Project1: Will Trump win again? Text mining from inaugural speeches

Chaoyue Tan ct2774

Figure 1: Trump and other presidents

Introduction

What does my project do?

- There have been nearly a dozen one-term presidents who ran for second terms but were denied by voters, but only three one term presidents since World War II. The most recent one term president who lost his re-election bid was George H.W. Bush, a Republican who lost to Democrat Bill Clinton in 1992.
- How about our current president? Will Trump win again if he runs for the second term? My project focused on America's one term presidents - those who ran for, but lost, re-election - through history, and conducted data mining and visualization from those inaugural speeches.
- According to my result, I found there are some similarity between Trump's and other one-term presidents' speeches.

Outline structure of my code

- First, checked libraries needed to be installed and the R notebook environmental settings.

- Then, read in inaugural speeches and did data pre-processing(prepare president list). Generated Term-Document Matrix and Document-Term Matrix.
- Next, inspected wordcloud, conducted sentiment analysis.
- Last, conducted emotion analysis, classification and visualization.

How to run my code

- Be sure to install all the libraries in the right version and check if it's the right version of R
- Download all the folders(data) from github

Step 0 - libraries

```
packages.used <- c("rvest", "tibble", "qdap", "sentimentr",
                   "gplots", "dplyr", "tm", "syuzhet",
                   "factoextra", "beeswarm",
                   "scales", "RColorBrewer",
                   "RANN", "topicmodels",
                   "wordcloud", "splitstackshape",
                   "tidytext", "tidyr")

#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)
}

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("topicmodels")
library("wordcloud")
library("splitstackshape")
```

```
library("tidytext")
library("tidyr")
```

This notebook was prepared with the following environmental settings.

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

```
##                _
## platform       i386-w64-mingw32
## arch           i386
## os             mingw32
## system         i386, mingw32
## status
## major          3
## minor          4.1
## year           2017
## month          06
## day            30
## svn rev        72865
## language       R
## version.string R version 3.4.1 (2017-06-30)
## nickname       Single Candle
```

Step 1 - Read in the speeches and pre-processing

```
folder.path = "../data/InauguralSpeeches/"
speeches = list.files(path = folder.path, pattern = "*.txt")
prex.out = substr(speeches, 6, nchar(speeches)-4)
ff.all<-Corpus(DirSource(folder.path))

folder.path1 = "../data/InauguralOneTerm/"
speechesOneTerm = list.files(path = folder.path1, pattern = "*.txt")
prex.outOneTerm = substr(speechesOneTerm, 6, nchar(speeches)-4)
ff.oneTerm<-Corpus(DirSource(folder.path1))
```

```
#speech of "DonaldTrump", "GeorgeBush", "JimmyCarter", "HerbertHoover", "WilliamHowardTaft", "BenjaminH
#speechesOneTerm = speeches[c(9, 17, 33, 25, 51, 7, 22, 38, 36, 34)]

president.list.OneTerm <- c("GeorgeBush", "JimmyCarter", "HerbertHoover", "WilliamHowardTaft", "Benjamin

president.list <- c("GeorgeWashington", "JohnAdams", "ThomasJefferson", "JamesMadison", "JamesMonroe",
                    "JohnQuincyAdams", "AndrewJackson", "MartinvanBuren", "WilliamHenryHarrison", "James
                   "ZacharyTaylor", "FranklinPierce", "JamesBuchanan", "AbrahamLincoln", "UlyssesSGrant"
                   "RutherfordBHayes", "JamesGarfield", "GroverCleveland-I", "BenjaminHarrison", "Grover
                   "WilliamMcKinley", "TheodoreRoosevelt", "WilliamHowardTaft", "WoodrowWilson", "Warren
                   "CalvinCoolidge", "HerbertHoover", "FranklinDRoosevelt", "HarrySTruman", "DwightDEise
                    "JohnFKennedy", "LyndonBJohnson", "RichardNixon", "JimmyCarter", "RonaldReagan", "Ge
                   "WilliamJClinton", "GeorgeWBush", "BarackObama","DonaldJTrump")
```

Step 2 - Generate Term-Document Matrix and Document-Term Matrix

write TDM functions

```
generateTDM <- function(corpus){
  corpus<-tm_map(corpus, stripWhitespace)
  corpus<-tm_map(corpus, content_transformer(tolower))
  corpus<-tm_map(corpus, removeWords, stopwords("english"))
  corpus<-tm_map(corpus, removeWords, character(0))
  corpus<-tm_map(corpus, removePunctuation)

  #TF-IDF weighted Term-Document Matrix

  TDM <- TermDocumentMatrix(corpus,
                            control = list(weighting = function(x)
                                                 weightTfIdf(x, normalize =FALSE),
                                           stopwords = TRUE))
  TDM.tidy=tidy(TDM)
  TDM.overall=summarise(group_by(TDM.tidy, term), sum(count))

  return(TDM.overall)
}
```

```
generateIndividualTdm <- function(corpus, fileName){
  corpus<-tm_map(corpus, stripWhitespace)
  corpus<-tm_map(corpus, content_transformer(tolower))
  corpus<-tm_map(corpus, removeWords, stopwords("english"))
  corpus<-tm_map(corpus, removeWords, character(0))
  corpus<-tm_map(corpus, removePunctuation)

  #TF-IDF weighted Term-Document Matrix

  TDM <- TermDocumentMatrix(corpus,
                            control = list(weighting = function(x)
                                                 weightTfIdf(x, normalize =FALSE),
                                           stopwords = TRUE))
  TDM.tidy=tidy(TDM)
  TDM.tidy = subset(TDM.tidy,document %in% speeches)

  tdm.indi = subset(TDM.tidy,document == fileName)

  return(tdm.indi)
}
```

write DTM functions

```
generateDTM <- function(corpus){
  corpus<-tm_map(corpus, stripWhitespace)
  corpus<-tm_map(corpus, content_transformer(tolower))
  corpus<-tm_map(corpus, removeWords, stopwords("english"))
```

```
  corpus<-tm_map(corpus, removeWords, character(0))
  corpus<-tm_map(corpus, removePunctuation)

  #Document-Term Matrix

  DTM <- DocumentTermMatrix(corpus,
                            control = list(weighting = function(x)
                                                 weightTfIdf(x, normalize =FALSE),
                                           stopwords = TRUE))
  DTM.tidy <- tidy(DTM)
  DTM.overall <- summarise(group_by(DTM.tidy, term), sum(count))
  r <- list(tidy = DTM.tidy, overall = DTM.overall)

  return(r)
}
```

```
generateIndividualDtm <- function(corpus, fileName){
  corpus<-tm_map(corpus, stripWhitespace)
  corpus<-tm_map(corpus, content_transformer(tolower))
  corpus<-tm_map(corpus, removeWords, stopwords("english"))
  corpus<-tm_map(corpus, removeWords, character(0))
  corpus<-tm_map(corpus, removePunctuation)

  #Document-Term Matrix

  DTM <- TermDocumentMatrix(corpus,
                            control = list(weighting = function(x)
                                                 weightTfIdf(x, normalize =FALSE),
                                           stopwords = TRUE))
  DTM.tidy=tidy(DTM)
  DTM.tidy = subset(DTM.tidy,document %in% speeches)

  dtm.indi = subset(DTM.tidy,document == fileName)

  return(dtm.indi)
}
```

generate TDM & DTM for all speeches, OneTerm presidents and Trump

```
tdm.trump <- generateIndividualTdm(ff.all, "inaugDonaldJTrump-1.txt")
tdm.all <- generateTDM(ff.all)
tdm.oneTerm <- generateTDM(ff.oneTerm)

dtm.trump <- generateIndividualDtm(ff.all, "inaugDonaldJTrump-1.txt")
dtm.all <- generateDTM(ff.all)[[2]]
dtm.oneTerm <- generateDTM(ff.oneTerm)[[2]]
dtm.all.tidy <- generateDTM(ff.all)[[1]]
dtm.oneTerm.tidy <- generateDTM(ff.oneTerm)[[1]]

dtm.all.tidy$president <- substr(dtm.all.tidy$document, 6, nchar(dtm.all.tidy$document)-6)
dtm.oneTerm.tidy$president <- substr(dtm.oneTerm.tidy$document, 6, nchar(dtm.oneTerm.tidy$document)-6)
```

```
#One Term or not

n <- dim(dtm.all.tidy)[1]
#m <- length(president.list.OneTerm)

dtm.all.tidy$isOneTerm <- rep(F, n)

dtm.all.tidy[dtm.all.tidy$president == "DonaldJTrump",]
```

```
## # A tibble: 457 x 5
##                    document           term     count    president isOneTerm
##                       <chr>          <chr>     <dbl>        <chr>     <lgl>
##  1 inaugDonaldJTrump-1.txt           2017  5.857981 DonaldJTrump     FALSE
##  2 inaugDonaldJTrump-1.txt           20th  4.273018 DonaldJTrump     FALSE
##  3 inaugDonaldJTrump-1.txt         accept  1.610053 DonaldJTrump     FALSE
##  4 inaugDonaldJTrump-1.txt         across 10.253130 DonaldJTrump     FALSE
##  5 inaugDonaldJTrump-1.txt         action  2.000000 DonaldJTrump     FALSE
##  6 inaugDonaldJTrump-1.txt administration  0.857981 DonaldJTrump     FALSE
##  7 inaugDonaldJTrump-1.txt        affairs  1.157541 DonaldJTrump     FALSE
##  8 inaugDonaldJTrump-1.txt            aid  1.273018 DonaldJTrump     FALSE
##  9 inaugDonaldJTrump-1.txt       airports  4.857981 DonaldJTrump     FALSE
## 10 inaugDonaldJTrump-1.txt     allegiance  7.715962 DonaldJTrump     FALSE
## # ... with 447 more rows
```

```
for (i in president.list.OneTerm){
  mylist <- grep(i, dtm.all.tidy$president)
  dtm.all.tidy$isOneTerm[mylist] <- T
}
```

Step 3 - Inspect wordclouds

```
generateWordCloud <- function(tdm, mytitle){
  wordcloud(tdm$term, tdm$`sum(count)`,
          scale=c(5,0.5),
          max.words=100,
          min.freq=1,
          random.order=FALSE,
          rot.per=0.3,
          use.r.layout=T,
          random.color=FALSE,
          colors=brewer.pal(9,"Blues"))
  title(main = mytitle)
}
```

```
generateWordCloudIndi <- function(tdm, mytitle){
  wordcloud(tdm$term, tdm$count,
          scale=c(5,0.5),
          max.words=100,
          min.freq=1,
          random.order=FALSE,
          rot.per=0.3,
          use.r.layout=T,
```

```
          random.color=FALSE,
          colors=brewer.pal(9,"Blues"))
  title(main = mytitle)
}
```

generate wordcloud for all all speeches, OneTerm presidents and Trump

```
par(mfcol = c(2, 2))

#wordcloud(tdm.oneTerm$term, tdm.oneTerm$`sum(count)`, scale = c(2, 0.5))

generateWordCloud(tdm.all, 'AllSpeeches')
```

```
## Warning in strwidth(words[i], cex = size[i], ...): font width unknown for
## character 0x9d

## Warning in strwidth(words[i], cex = size[i], ...): font width unknown for
## character 0x9d

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font width unknown for character 0x9d

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font width unknown for character 0x9d

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font metrics unknown for character 0x9d
```

```
generateWordCloud(tdm.oneTerm, 'AllOneTerm')
generateWordCloudIndi(tdm.trump, 'Trump')
```

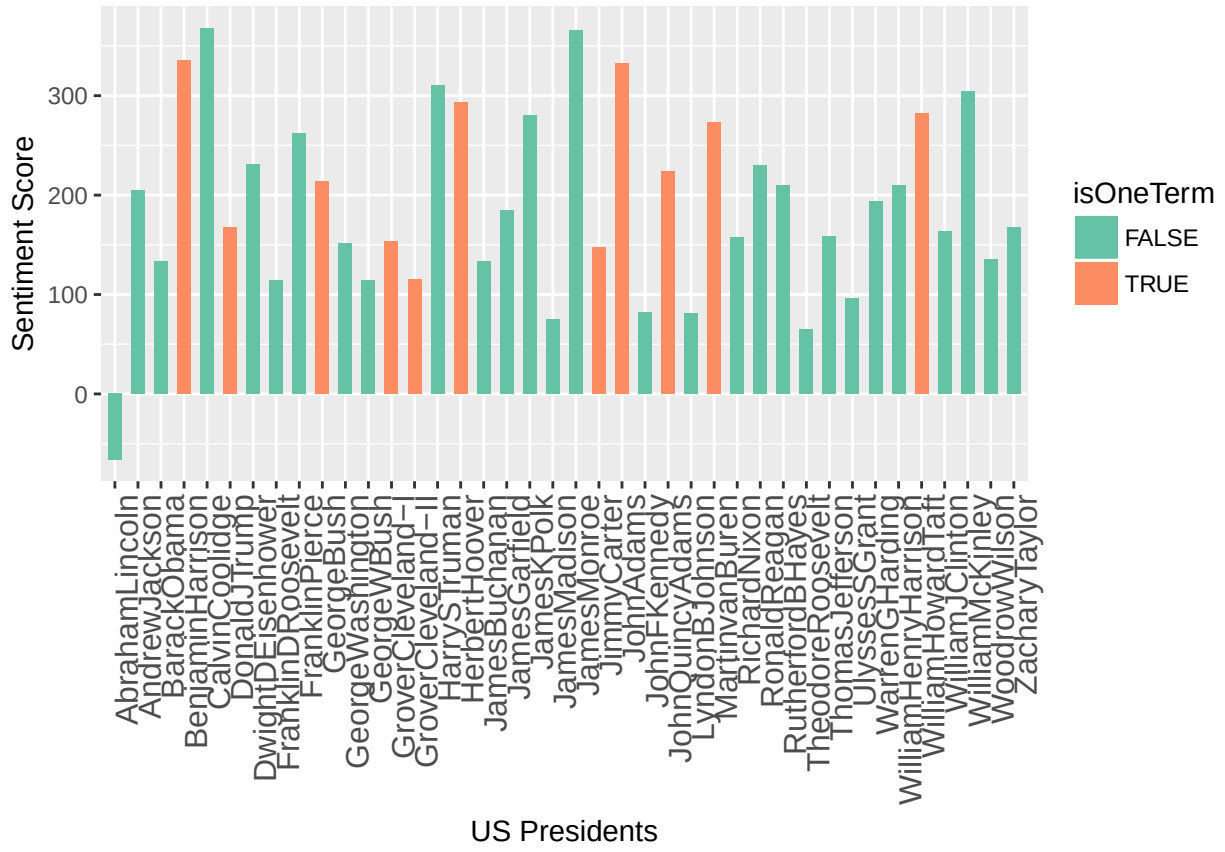
The wordcloud of Trump's speech reminds me exactly of his speech even if I'm not a big fan of politics. As the word "America" became the probably most used word in both overall inaugural speeches and President Trump's inaugural speech, the wordcoud looks different back to the one- term presidents' speeches: they focused more on stuffs like "business", "south", "union" and "tariff". I make some assumption for the reasons: it's both time and historical background for them to talk about totally different things in their speeches. Only three of them are from World War II after all. They were very likely to talk about the wars and discrimination. So it's safe for us to say Trump's speech is not like those one term presidents in the wordcloud way.

There is one fact that catch my attention: the use of the N-word in previous speeches. And I looked it up and found it's safe to use it until back to 1960s.

A wordcloud doesn't allow us to have an overall appreciation of the speech. One would like to know the overall atmosphere - and to be able to know if the speech is optimistic or not. To do that we will focus on sentiment analysis in the next step.

Step4 - sentiment analysis

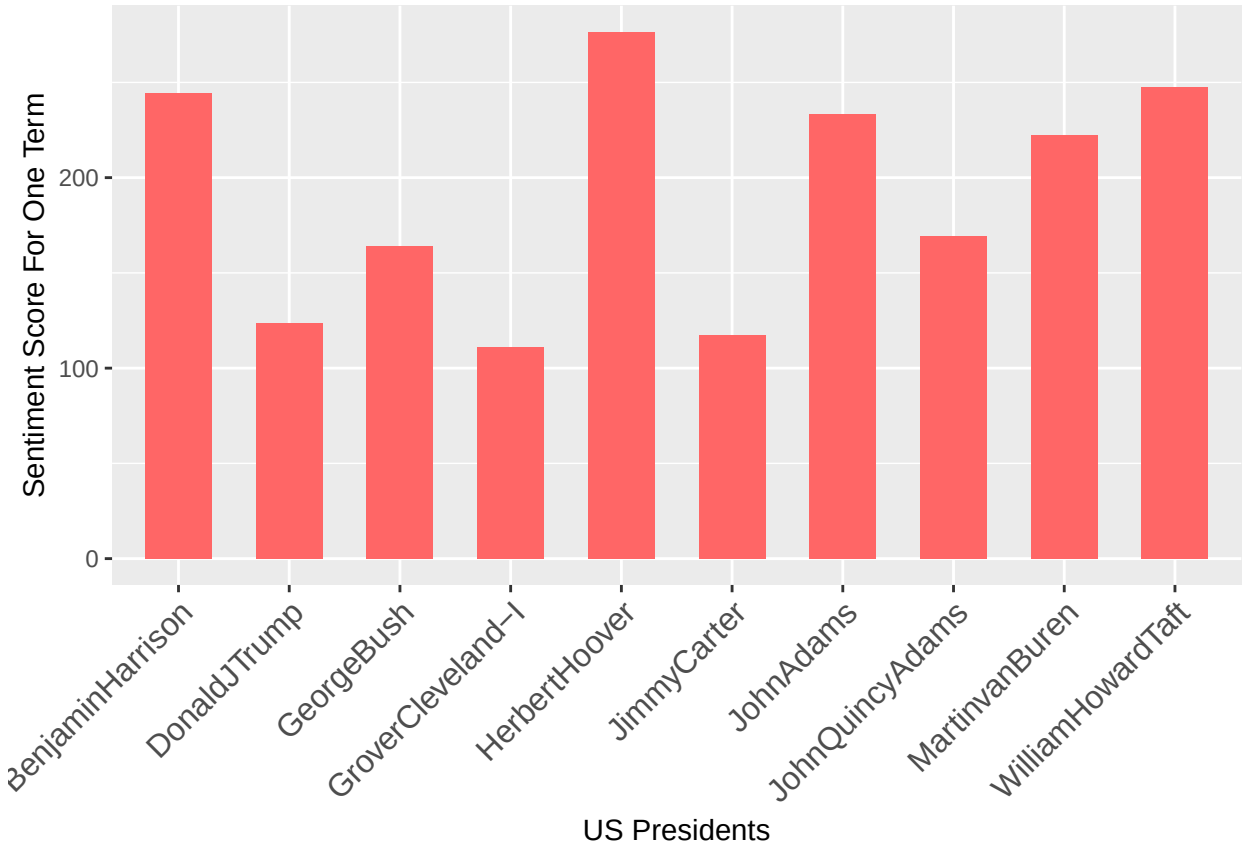
```
dtm.sentiment <- inner_join(dtm.all.tidy,
                            get_sentiments("nrc"),
                            by=c("term"="word"))

#spread sentiments in different columns

dtm.sentiment <- spread(dtm.sentiment, sentiment, count, fill=0)

#compute a sentiment score for each speech

sentiment.score <- mutate(dtm.sentiment, feeling = positive - negative)

sentiment.score <- summarise(group_by(sentiment.score, document, president, isOneTerm),
                             score = sum(feeling))

sentimentPlot <- ggplot(sentiment.score, aes(x=sentiment.score$president,
                                y=sentiment.score$score,
                                fill=sentiment.score$isOneTerm)) +
  geom_bar(stat="identity", position = "dodge", width=.6) +
  ylab("Sentiment Score") +
  xlab("US Presidents") +
  scale_fill_manual("isOneTerm", values=brewer.pal(2,"Set2")) +
  theme(axis.text.x = element_text(angle = 90, size=12, hjust = 1))

sentimentPlot
```



I applied 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). Then I drew a sentiment analysis plot of all speeched based on if they are from one term presidents(including trump) or not. We can see most one term presidents'(Benjamin Harrison, George Bush) sentiment scores are above average.

```
#get sentiments for OneTerm

dtm.oneTerm.sentiment <- inner_join(dtm.oneTerm.tidy,
                                    get_sentiments("nrc"),
                                    by=c("term"="word"))

#spread sentiments in different columns

dtm.oneTerm.sentiment <- spread(dtm.oneTerm.sentiment, sentiment, count, fill=0)

#compute a sentiment score for each OneTerm speech

sentiment.score.oneTerm <- mutate(dtm.oneTerm.sentiment, feeling = positive - negative)

sentiment.score.oneTerm <- summarise(group_by(sentiment.score.oneTerm, document, president),
                                     score = sum(feeling))

sentimentPlot.oneTerm <- ggplot(sentiment.score.oneTerm, aes(x=sentiment.score.oneTerm$president,
                                      y=sentiment.score.oneTerm$score)) +
  geom_bar(stat="identity", position = "dodge", width=.6, fill = "#FF6666") +
  ylab("Sentiment Score For One Term") +
```

```
  xlab("US Presidents") +
  #scale_fill_manual("president", values=brewer.pal(10,"Set1")) +
  theme(axis.text.x = element_text(angle = 45, size=12, hjust = 1))

sentimentPlot.oneTerm
```



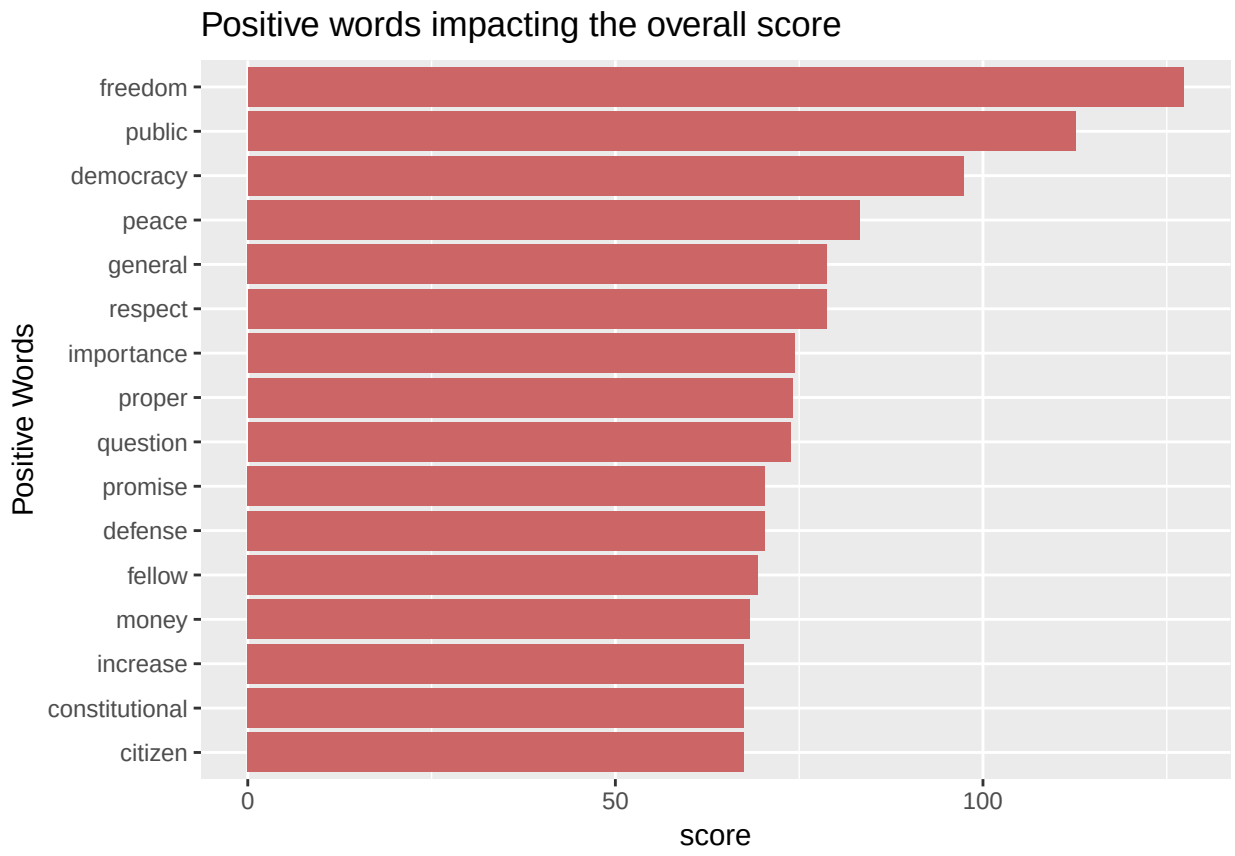
Trump's speech, compared with other one term speeches, got almost lowest sentiment score. What would be interesting is to know if this difference came from a less optimistic (positive lexicon) or a less pessimistic mindset.

```
#get sentiments for Trump

dtm.trump.sentiment <- inner_join(dtm.trump,
                                  get_sentiments("nrc"),
                                  by=c("term"="word"))

#spread sentiments in different columns

dtm.trump.sentiment <- spread(dtm.trump.sentiment, sentiment, count, fill=0)
```

```
par(mfcol = c(2, 3))

#select positive words

word.score.positive <- summarise(group_by(dtm.sentiment, term), score=sum(positive))
word.score.positive <- arrange(word.score.positive, desc(score))
```

```
#plot

word.score.positive %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
  ggplot(aes(x=term, y=score, fill=isOneTerm)) +
    geom_bar(stat="identity", fill="#CC6666") +
    #geom_bar(stat="identity") +
    xlab("Positive Words") +
    ggtitle("Positive words impacting the overall score") +
    coord_flip()
```

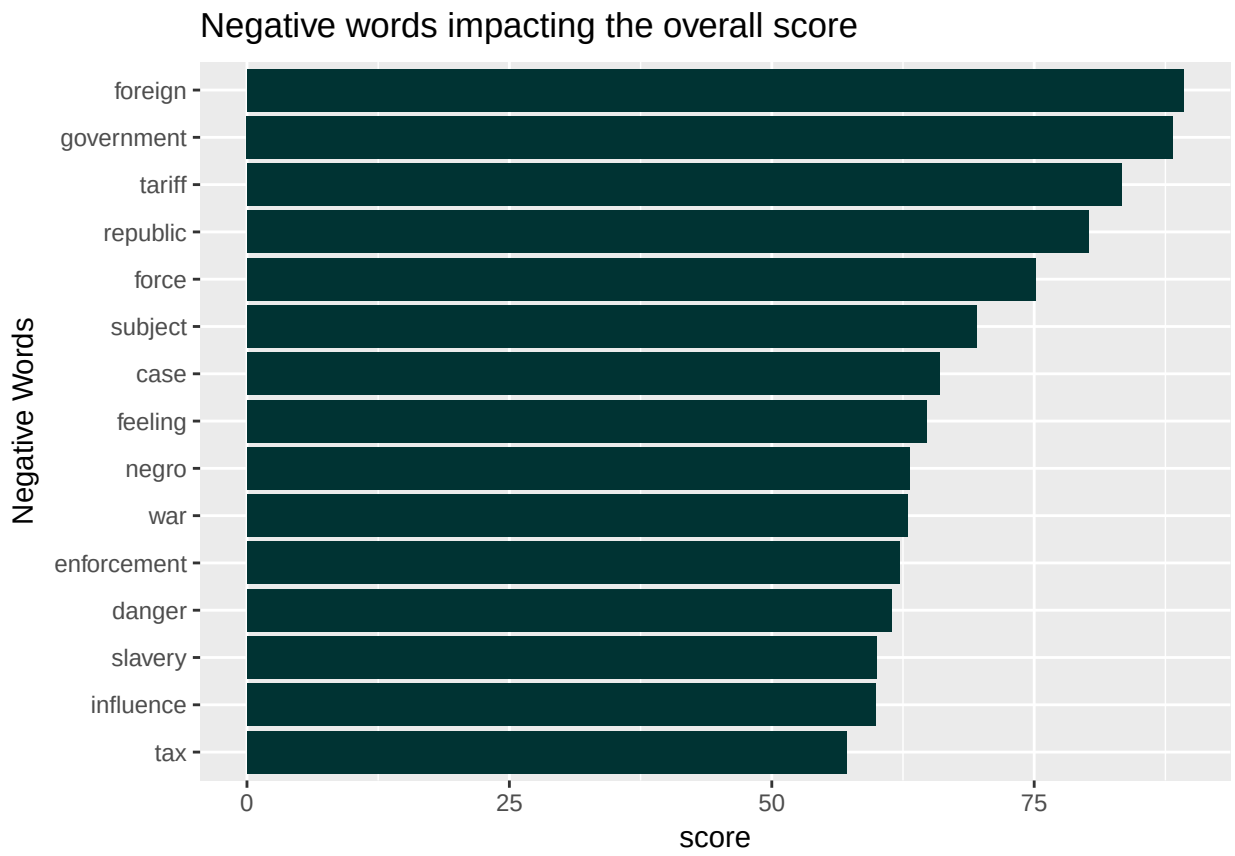
```
## Selecting by score
```




```
#select negative words

word.score.negative <- summarise(group_by(dtm.sentiment, term), score=sum(negative))
word.score.negative <- arrange(word.score.negative, desc(score))

#plot

word.score.negative %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
```

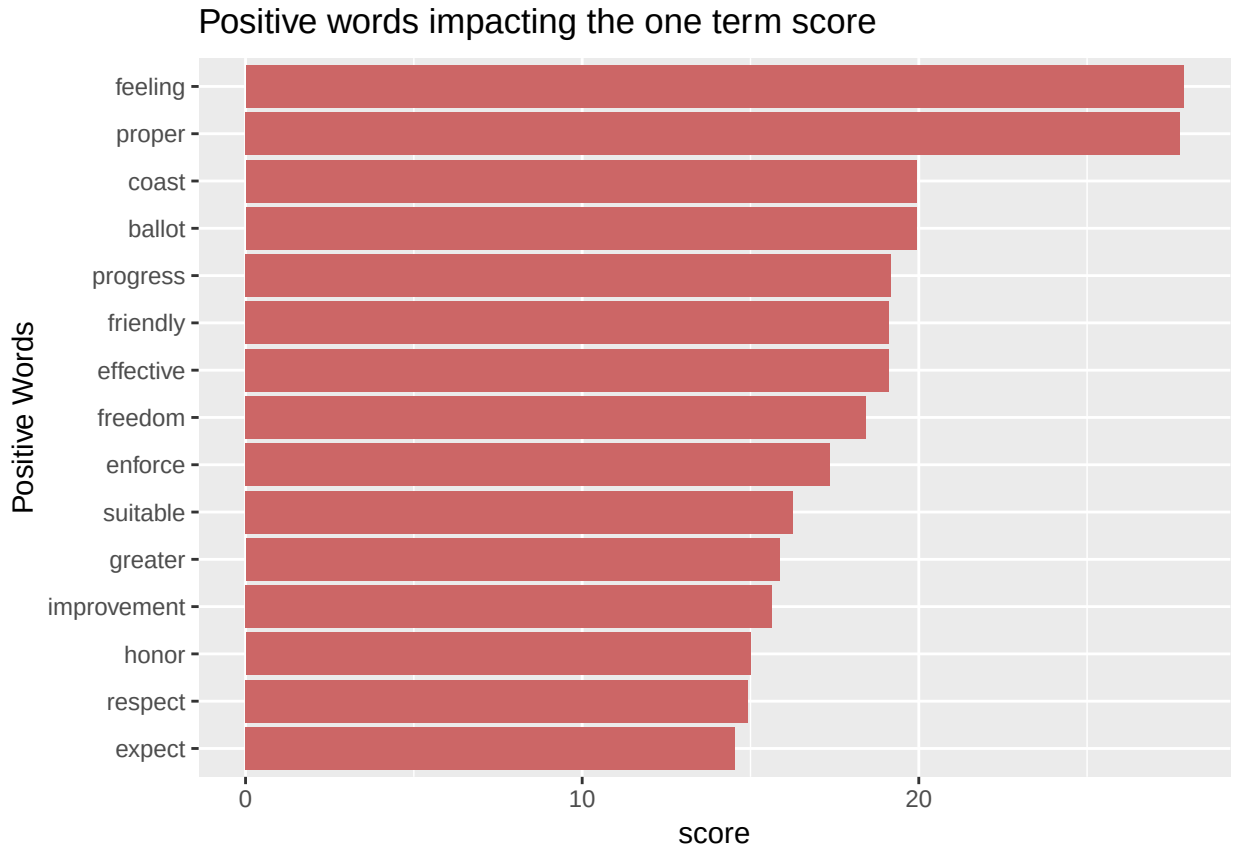
```
  ggplot(aes(x=term, y=score)) +
    geom_bar(stat="identity", fill="#003333") +
    xlab("Negative Words") +
    ggtitle("Negative words impacting the overall score") +
    coord_flip()
```

```
## Selecting by score
```



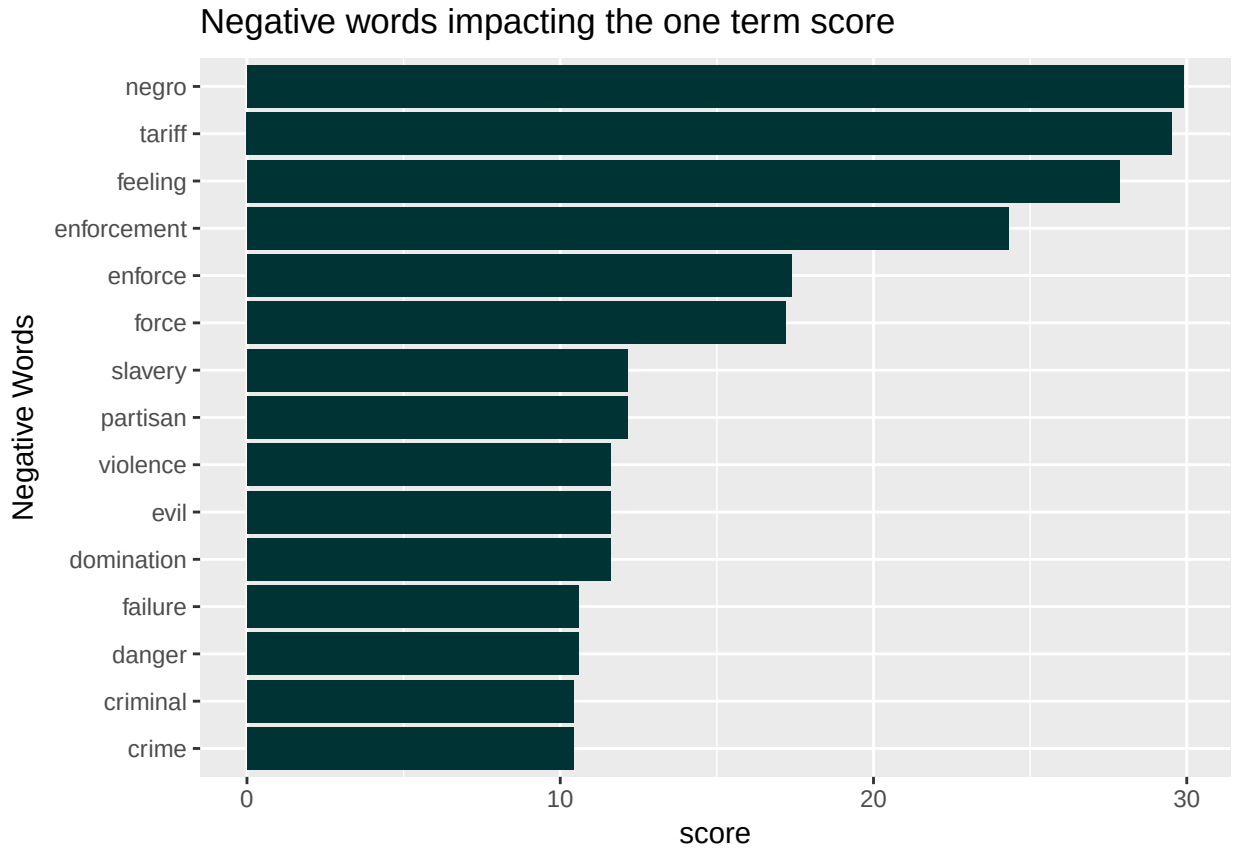

```
#select positive words for one term

word.score.positive.oneTerm <- summarise(group_by(dtm.oneTerm.sentiment, term), score=sum(positive))
word.score.positive.oneTerm <- arrange(word.score.positive.oneTerm, desc(score))

#plot

word.score.positive.oneTerm %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
  ggplot(aes(x=term, y=score)) +
    geom_bar(stat="identity", fill="#CC6666") +
    xlab("Positive Words") +
    ggtitle("Positive words impacting the one term score") +
    coord_flip()
```

```
## Selecting by score
```



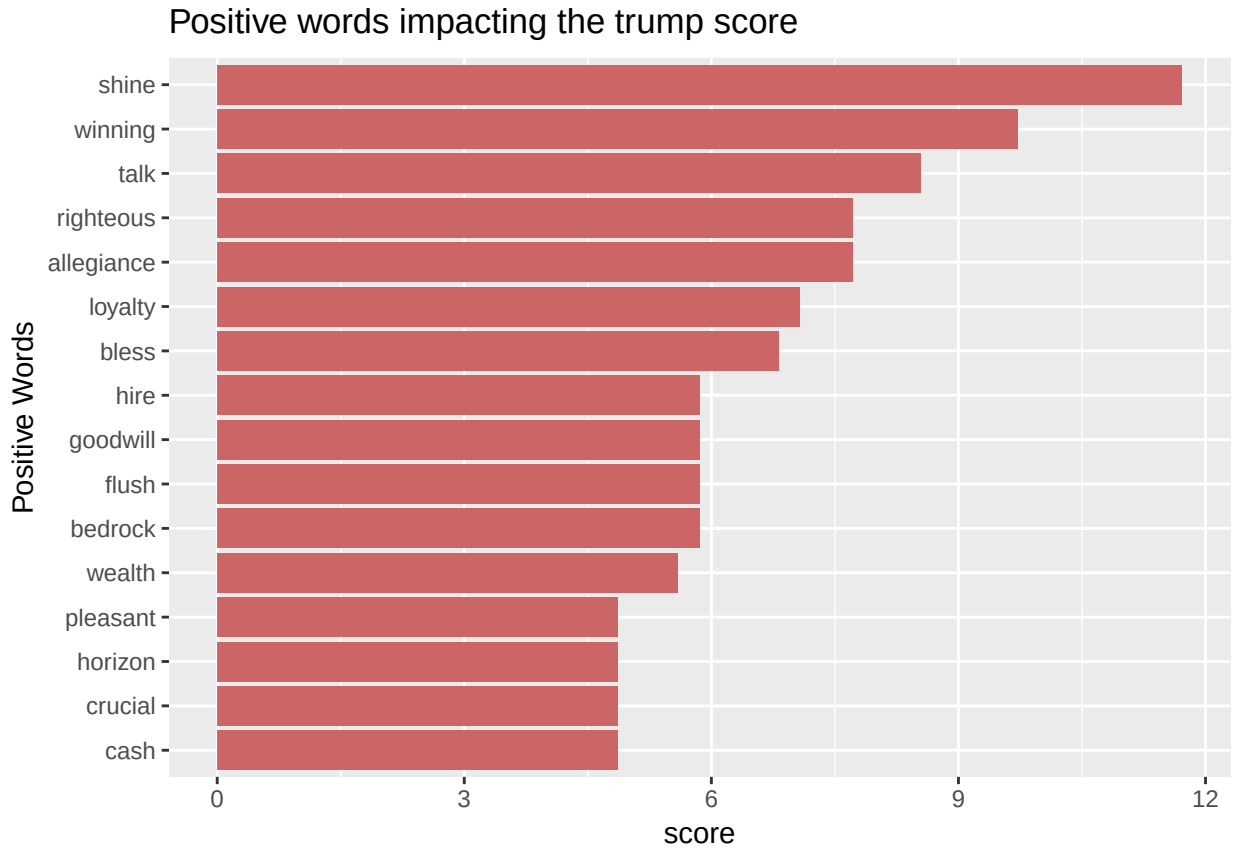

```
#select negative words for one term

word.score.negative.oneTerm <- summarise(group_by(dtm.oneTerm.sentiment, term), score=sum(negative))
word.score.negative.oneTerm <- arrange(word.score.negative.oneTerm, desc(score))

#plot

word.score.negative.oneTerm %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
  ggplot(aes(x=term, y=score)) +
    geom_bar(stat="identity", fill="#003333") +
    xlab("Negative Words") +
    ggtitle("Negative words impacting the one term score") +
    coord_flip()
```

```
## Selecting by score
```



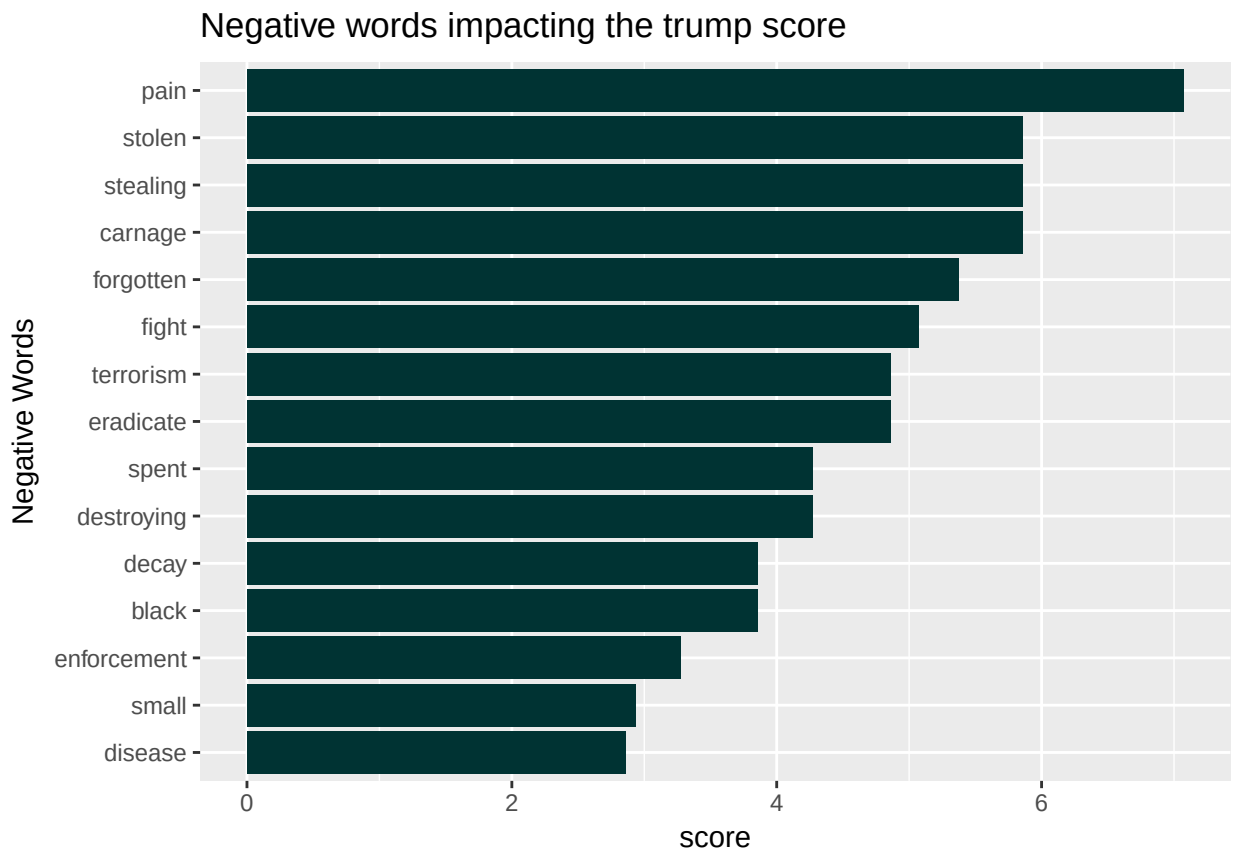

```
#select positive words for Trump

word.score.positive.trump <- summarise(group_by(dtm.trump.sentiment, term), score=sum(positive))
word.score.positive.trump <- arrange(word.score.positive.trump, desc(score))

#plot

word.score.positive.trump %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
  ggplot(aes(x=term, y=score)) +
    geom_bar(stat="identity", fill="#CC6666") +
    xlab("Positive Words") +
    ggtitle("Positive words impacting the trump score") +
    coord_flip()
```

```
## Selecting by score
```




```
#select negative words for Trump

word.score.negative.trump <- summarise(group_by(dtm.trump.sentiment, term), score=sum(negative))
word.score.negative.trump <- arrange(word.score.negative.trump, desc(score))

#plot

word.score.negative.trump %>%
  arrange(score) %>%
  mutate(term = factor(term, term)) %>%
  top_n(15) %>%
  ggplot(aes(x=term, y=score)) +
    geom_bar(stat="identity", fill="#003333") +
    xlab("Negative Words") +
    ggtitle("Negative words impacting the trump score") +
    coord_flip()
```

```
## Selecting by score
```



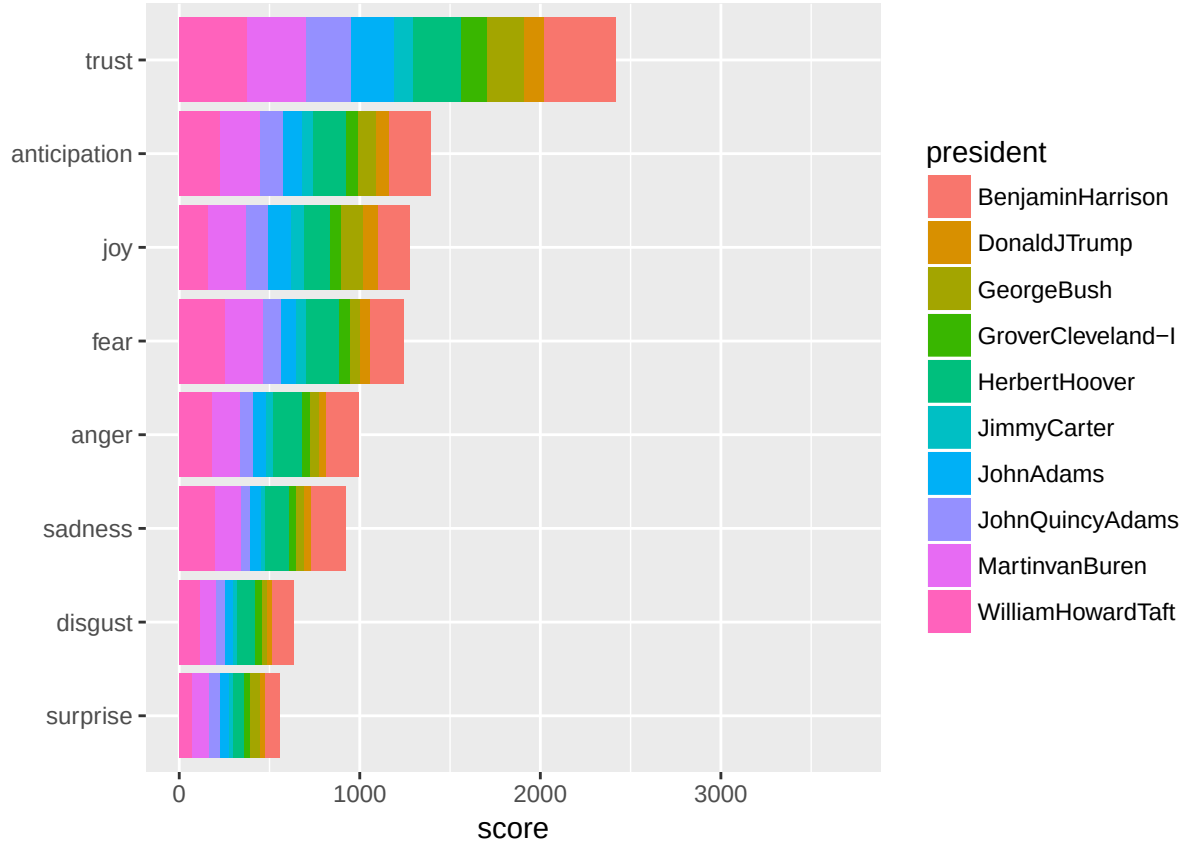
I drew the highest score positive and negative words for all speeches, one term and trump. As for positive words, the words "public", "democracy" and "freedom" are widely used in all speeches and contribute to positive scores, while "feeling" and "proper" are most used in the one term speeches. And the most-used words are similar for all speeches and one-terms, while it's a little different in Trump's positive words. It's the same for negative words. So both positive and neagtive words contribute to the different sentiment score of Trump from one-term presidents.

Step5 - emotion analysis

We can also display the emotions present in the speeches. Eight emotions will be displayed: trust, surprise, sadneess, joy feat, disgust, anticipation and anger. Those emotions are mapped with specific words from the speeches. The goal of this step is to determine what emotions are responsible for the optimistic or permistic mindset of presidents.

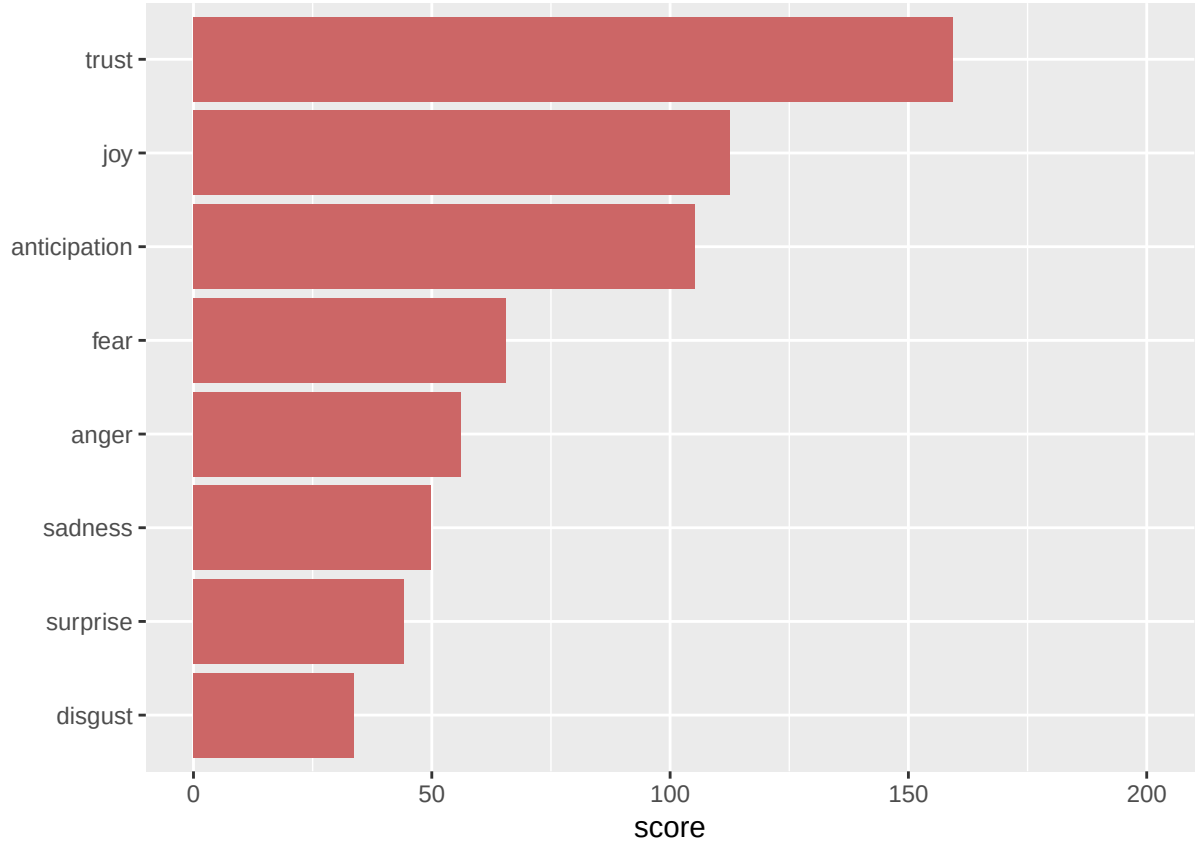
```
par(mfcol = c(1, 2))

emotion.score <- summarise(group_by(dtm.oneTerm.sentiment, president),
                           anger.score = sum(anger),
                           joy.score = sum(joy),
                           anticipation.score = sum(anticipation),
                           disgust.score = sum(disgust),
                           fear.score = sum(fear),
                           sadness.score = sum(sadness),
                           surprise.score = sum(surprise),
                           trust.score = sum(trust))
```

```
emotion.score <- gather(emotion.score, emotion, score,
                c(joy.score, anger.score, disgust.score, fear.score,
                  anticipation.score, sadness.score, surprise.score, sadness.score,
                  trust.score))

emotion.score$emotion <- substr(emotion.score$emotion, 0, nchar(emotion.score$emotion)-6)

emotion.score.plot <- ggplot(emotion.score,
            aes(x=reorder(emotion,score), y=score, fill=president)) +
  geom_bar(stat="identity", position="stack") +
  coord_flip() +
  xlab("") +
  #facet_wrap(~term, nrow=2) +
  expand_limits(y=c(0,3700))

emotion.score.plot
```




```
emotion.score.trump <- summarise(group_by(dtm.trump.sentiment),
                          anger.score = sum(anger),
                          joy.score = sum(joy),
                          anticipation.score = sum(anticipation),
                          disgust.score = sum(disgust),
                          fear.score = sum(fear),
                          sadness.score = sum(sadness),
                          surprise.score = sum(surprise),
```

```
                          trust.score = sum(trust))

emotion.score.trump <- gather(emotion.score.trump, emotion, score,
               c(joy.score, anger.score, disgust.score, fear.score,
                 anticipation.score, sadness.score, surprise.score, sadness.score,
                 trust.score))

emotion.score.trump$emotion <- substr(emotion.score.trump$emotion, 0, nchar(emotion.score.trump$emotion)

emotion.score.plot.trump <- ggplot(emotion.score.trump,
            aes(x=reorder(emotion,score), y=score)) +
  geom_bar(stat="identity", position="stack", fill = "#CC6666") +
  coord_flip() +
  xlab("") +
  #facet_wrap(~term, nrow=2) +
  expand_limits(y=c(0,200))

emotion.score.plot.trump
```

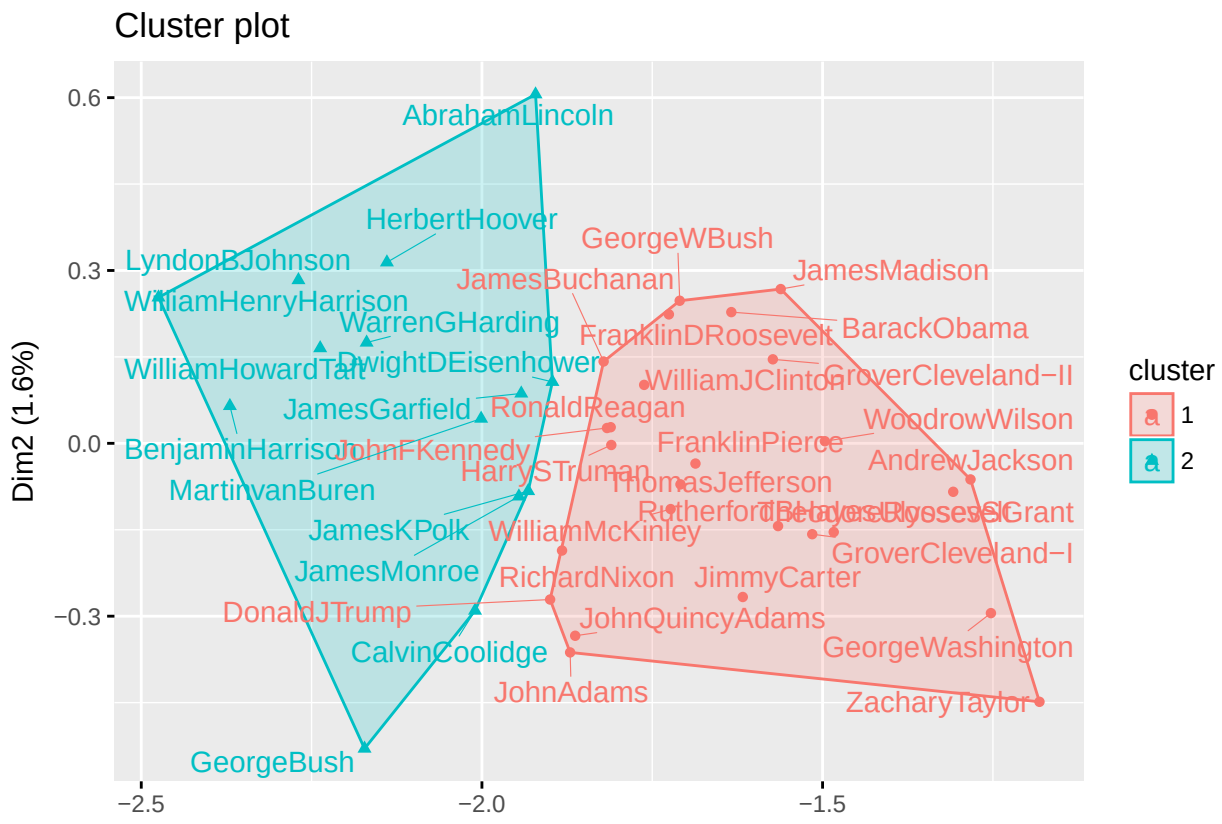


I found the emotion score of trump and other one-term presidents are pretty much the same. Trust is indeed a very important emotion, which takes the more space in both plots. The same enotion contribute to their sentiment. Their speeches are similar in the emotion analysis aspect.

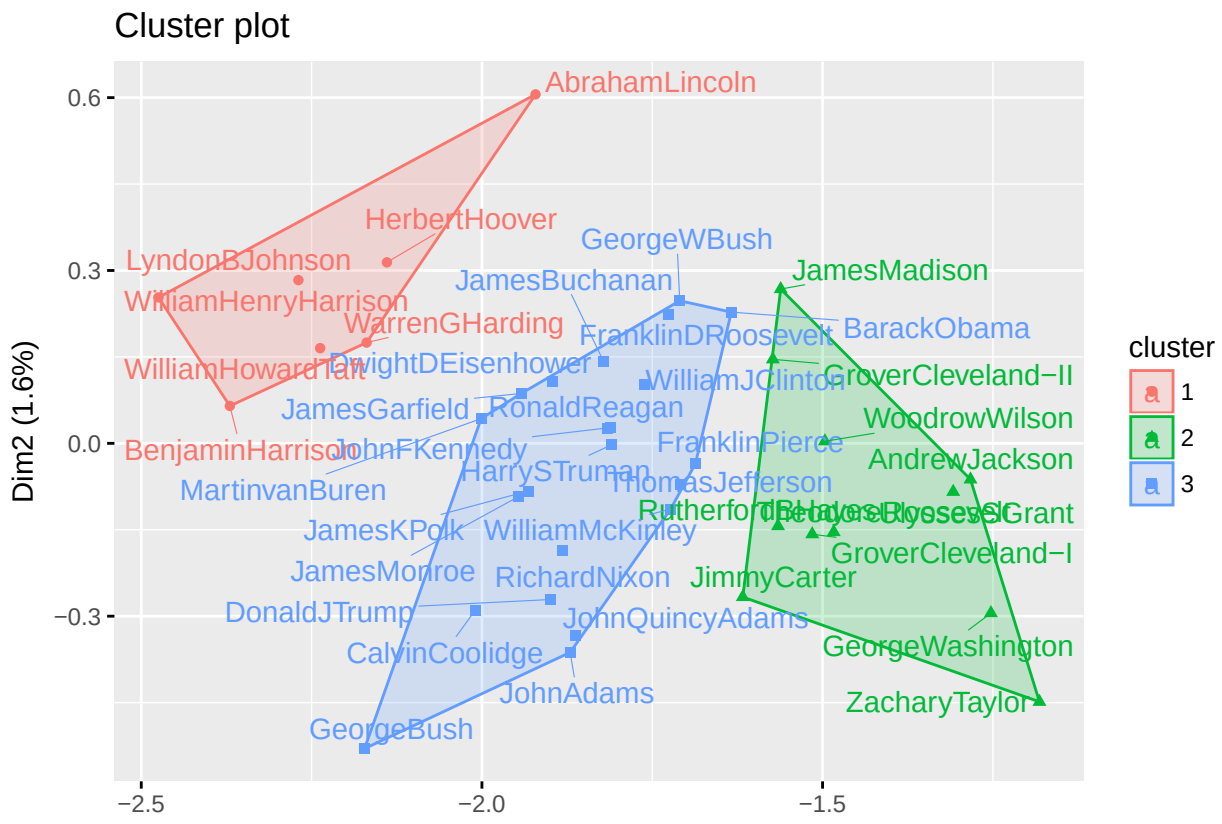
Step6 - emotion cluster plot

The fact that the speeches from both one-term presidents and trump share similar top emotions intrigues me to see whether I can perform a cluster analysis to classify these speeches into different classes based on their emotions. The Cluster analysis is based on the emotion scores of each speech and k-means cluster algorithm.

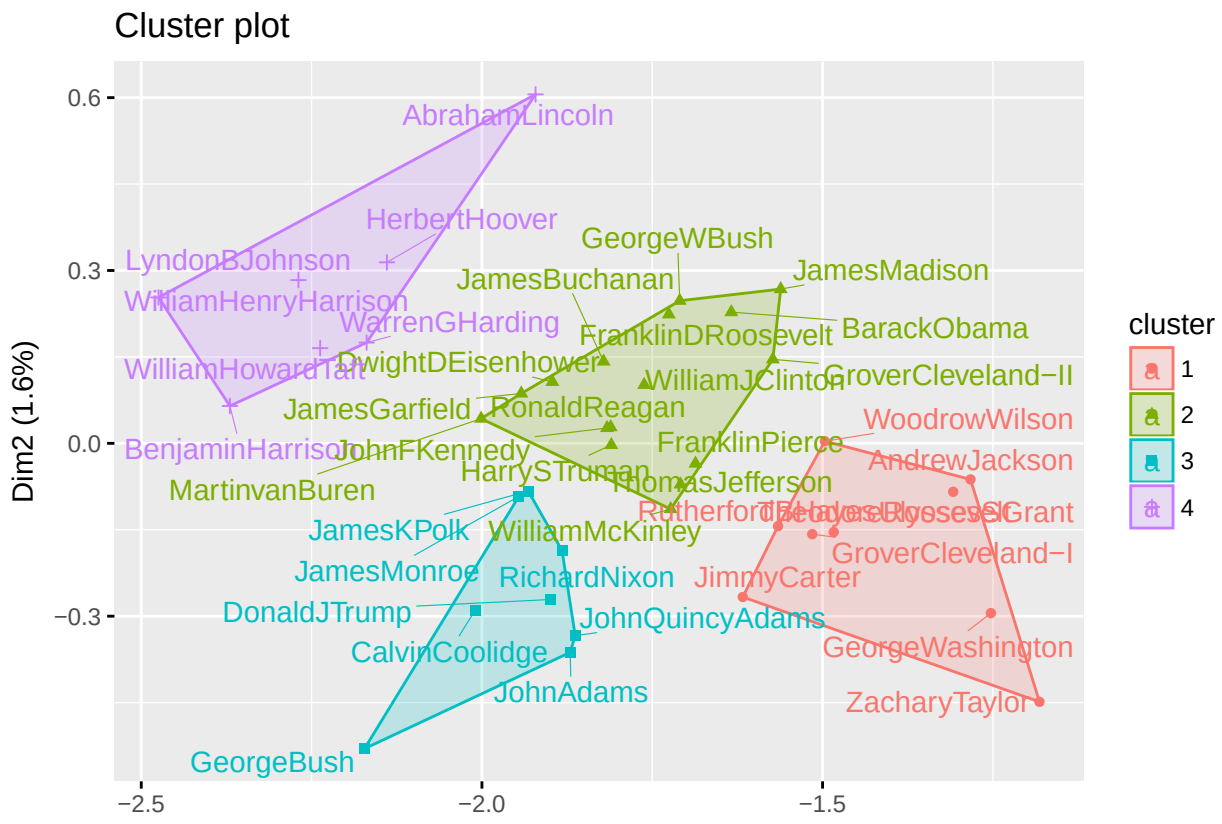
```
presid.summary=tbl_df(dtm.sentiment)%>%

  filter(president %in% c(president.list))%>%
  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=as.data.frame(presid.summary)
rownames(presid.summary)=as.character((presid.summary[,1]))

par(mfcol = c(2, 2))

km.res=kmeans(presid.summary[,-1],2,iter.max=200,nstart=5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = presid.summary[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```



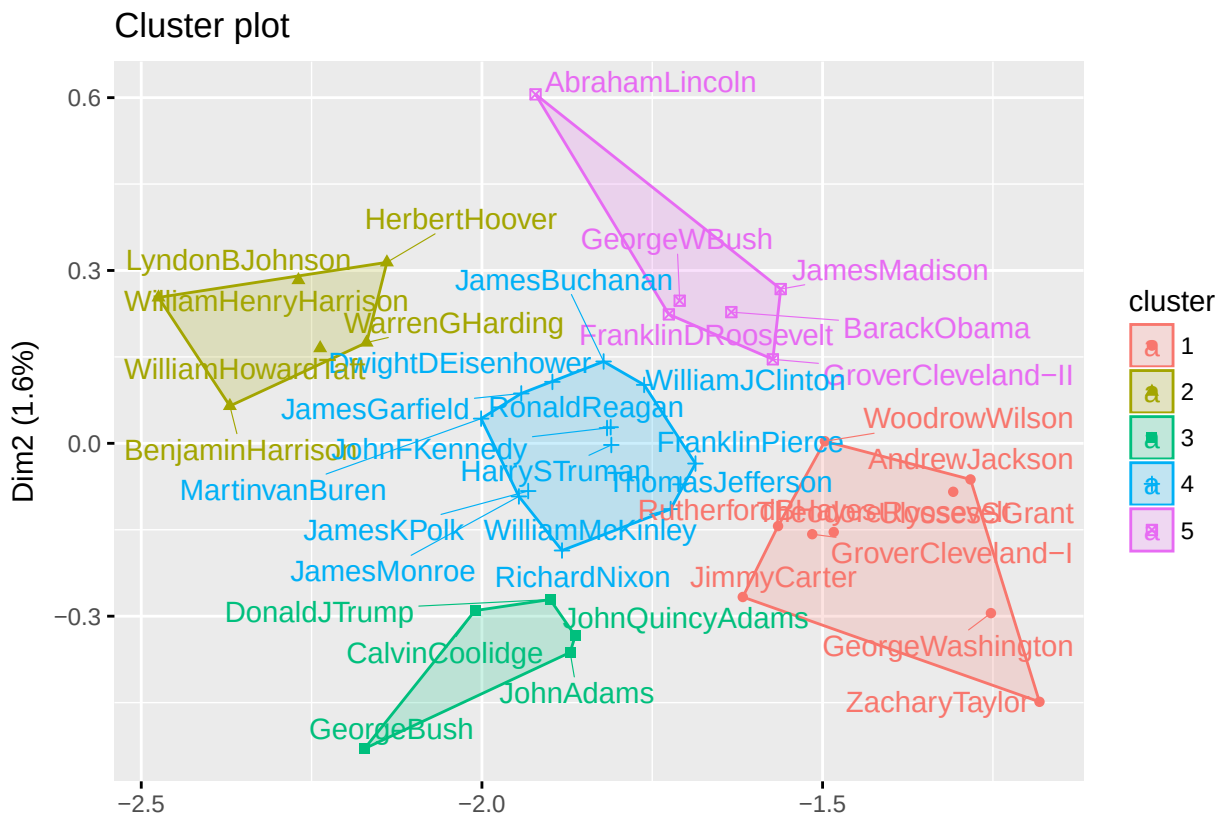

```
km.res=kmeans(presid.summary[,-1],3,iter.max=200,nstart=5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = presid.summary[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```




```
km.res=kmeans(presid.summary[,-1],4,iter.max=200,nstart=5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = presid.summary[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```



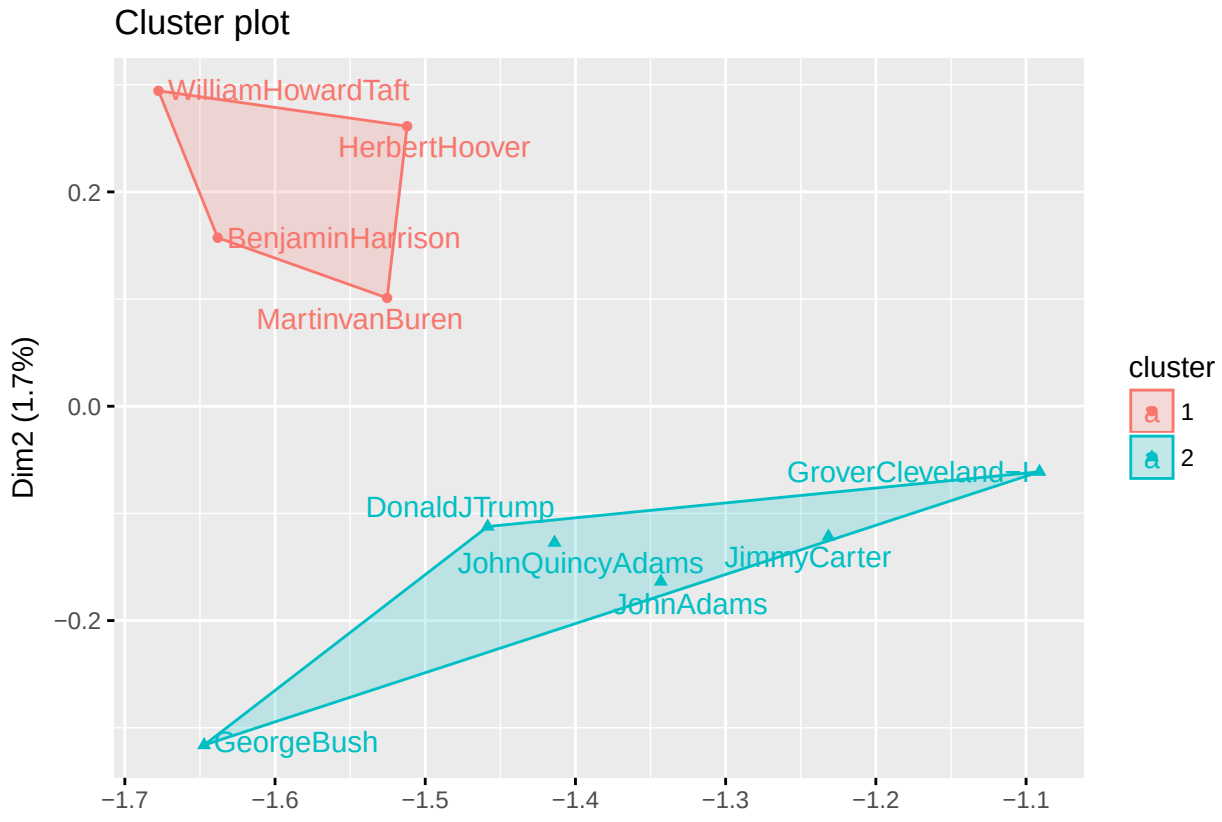

```
km.res=kmeans(presid.summary[,-1],5,iter.max=200,nstart=5)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = presid.summary[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```



First, I tried to classified all speeches into 2,3,4 and 5 classes. Basically, Trump's is always in the same cluster with George Bush, John Admas and John Quincy Adams, who are one-term presidents. We can see there's some similarity in their inaugural speeches. While other one-term presidents like Benjamin Harrison, Herbert Hoover and WilliamHowardTaft always stay together.

```
presid.summaryOneTerm <- tbl_df(dtm.oneTerm.sentiment)%>%

  filter(president %in% c(president.list.OneTerm))%>%
  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.summaryOneTerm <- as.data.frame(presid.summaryOneTerm)
rownames(presid.summaryOneTerm)=as.character((presid.summaryOneTerm[,1]))



km.resOneTerm=kmeans(presid.summaryOneTerm[,-1],2,iter.max=200,nstart=5)
```

```
fviz_cluster(km.resOneTerm,
             stand=F, repel= TRUE,
             data = presid.summaryOneTerm[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```



Then I classifies all the one-term presidents into two classes, I got similar result for Trump's class.

Conclusion

From the above analysis, I found there are some similarity between Trump's and other one-term presidents' speeches, such as their emotions. But in other aspects like sentiments and wordclouds, they are different due to historical reasons.