Project 01

Xiaochen Fan UNI: xf2170

Jan 30, 2018

Step 0: check and install needed packages. Load the libraries and functions.

```
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")
# You may need to run
# sudo ln -f -s $(/usr/libexec/java_home)/jre/lib/server/libjvm.dylib /usr/local/lib
# in order to load qdap
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")
library("readxl")

source("../lib/plotstacked.R")
source("../lib/speechFuncs.R")
```

Step 1: Data Importing

```
speech <- read_excel("E:/GitHub/Spring2018-Project1-XiaochenFan01/data/InaugurationInfo.xlsx")
#remove line 41 that is missing
speech <- speech[-41,]
#remove line 57 because we do not analyse it now
speech <- speech[-57,]

speech$fulltext <- NA
for (i in seq(nrow(speech))){
  text <- readLines(paste0("E:/Github/Spring2018-Project1-XiaochenFan01/data/InauguralSpeeches/inaug", s
  speech$fulltext[i] <- text
}
```

Step 2: Data Categorizing

I will categorize data according to term.

```
speech$Words <- as.numeric(speech$Words)
unique(speech$Term)
```

```
## [1] 1 2 3 4
```

```
term1 <- speech[speech$Term == "1",]
term2 <- speech[speech$Term == "2",]
speech <- rbind(term1, term2)
```

There are four categories of term as 1, 2, 3 and 4. Only Franklin D. Roosevelt had term 3 and 4, therefore, we only analyse term 1 and 2.

```
nrow(term1)
```

```
## [1] 37
```

```
unique(term1$President)
```

```
##  [1] "George Washington"      "John Adams"
##  [3] "Thomas Jefferson"       "James Madison"
##  [5] "James Monroe"           "John Quincy Adams"
##  [7] "Andrew Jackson"         "Martin van Buren"
##  [9] "William Henry Harrison" "James K. Polk"
## [11] "Zachary Taylor"         "Franklin Pierce"
## [13] "James Buchanan"         "Abraham Lincoln"
## [15] "Ulysses S. Grant"       "Rutherford B. Hayes"
## [17] "James Garfield"         "Grover Cleveland - I"
## [19] "Benjamin Harrison"      "William McKinley"
## [21] "Theodore Roosevelt"     "William Howard Taft"
## [23] "Woodrow Wilson"         "Warren G. Harding"
## [25] "Calvin Coolidge"        "Herbert Hoover"
## [27] "Franklin D. Roosevelt"  "Dwight D. Eisenhower"
## [29] "John F. Kennedy"        "Lyndon B. Johnson"
## [31] "Richard Nixon"          "Jimmy Carter"
## [33] "Ronald Reagan"          "George Bush"
## [35] "William J. Clinton"     "George W. Bush"
```

```
## [37] "Barack Obama"
```

```
nrow(term2)
```

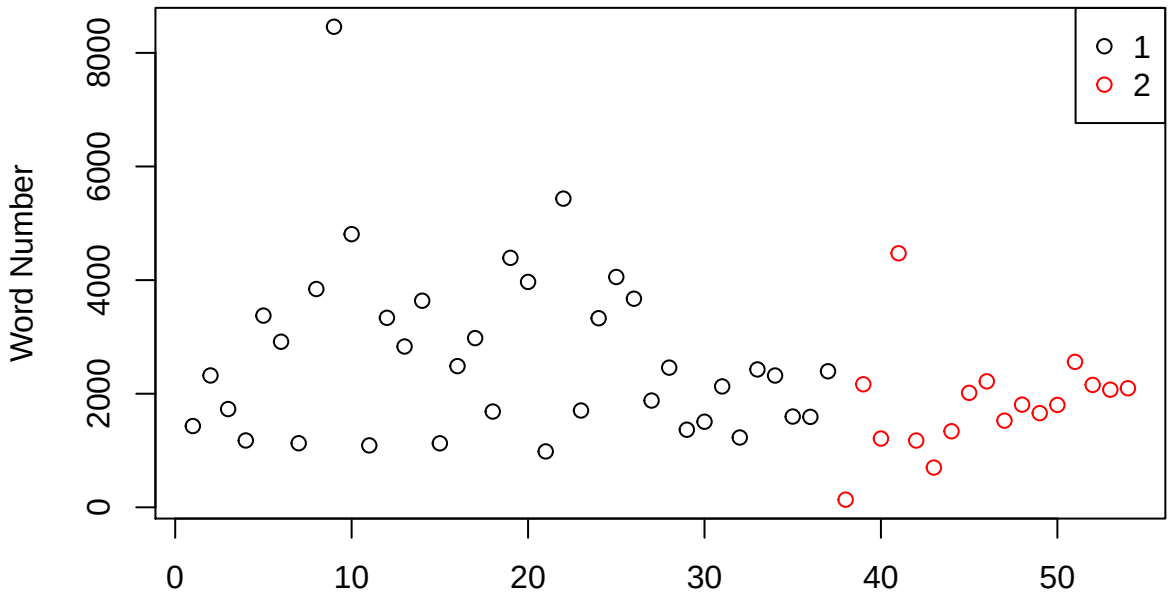
```
## [1] 17
```

```
unique(term2$President)
```

```
##  [1] "George Washington"     "Thomas Jefferson"
##  [3] "James Madison"         "James Monroe"
##  [5] "Andrew Jackson"        "Abraham Lincoln"
##  [7] "Ulysses S. Grant"      "Grover Cleveland - II"
##  [9] "William McKinley"      "Woodrow Wilson"
## [11] "Franklin D. Roosevelt" "Dwight D. Eisenhower"
## [13] "Richard Nixon"         "Ronald Reagan"
## [15] "William J. Clinton"    "George W. Bush"
## [17] "Barack Obama"
```

There are totally 37 presidents and 17 of them had their second term and gave two speeches.

```
plot(speech$Words, col = factor(speech$Term), xlab = "", ylab = "Word Number")
legend("topright", legend = levels(factor(speech$Term)), col = 1:length(levels(factor(speech$Term))), p
```




```
mean(term1$Words)
```

```
## [1] 2670.27
```

```
mean(term2$Words)
```

```
## [1] 1830
```

The plot and the calculation for the mean of word number shows that for term two, presidents tended to give a shorter inaugural speech.

Step 3: Data Processing – Generate List of Sentences

We will use sentences as units of analysis for this project, as sentences are natural languge units for organizing thoughts and ideas. For each extracted sentence, we apply sentiment analysis using NRC sentiment lexion. "The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing."

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`).

```
sentence.list.term1=NULL
for(i in 1:nrow(term1)){
  sentences=sent_detect(term1$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.term1=rbind(sentence.list.term1,
                        cbind(term1[i, ],
                              sentences=as.character(sentences),
                              word.count,
                              emotions,
                              sent.id=1:length(sentences)
                              )
    )
  }
}

sel.comparison=c("George Washington", "Thomas Jefferson", "James Madison", "James Monroe", "Andrew Jacks
sentence.list.term01=filter(sentence.list.term1,
                        President%in%sel.comparison)
```

We choose presidents who made both term one and two speeches to do the data analysis.

```
sentence.list.term2=NULL
for(i in 1:nrow(term2)){
  sentences=sent_detect(term2$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.term2=rbind(sentence.list.term2,
                        cbind(term2[i, ],
                              sentences=as.character(sentences),
```

```
                                word.count,
                                emotions,
                                sent.id=1:length(sentences)
                                )
    )
  }
}

sentence.list.term02=filter(sentence.list.term2,
                            President%in%sel.comparison)
```

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

```
sentence.list.term01=
  sentence.list.term01%>%
  filter(!is.na(word.count))
sentence.list.term02=
  sentence.list.term02%>%
  filter(!is.na(word.count))
```

Step 4: Data Analysis

Length of Term One Speeches

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

sentence.list.term01$President=factor(sentence.list.term01$President)

sentence.list.term01$PresidentOrdered=reorder(sentence.list.term01$President,
                                    sentence.list.term01$word.count,
                                    mean,
                                    order=T)

beeswarm(word.count~PresidentOrdered,
         data=sentence.list.term01,
         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.term01$PresidentOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="Length of Term One Speeches")
```

Length of Term Two Speeches

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

sentence.list.term02$President=factor(sentence.list.term02$President)
```

```r
sentence.list.term02$PresidentOrdered=reorder(sentence.list.term02$President,
                                    sentence.list.term02$word.count,
                                    mean,
                                    order=T)

beeswarm(word.count~PresidentOrdered,
         data=sentence.list.term02,
         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.term02$PresidentOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="Length of Term Two Speeches")
```

Sentiment Analysis

Sentence length variation over the course of the speech, with emotions.

First, define a function that shows sentence length variation.

```
f.plotsent.len=function(In.list, InFile, InTerm, President){

  col.use=c("lightgray", "red2", "darkgoldenrod1",
            "chartreuse3", "blueviolet",
            "darkgoldenrod2", "dodgerblue3",
            "darkgoldenrod1", "darkgoldenrod1",
            "black", "darkgoldenrod2")

  In.list$topemotion=apply(select(In.list,
                                  anger:positive),
                           1, which.max)
  In.list$topemotion.v=apply(select(In.list,
                                    anger:positive),
                             1, max)
  In.list$topemotion[In.list$topemotion.v<0.05]=0
  In.list$topemotion=In.list$topemotion+1

  temp=In.list$topemotion.v
  In.list$topemotion.v[temp<0.05]=1

  df=In.list%>%filter(File==InFile, Term==InTerm)%>%
    select(sent.id, word.count,
           topemotion, topemotion.v)

  ptcol.use=alpha(col.use[df$topemotion], sqrt(sqrt(df$topemotion.v)))

  plot(df$sent.id, df$word.count,
       col=ptcol.use,
       type="h", #ylim=c(-10, max(In.list$word.count)),
       main=President)
}
```

We choose four presidents and demonstrate sentence length variation of their term one speeches.

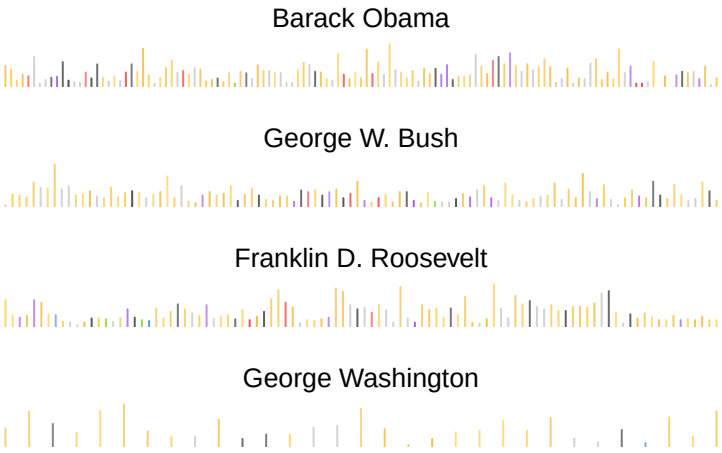
```
par(mfrow=c(4,1), mar=c(1,0,2,0), bty="n", xaxt="n", yaxt="n", font.main=1)

f.plotsent.len(In.list=sentence.list.term01, InFile="BarackObama", InTerm=1, President="Barack Obama")

f.plotsent.len(In.list=sentence.list.term01, InFile="GeorgeWBush", InTerm=1, President="George W. Bush"

f.plotsent.len(In.list=sentence.list.term01, InFile="FranklinDRoosevelt", InTerm=1, President="Franklin

f.plotsent.len(In.list=sentence.list.term01, InFile="GeorgeWashington", InTerm=1, President="George Was
```



Then, do the same to their term two speeches.

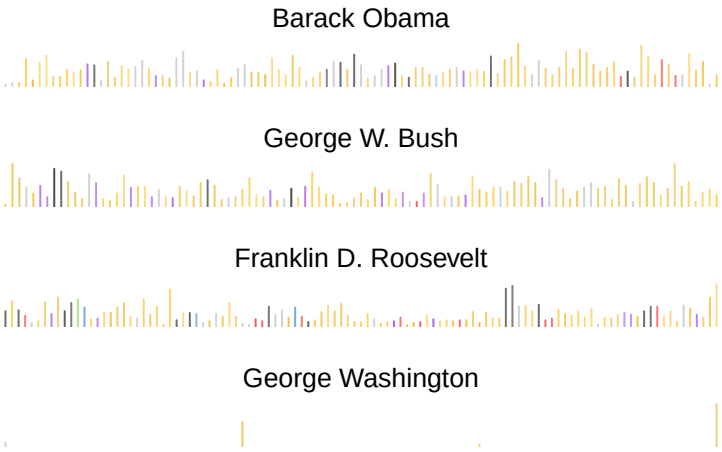
```
par(mfrow=c(4,1), mar=c(1,0,2,0), bty="n", xaxt="n", yaxt="n", font.main=1)

f.plotsent.len(In.list=sentence.list.term02, InFile="BarackObama", InTerm=2, President="Barack Obama")

f.plotsent.len(In.list=sentence.list.term02, InFile="GeorgeWBush", InTerm=2, President="George W. Bush")

f.plotsent.len(In.list=sentence.list.term02, InFile="FranklinDRoosevelt", InTerm=2, President="Franklin

f.plotsent.len(In.list=sentence.list.term02, InFile="GeorgeWashington", InTerm=2, President="George Was
```

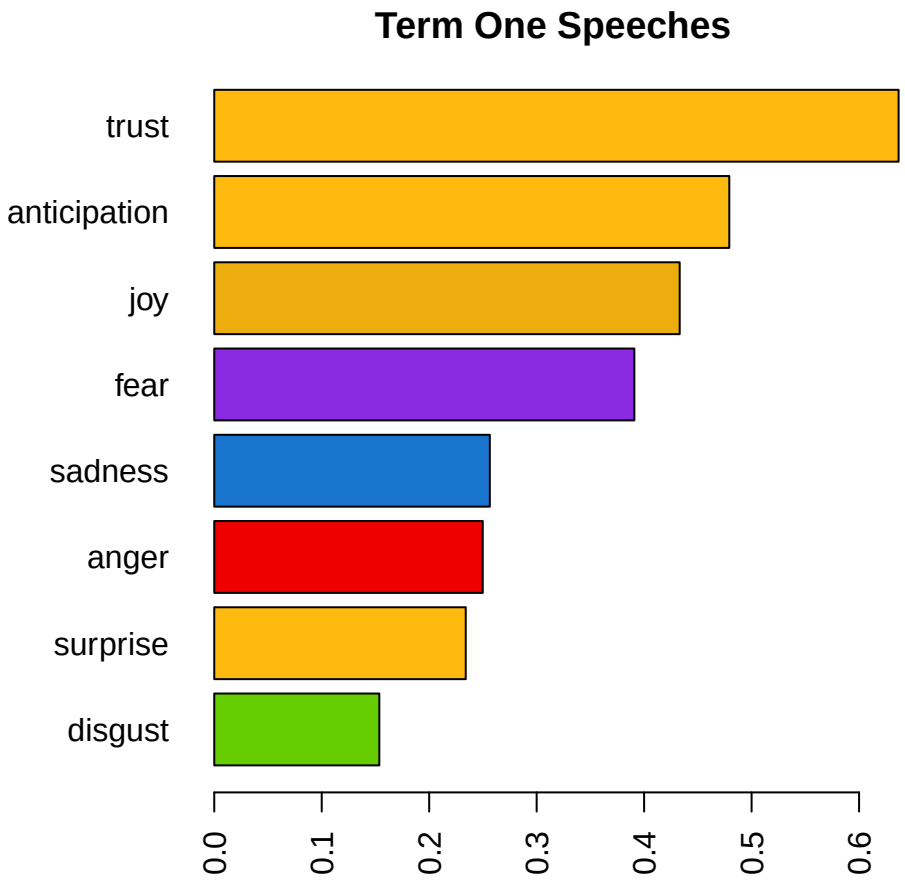


Clustering of emotions

Term One Speech

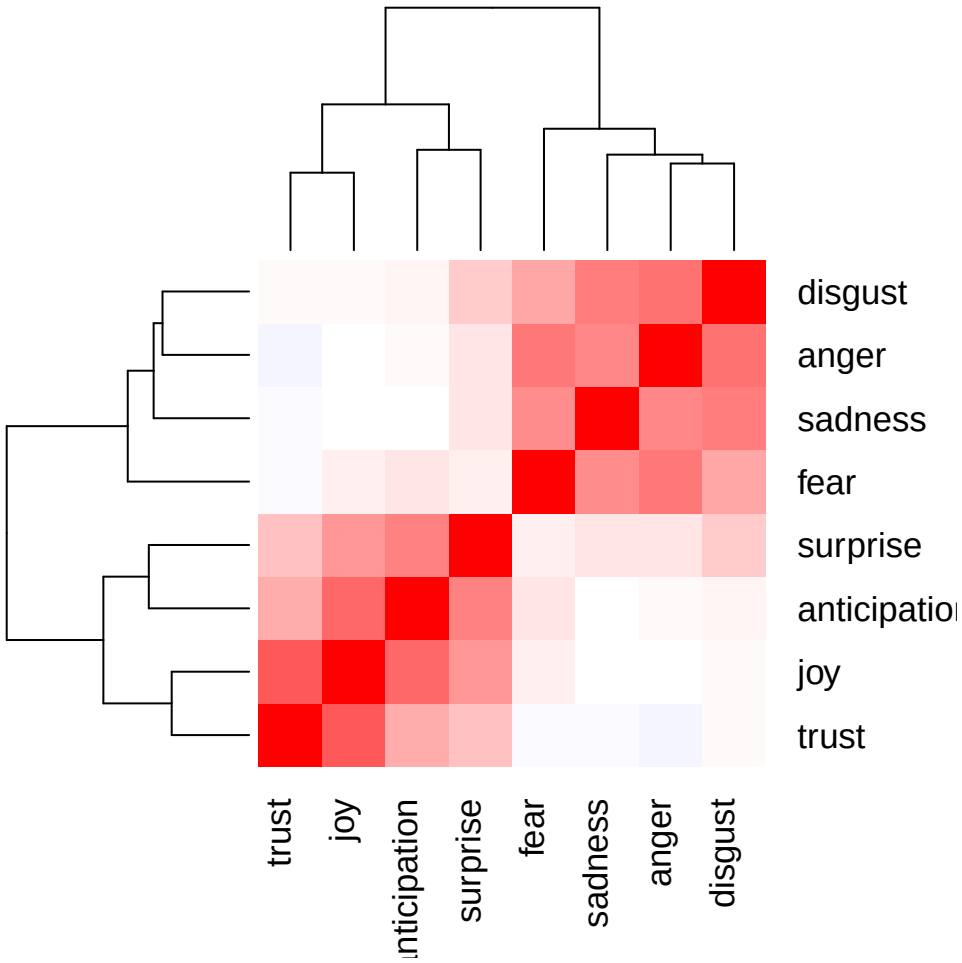
```
heatmap.2(cor(sentence.list.term01%>%select(anger:trust)),
          scale = "none",
          col = bluered(100), , margin=c(6, 6), key=F,
          trace = "none", density.info = "none")
```


```
par(mar=c(4, 6, 2, 1))
emo.means=colMeans(select(sentence.list.term01, anger:trust)>0.01)
col.use=c("red2", "darkgoldenrod1",
            "chartreuse3", "blueviolet",
            "darkgoldenrod2", "dodgerblue3",
            "darkgoldenrod1", "darkgoldenrod1")
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, main="Term One Spee
```

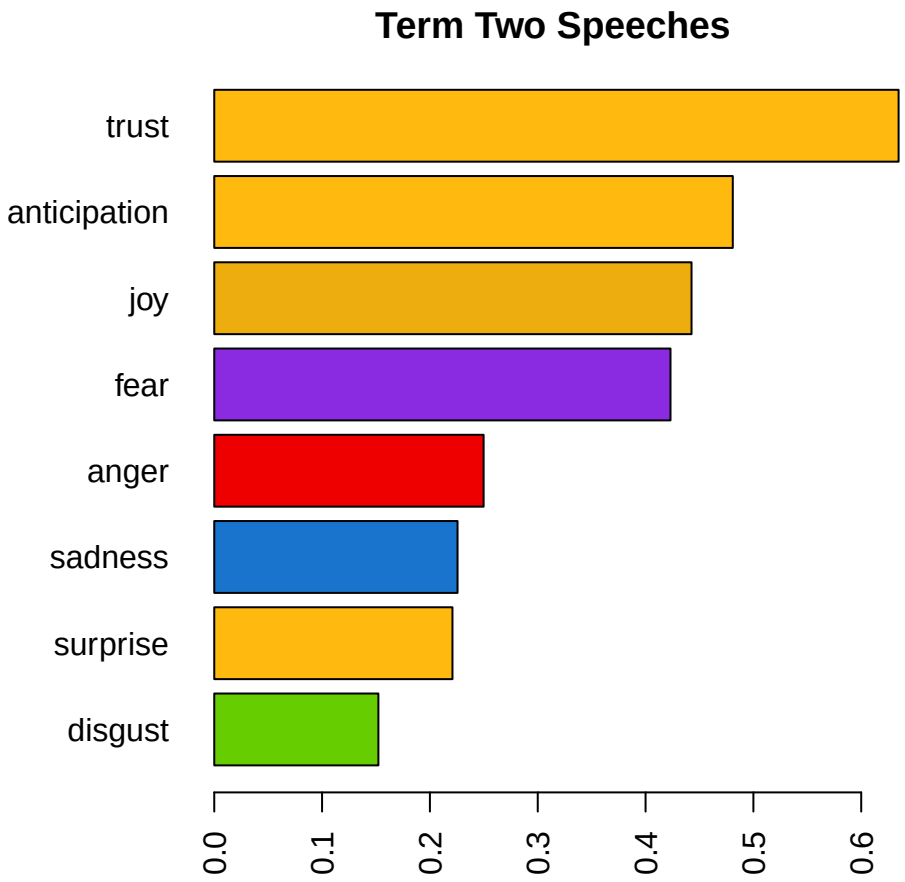


Term Two Speech

```
heatmap.2(cor(sentence.list.term02%>%select(anger:trust)),
          scale = "none",
          col = bluered(100), , margin=c(6, 6), key=F,
          trace = "none", density.info = "none")
```




```
par(mar=c(4, 6, 2, 1))
emo.means=colMeans(select(sentence.list.term02, anger:trust)>0.01)
col.use=c("red2", "darkgoldenrod1",
            "chartreuse3", "blueviolet",
            "darkgoldenrod2", "dodgerblue3",
            "darkgoldenrod1", "darkgoldenrod1")
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, main="Term Two Spee
```



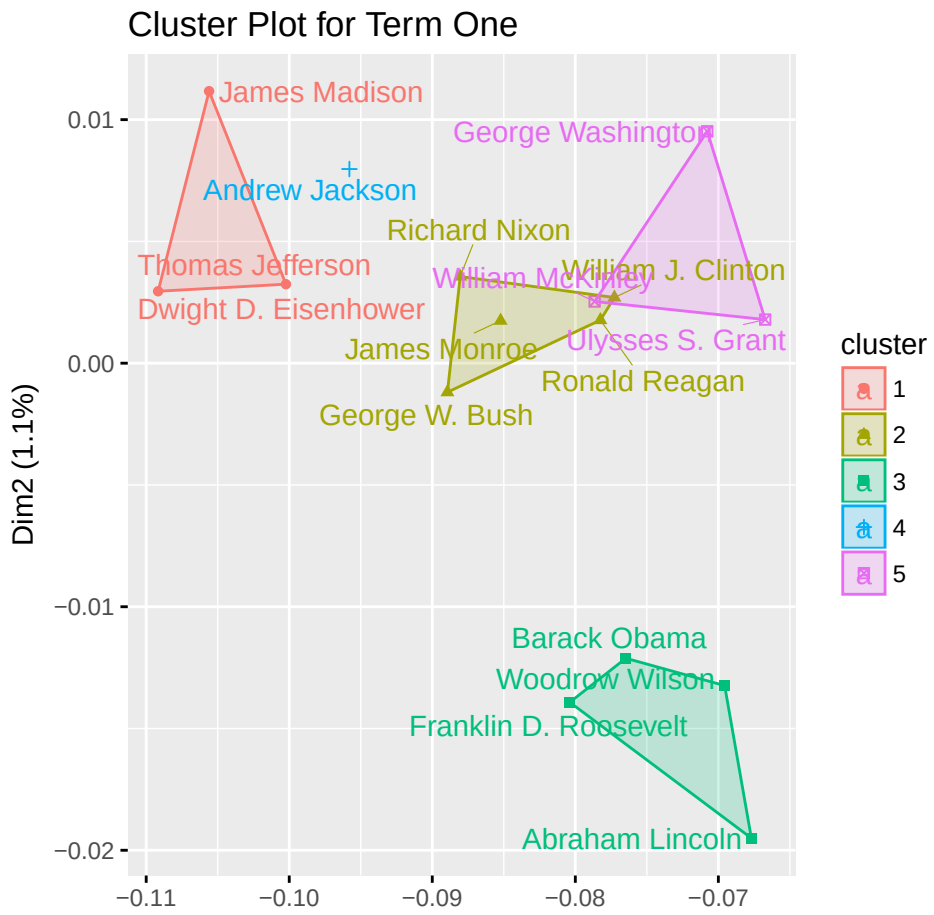
From the plots, we can see that the distributions of emotions seem to be a little different for term one and two speeches, for anger and sadness.

Then we plot the cluster plots.

```
presid.summary.term01=tbl_df(sentence.list.term01)%>%
  group_by(President)%>%
  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)
  )

presid.summary.term01=as.data.frame(presid.summary.term01)
rownames(presid.summary.term01)=as.character((presid.summary.term01[,1]))
km.res=kmeans(presid.summary.term01[,-1], iter.max=200,
              5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
```

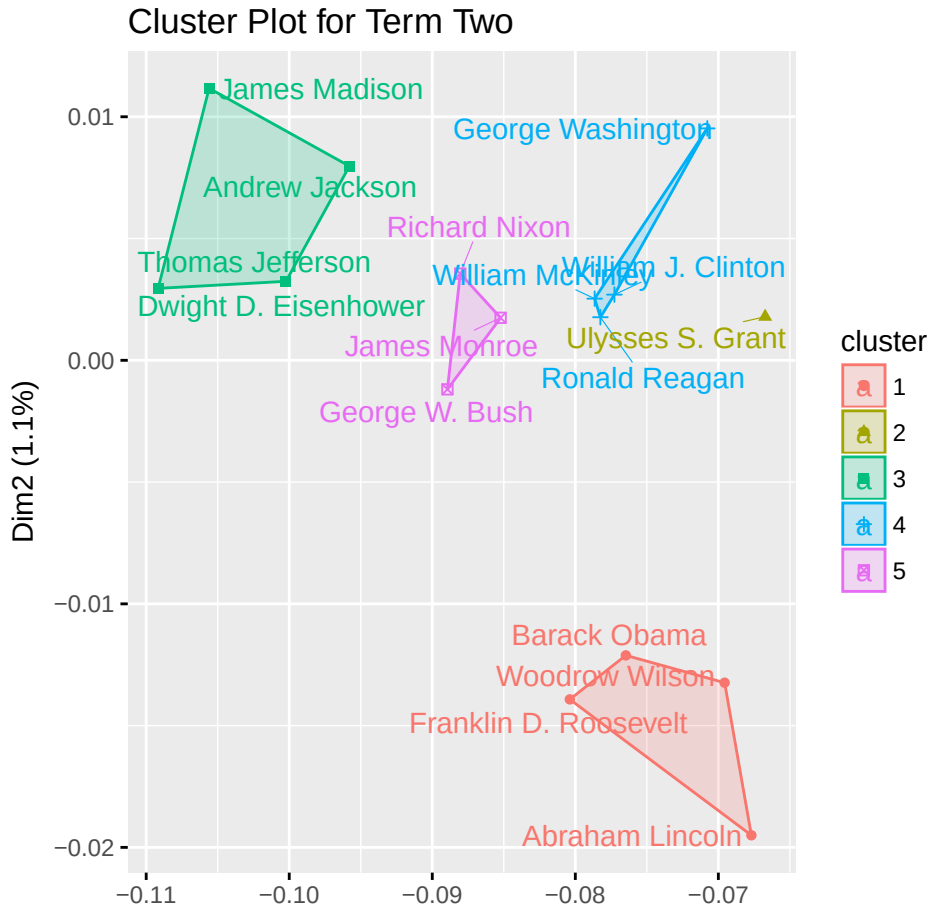
```
            data = presid.summary.term01[,-1], xlab="", xaxt="n",
            show.clust.cent=FALSE, main = "Cluster Plot for Term One")
```




```
presid.summary.term02=tbl_df(sentence.list.term02)%>%
  group_by(President)%>%
  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)
  )

presid.summary.term02=as.data.frame(presid.summary.term02)
rownames(presid.summary.term02)=as.character((presid.summary.term02[,1]))
km.res=kmeans(presid.summary.term01[,-1], iter.max=200,
              5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = presid.summary.term01[,-1], xlab="", xaxt="n",
```

```
        show.clust.cent=FALSE, main = "Cluster Plot for Term Two")
```



Cluster plots for term one and two speech are almost same.