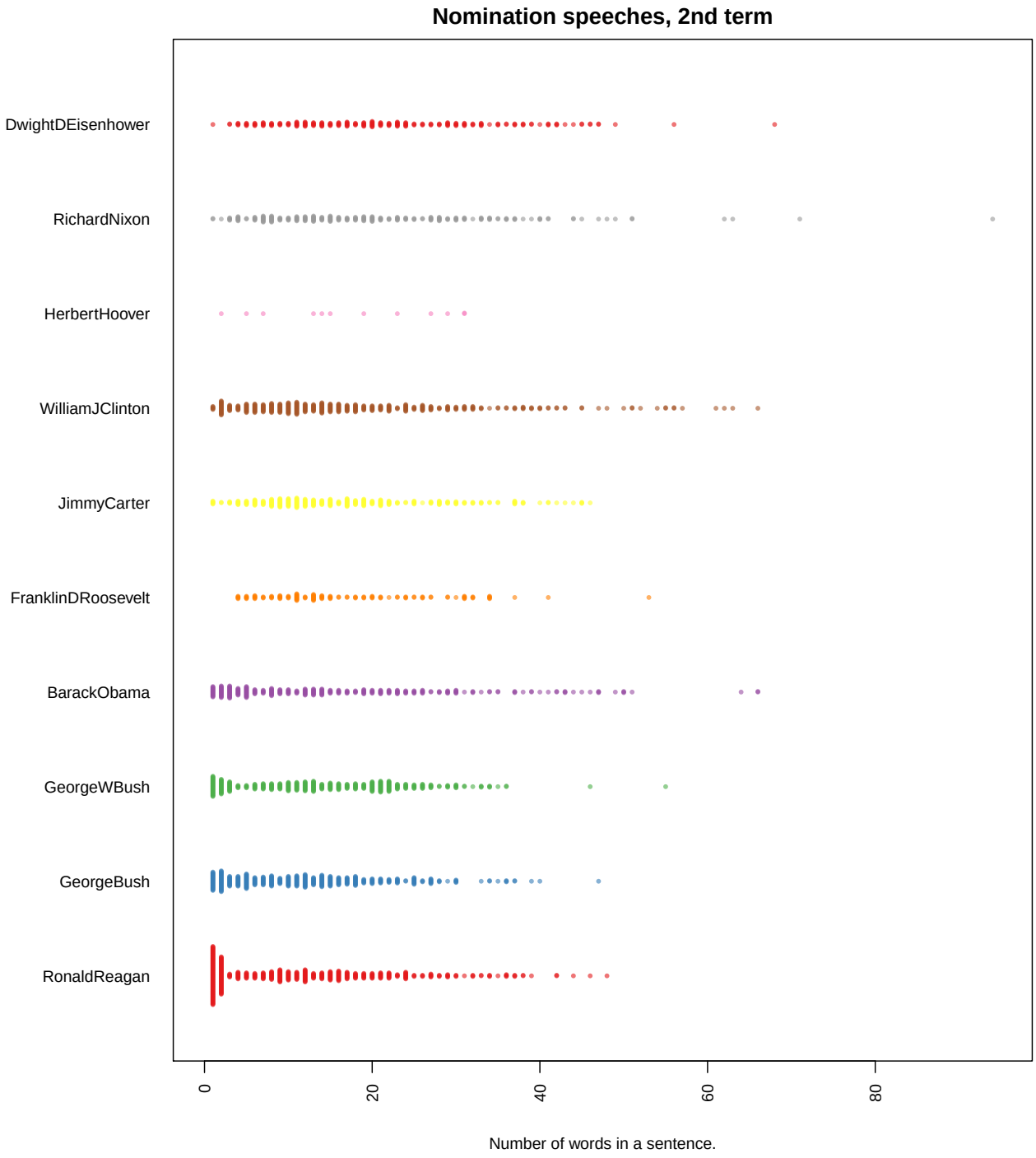
```
par(mar=c(4, 11, 2, 2))
#sel.comparison=levels(sentence.list$FileOrdered)
sentence.list.sel=filter(sentence.list,
                         type=="nomin", Term==1, File%in%sel.comparison)
sentence.list.sel$File=factor(sentence.list.sel$File)

sentence.list.sel$FileOrdered=reorder(sentence.list.sel$File,
                                      sentence.list.sel$word.count,
                                      mean,
                                      order=T)
beeswarm(word.count~FileOrdered,
         data=sentence.list.sel,
         horizontal = TRUE,
         pch=16, col=alpha(brewer.pal(9, "Set1"), 0.6),
         cex=0.55, cex.axis=0.8, cex.lab=0.8,
         spacing=5/nlevels(sentence.list.sel$FileOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="Nomination speeches")
```



second term

```
par(mar=c(4, 11, 2, 2))

#sel.comparison=levels(sentence.list$FileOrdered)
sentence.list.sel=filter(sentence.list,
                         type=="nomin", Term==2, File%in%sel.comparison)
```

```
sentence.list.sel$File=factor(sentence.list.sel$File)

sentence.list.sel$FileOrdered=reorder(sentence.list.sel$File,
                                      sentence.list.sel$word.count,
                                      mean,
                                      order=T)

beeswarm(word.count~FileOrdered,
         data=sentence.list.sel,
         horizontal = TRUE,
         pch=16, col=alpha(brewer.pal(9, "Set1"), 0.6),
         cex=0.55, cex.axis=0.8, cex.lab=0.8,
         spacing=1.2/nlevels(sentence.list.sel$FileOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="Nomination speeches, 2nd term")
```



find the longest length of word in each sentence

```
word<-matrix(0,nrow=21326,ncol=max(sentence.list$word.count)+1)
for (i in 1:21326){
  word[i,1:length(nchar(strsplit(as.character(sentence.list$sentences[i]),split="\\, |\\,| |\\:|\\-|\\>
}
word[,1:124]<-as.numeric(word[,1:124])
```

```r
rownames(word)<-sentence.list$File

# find the longest word in each sentence.
maxlength.word<-matrix(NA,nrow=nrow(word),ncol=2)
colnames(maxlength.word)<-c("President","max length of word")
maxlength.word[,1]<-sentence.list$File
president<-as.matrix(as.data.frame(table(sentence.list$File))[,1])
maxlength.word[,2]<-apply(word[,1:124],1,max)
table(as.numeric(maxlength.word[,2]))
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##  169  105   91  365  711  924 1915 2461 3182 3917 2818 2134 1213  990  181
##   16   17   18   19
##  117   26    6    1
```

```r
# find the number of complex words in each president's speeches.
complexword<-matrix(0,ncol=2,nrow=length(president))
complexword[,1]<-as.matrix(president)
for(i in 1:length(president)){
  complexword[i,2]<-sum(as.numeric(word[rownames(word)==president[i],1:124])>=9)
}
complexword.ordered<-complexword[order(as.numeric(complexword[,2])),]
head(complexword.ordered)
```

```
##      [,1]                [,2]
## [1,] "TheodoreRoosevelt" "94"
## [2,] "WalterFMondale"    "127"
## [3,] "CharlesEHughes"    "133"
## [4,] "ZacharyTaylor"     "165"
## [5,] "GeorgeMcGovern"    "182"
## [6,] "MichaelDukakis"    "215"
```

```r
tail(complexword.ordered)
```

```
##       [,1]                 [,2]
## [53,] "RichardNixon"       "1529"
## [54,] "HerbertHoover"      "1555"
## [55,] "DonaldJTrump"       "1565"
## [56,] "FranklinDRoosevelt" "1656"
## [57,] "BenjaminHarrison"   "1876"
## [58,] "WilliamHowardTaft"  "2408"
```

```r
# Surprisingly, President Trump is the highest three presidents of using complex words.
```