Assignment 3 jn2587

Jiangshan Ni

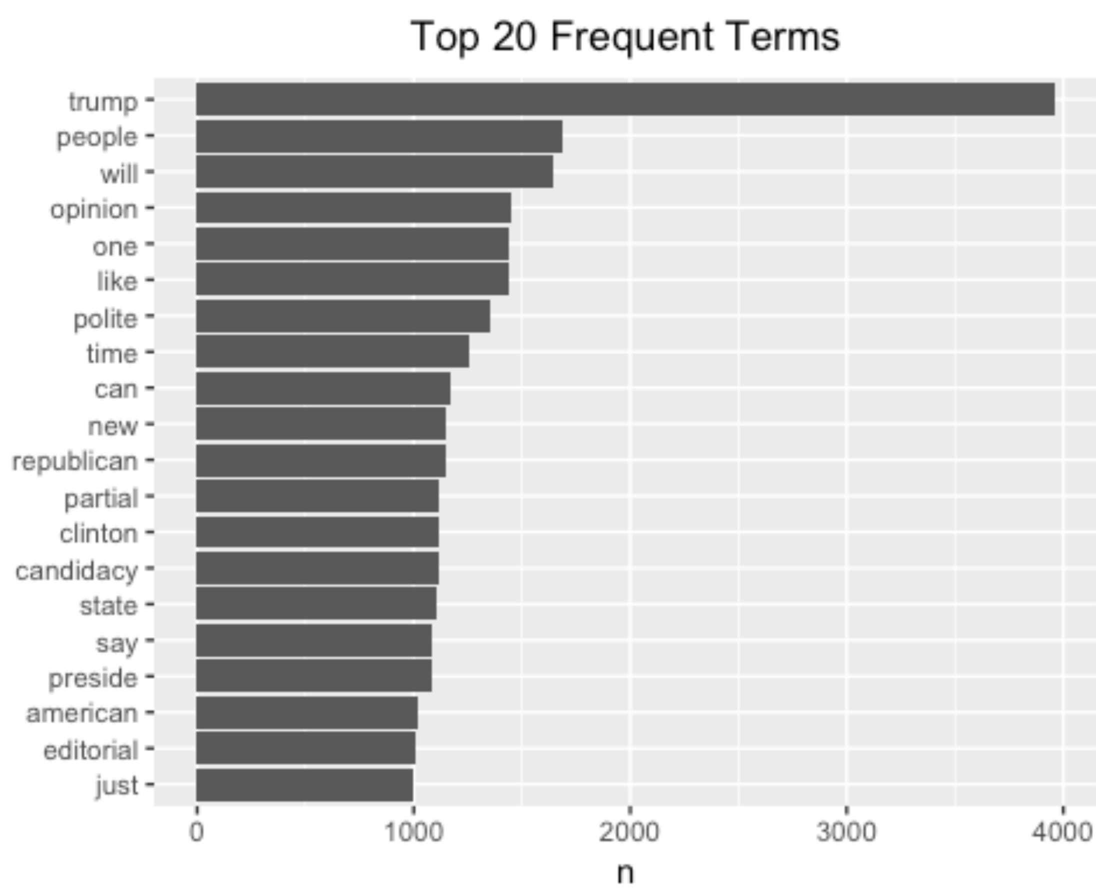
3/29/2017

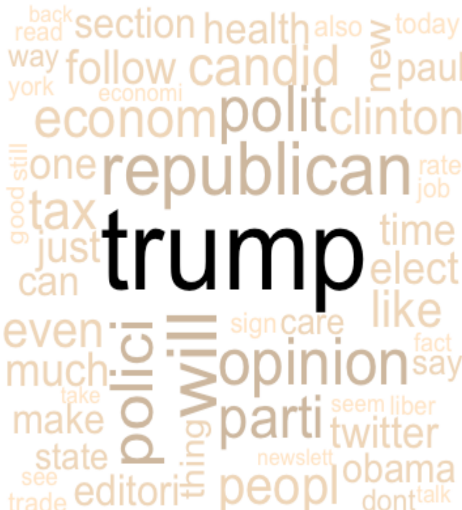
1. Comparing NY Times Columnists.

I want to explore how these five columnists compare and differ.I use different graphical displays to show how these five columnist differ in the words they use. From the wordcloud, I could see that the most frequency words they used in the their articles.

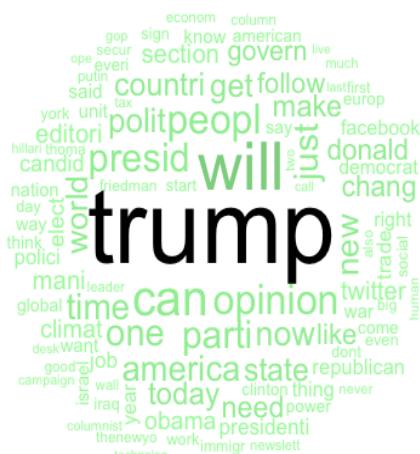
As we could see, all of them used "TRUMP" as the most frequency word. Since the political environment is changing rapidly, all of the columnist are following the trend and explore the TRUMP policy in New York Times.

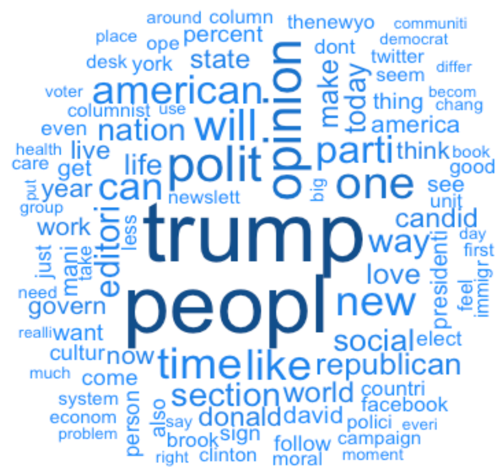
As we could see, from standard graph, the graph directely shows the top 20 frequent terms written by five columnists in the New York Times. The number one is the word " trump". The second one is "people". Especially, the top one frequency are well above the average.

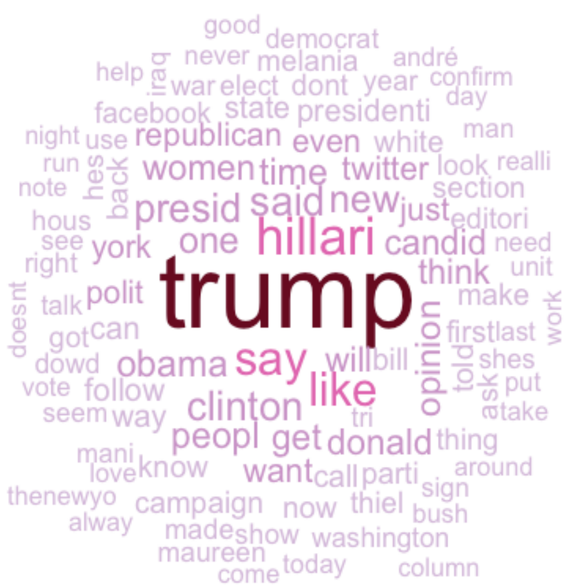



back read section health also new today
way follow candid paul
york economi
econom polit clinton
still one republican rate
good tax job
just trump time
can elect
even polici will sign care like
fact
much opinion say
take seem liber
make thing parti twitter
state newslett obama
see editori peopl
trade dont talk


democrat yet polici issu school
student world parti refuge public
part free invit donald dont presidenti seem
countri
join new nickkristof let
white unit war children debat
muslim famili that american day
jan help presid one percent
support editori say state even way
health kristof also trump nichola year get pleas
section follow
human just like clinton tri
make think mani now live
nation will watch call
twitter peopl video can want
much sign opinion today
poverti right time elect may
email women polit said
know america candid obama gun work
black republican facebook
govern column newslett
thenewyo campaign
univers


econom column
gop sign know american
secur live
ope section govern much
everi
putin countri get follow last first
said
tax make europ
york unit polit peopl say facebook
editori
hillari thoma just two donald
candid presid will democrat
nation friedman start call chang
day world
way elect right
think trump new also
polici trade social
mani leader twitter
global time can opinion war big human
climat come
desk want one parti now like even
good job dont
campaign america state republican
israel year
wall today clinton thing never
iraq need power
columnist obama presidenti
thenewyo work immigr newslett
technolog


around column thenewyo communiti
place ope percent dont democrat
desk york state twitter differ
seem
voter american make today thing becom
columnist use chang
even nation will opinion america
health live parti think book
care get life polit good
put year can one see
group newslett big unit
candid
work editori trump day
just mani take less way first
need love
govern peopl new presidenti feel immigr
realli want social elect
cultur now
much come time like republican
system section world countri
person also
econom donald david facebook
say polici everi
problem brook sign campaign
follow
right clinton moral moment


good
democrat
help never melania andré
iraq war elect dont year confirm
day
facebook state presidenti
night use republican even white man
run hes women time twitter look realli
note back section
hous presid said new just editori
see york one hillari candid need
right think unit
talk polit trump make
doesnt work
got can opinion first last
dowd obama say will bill shes
vote follow like told put
seem way clinton ask take
tri
peopl get donald thing
mani
love know want call parti around
thenewyo sign
campaign now thiel
alway bush
made show washington
maureen
come today column

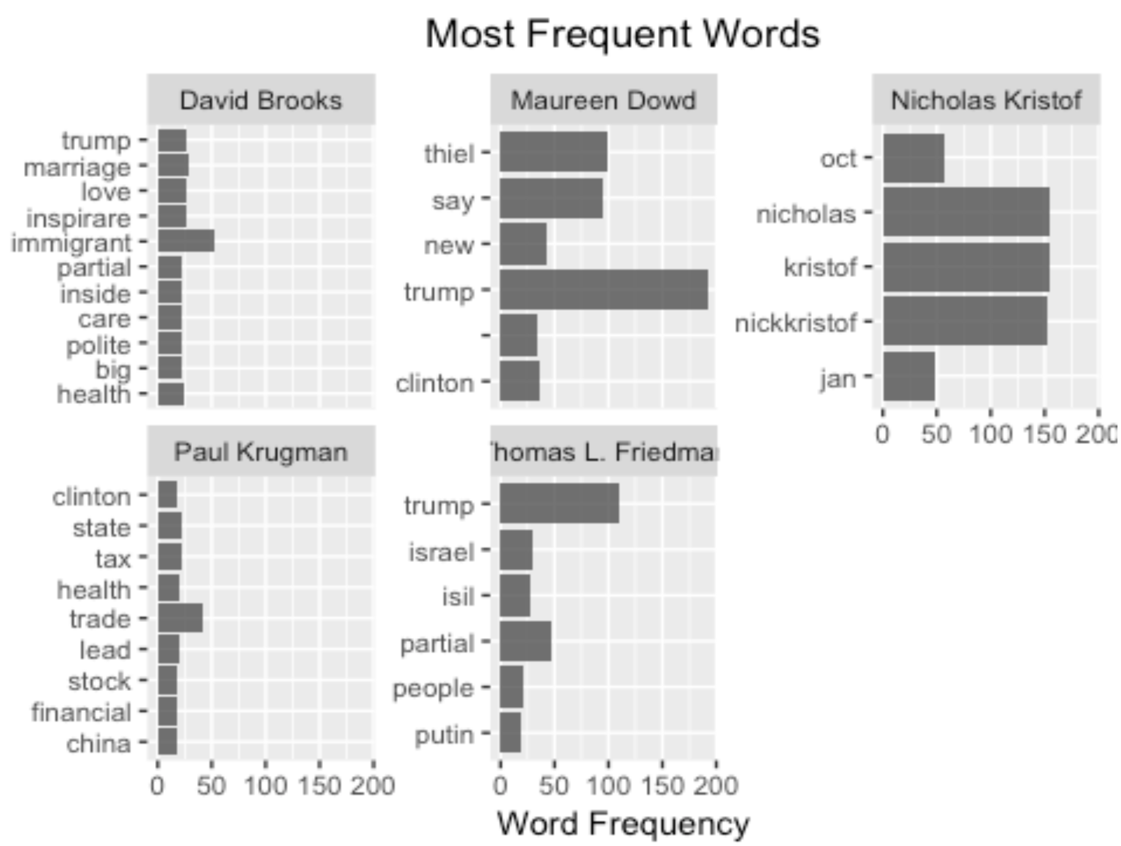
By using the metadata in the document , I want to explore the personal frequent words fur ther.

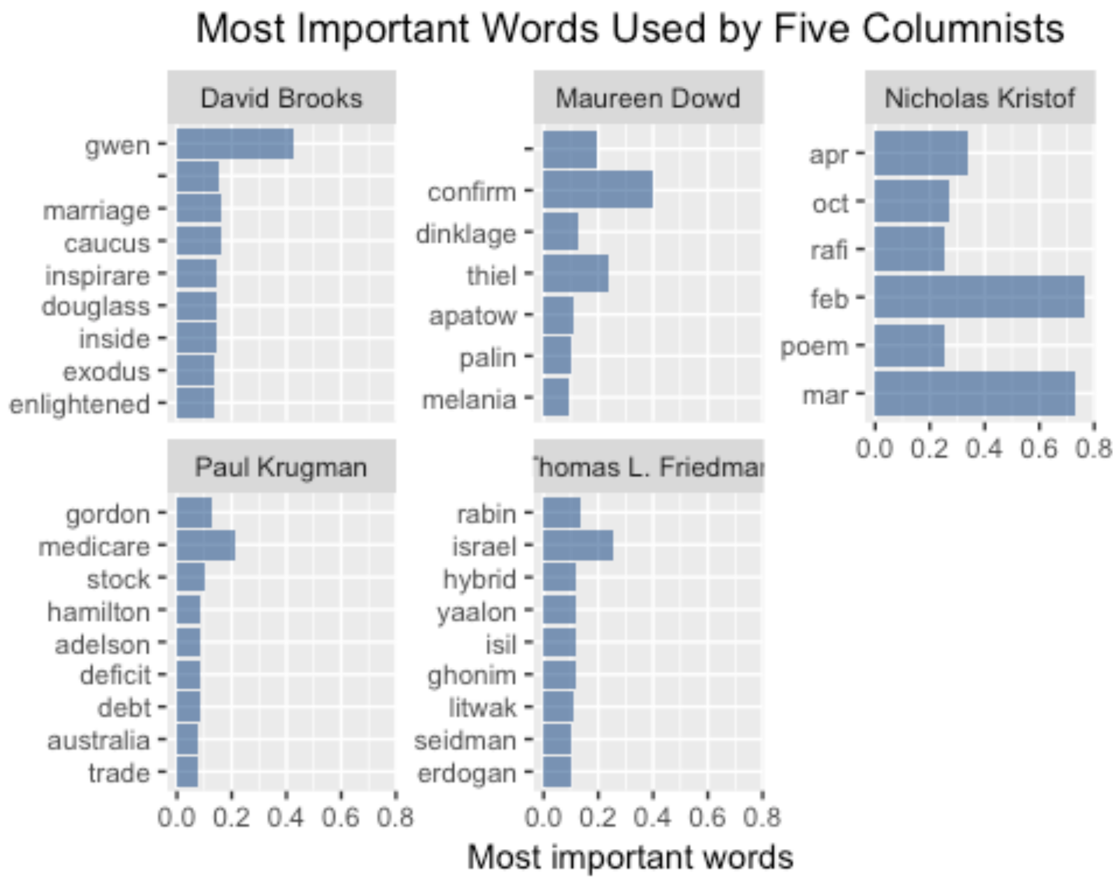
For the David Brooks, his most frequent word is immigrant. He focus on the domestic issues. He became an Op-Ed columnist for The New York Times in September 2003. Mr. Brooks also teaches at Yale University. He focus on how we live now in the future tense, so he put attention on the life issue such as immigration.

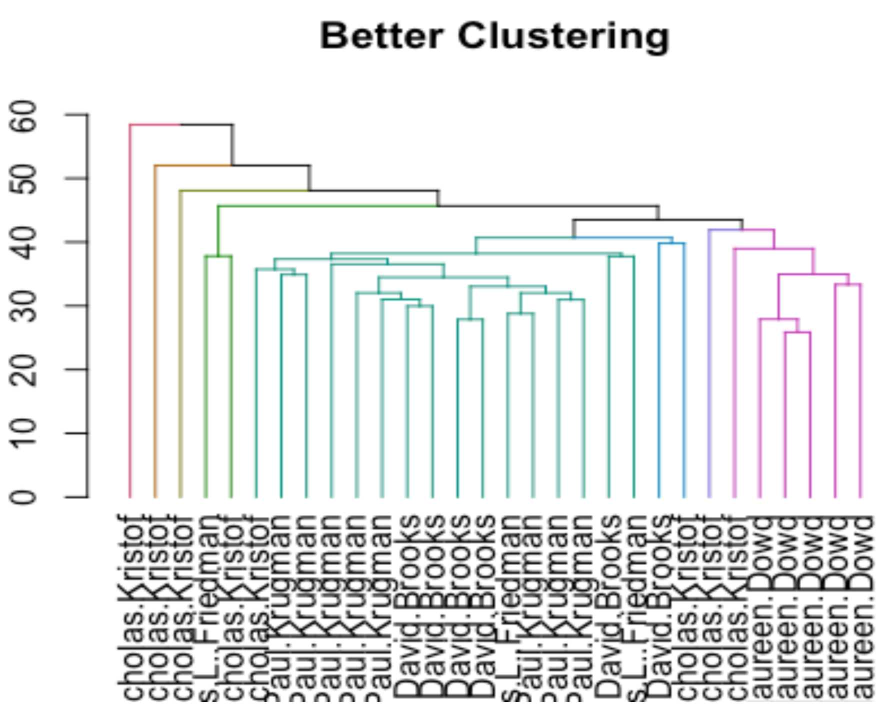
For Thomas L. Friedman, his most frequent word is Trump policy since he focus his articles on foreign affairs and globalization. Also, Mr. Friedman was awarded the 1983 Pulitzer Prize for international reporting.

For Nicholas Kristof, the most important word is feb, he focus on personal life in his articles.

So not all of them are focus on the national issues, some of them are interested in the rest of world. The diverse area of columnist give readers a better acknowledge to explore the world. They are all in difference in their writing style.



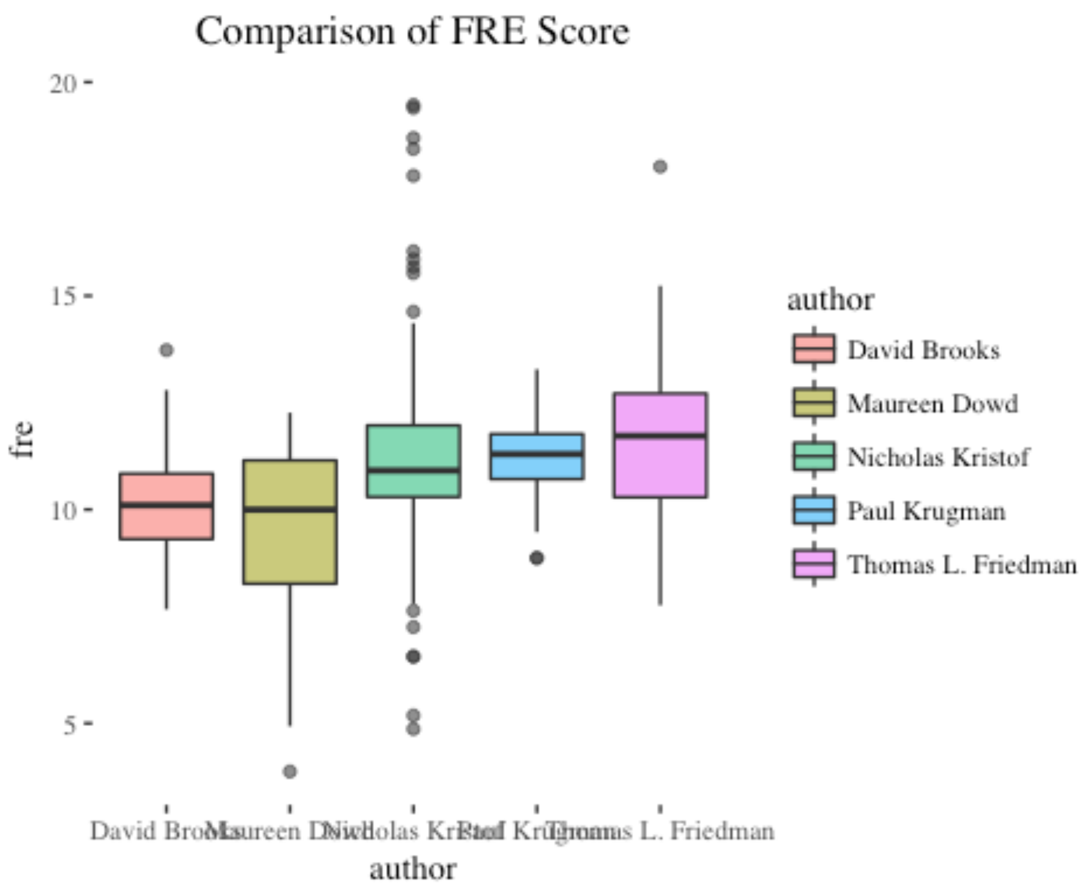

Most Important Words Used by Five Columnists
David Brooks
gwen
marriage
caucus
inspirare
douglass
inside
exodus
enlightened
Maureen Dowd
confirm
dinklage
thiel
apatow
palin
melania
Nicholas Kristof
apr
oct
rafi
feb
poem
mar
Paul Krugman
gordon
medicare
stock
hamilton
adelson
deficit
debt
australia
trade
homas L. Friedma
rabin
israel
hybrid
yaalon
isil
ghonim
litwak
seidman
erdogan
0.0 0.2 0.4 0.6 0.8
Most important words

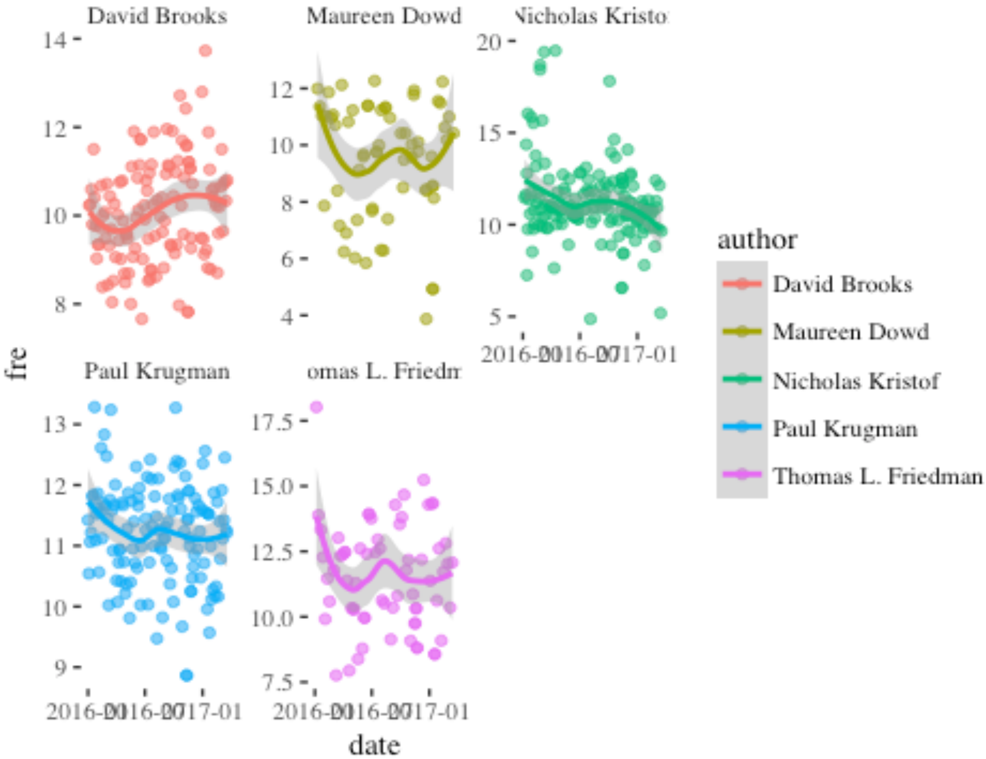

Better Clustering
0 10 20 30 40 50 60
cholas.Kristof
cholas.Kristof
cholas.Kristof
s.L..Friedman
cholas.Kristof
cholas.Kristof
Paul.Krugman
Paul.Krugman
Paul.Krugman
Paul.Krugman
Paul.Krugman
David.Brooks
David.Brooks
David.Brooks
David.Brooks
s.L..Friedman
Paul.Krugman
Paul.Krugman
Paul.Krugman
David.Brooks
s.L..Friedman
David.Brooks
cholas.Kristof
cholas.Kristof
cholas.Kristof
aureen.Dowd
aureen.Dowd
aureen.Dowd
aureen.Dowd
aureen.Dowd

2. Lingustic Complexity of NY Times Columns

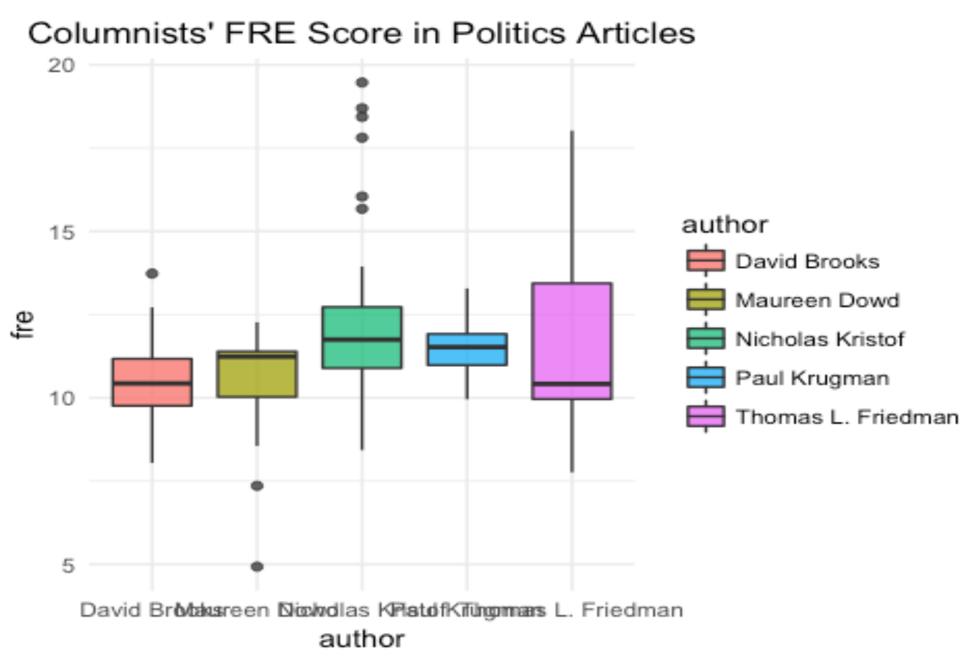
I find that the columnists differ in the complexity of their writing. The complexity is estimated by Flesch-Kincaid Reading Ease score. By calculating the FRE, the graph shows that Thomas has the highest score among those five columnist. Higher scores indicate material that is easier to read; It means that Thomas's articles are easy for readers to read. The Nicholas has the largest range of FRE score. It means that Nicholas's articles contains a lot of word complexity than others.

Then we could see the change with time of these five columnist. The graphs shows that the trend of Nicholas and Pual are declining. The trend of Maureen has large volatility. The trend of David is increasing. I think the change of trends are related to their topic and writing style.

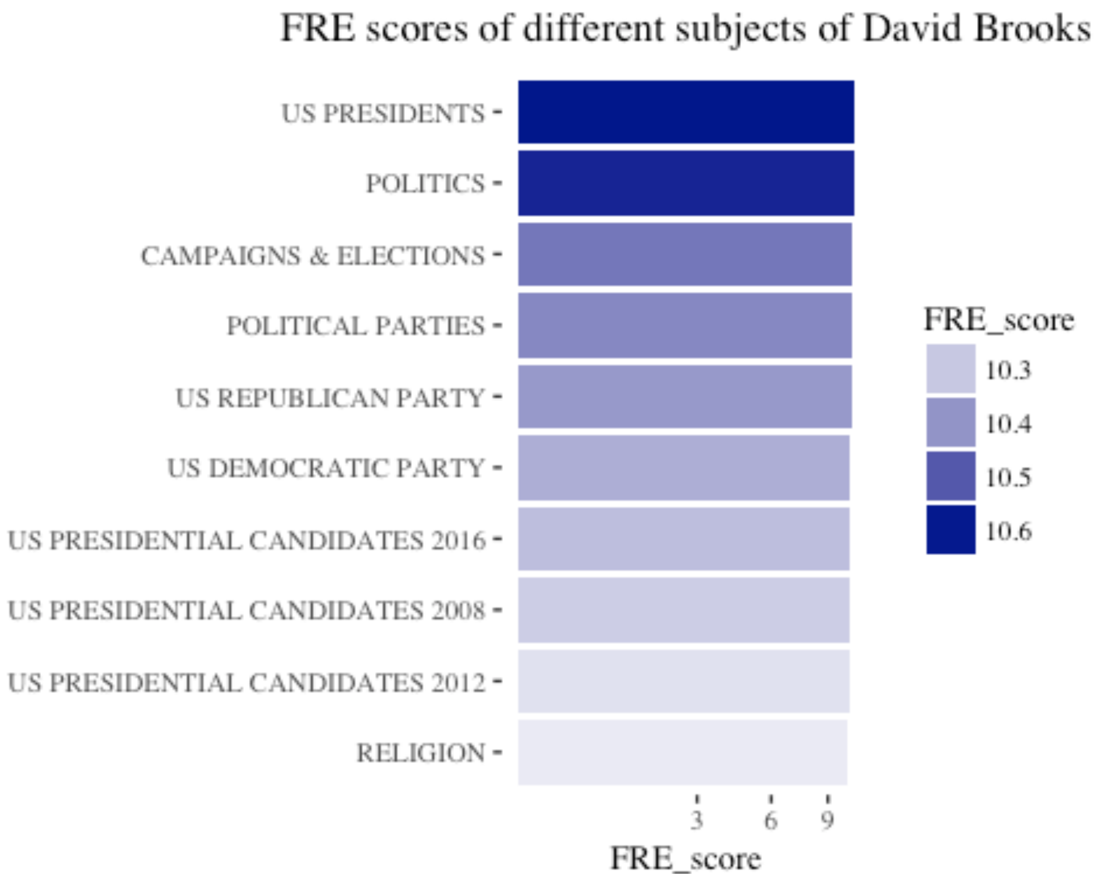




The graph of politics articles shows their words complexity in the political articles. I could see that Thomas has the highest FRE scores among others, it means that his articles are easy to understand. And he also has the largest range of FRE scores, since his articles are diverse and have word complexity.



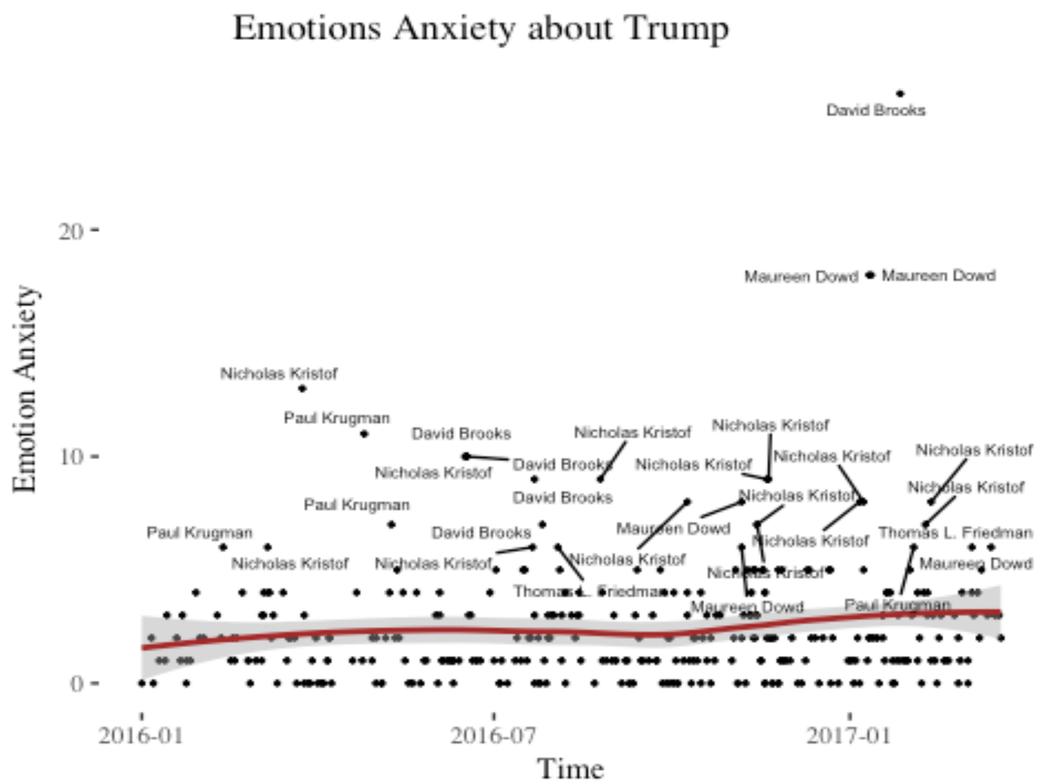
I want to make explore the subject of columnist further. David has his highest score for US president. The seond high score is politics. Since David recently focus on the the Republican Health Care Crackup, and his topics are political sicence. I could figure out from the graph that David has a lower FRE score. And he is different from others. To make further exploration, I find that his topics are policy culture.

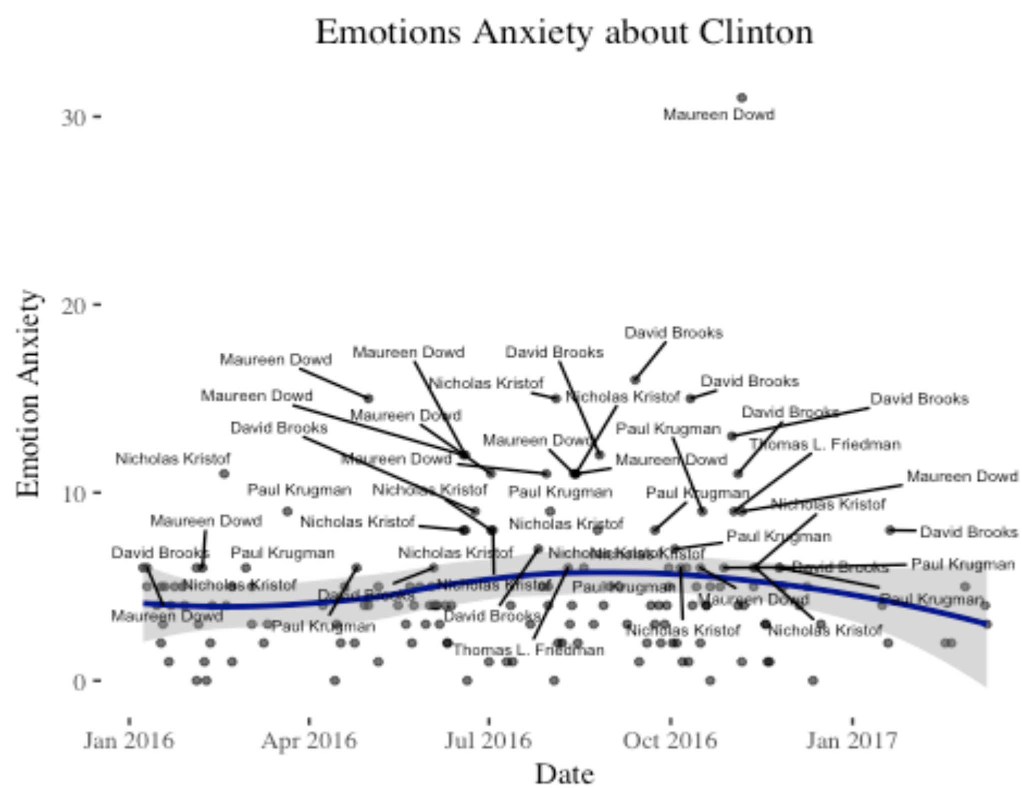


3. Sentiment Analysis

I identify the words the columnists that are associated with the candidates. By using the positive and negatinve words in the data, I find the relationship among those words and columnists. And I analyze how the tone of the texts differs between these two candidates and across the columnists, and describe the patterns.

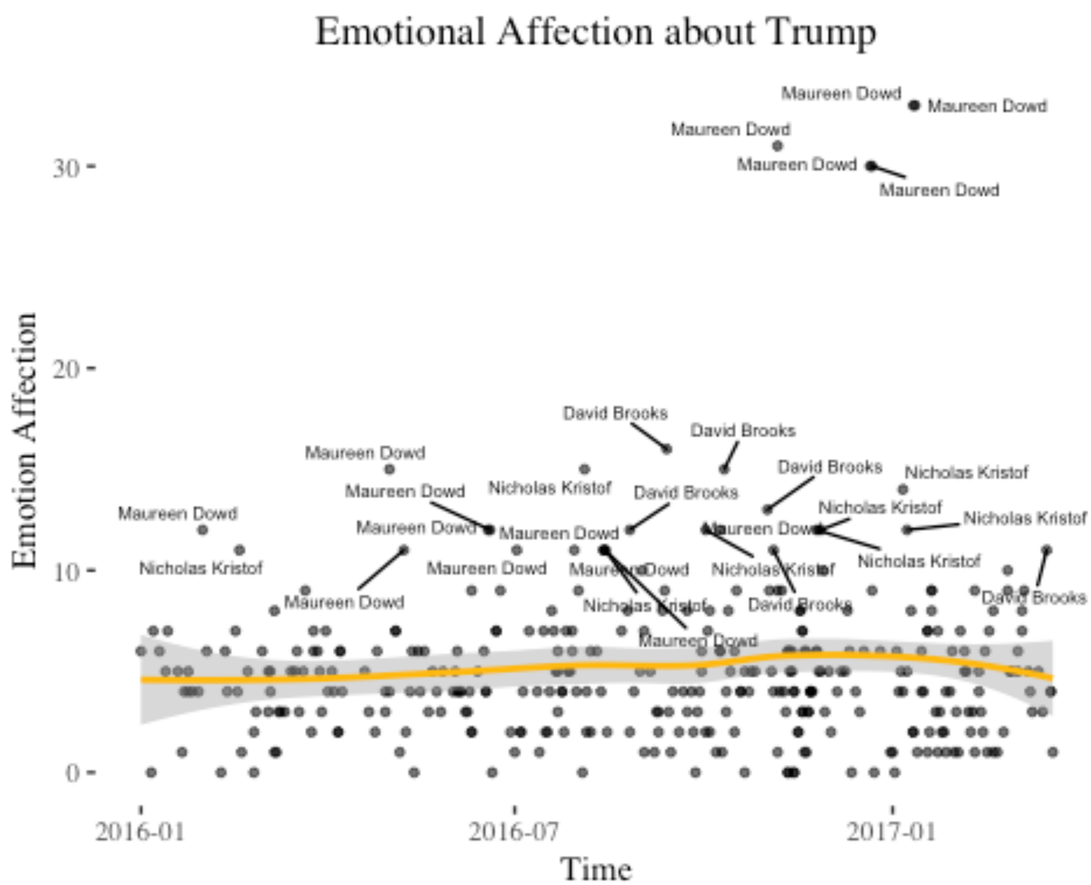
For negative words about Trump, the trend is increasing from Jan, 2016 to Jan, 2017. The David Brooks has the highest emotional anxiety. For negative words about Clinton, the trend is decreasing since Clinton didn't success in the presidential election.

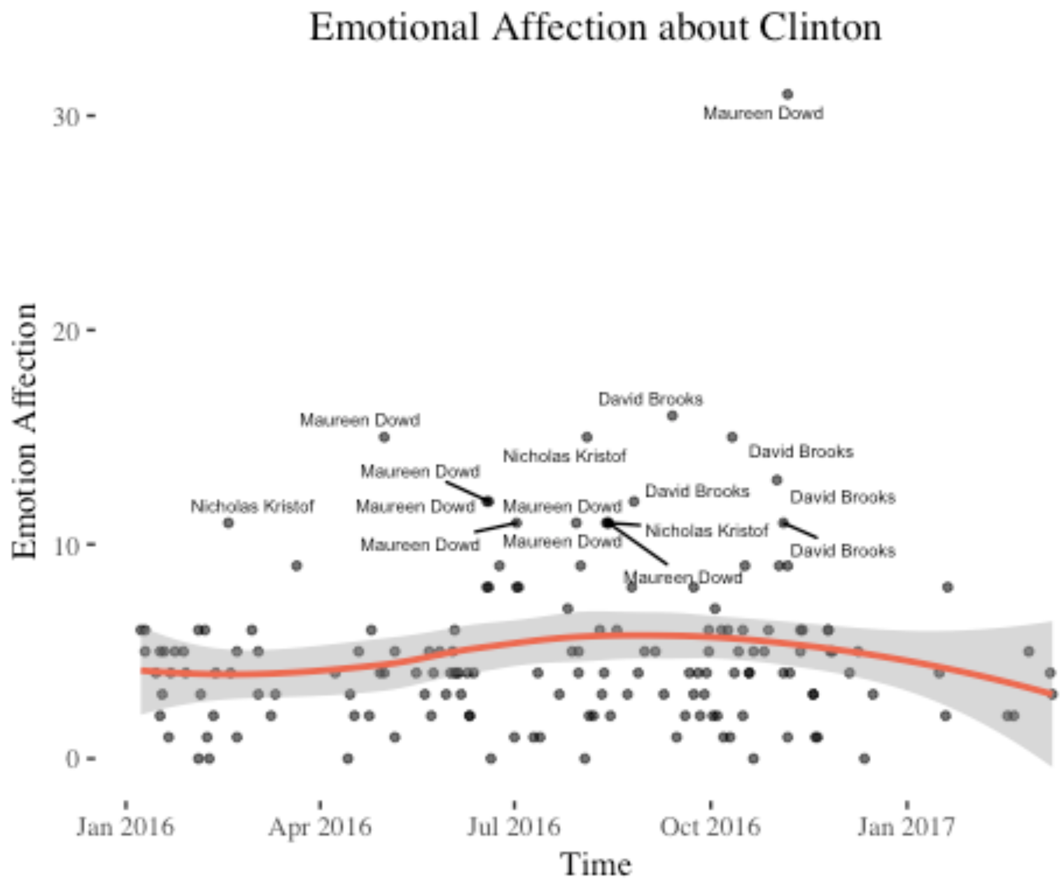

Emotions Anxiety about Trump
Emotion Anxiety
0
10
20
2016-01
2016-07
2017-01
Time


Emotions Anxiety about Clinton
Emotion Anxiety
0
10
20
30
Jan 2016
Apr 2016
Jul 2016
Oct 2016
Jan 2017
Date

As for positive words, the trend of Trump shows doesn't shows a large volatility, however, the trend of Hilary is decreasing. Since Trump policy are raise more and more attention, so some of columnist have higher emotion affection. Also, Maureen has the high attention on the emotion affection of Clinton. It appears that the columnist has their own tones and political choice. Maureen support Trump for a while. But others are on the average level about emotion affection for Trump. So five columnists have their own tones and make the NYT more diverse in the political sight.

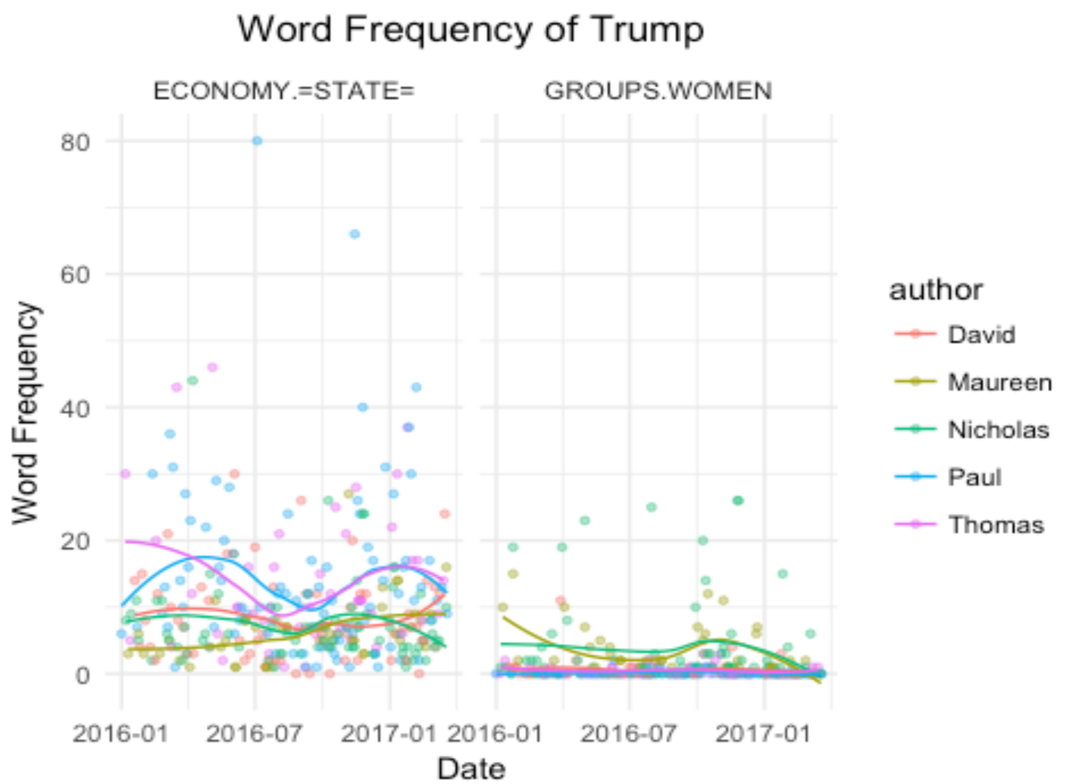
There are some major political events influenced the sentiment, the order of Trump issues visa and refugee restriction raise the anxiety among the five columnist in the 2017. The sentiment of anxiety are raising and Nicholas's artilcles show the anxiety and didn't support Trump policy.

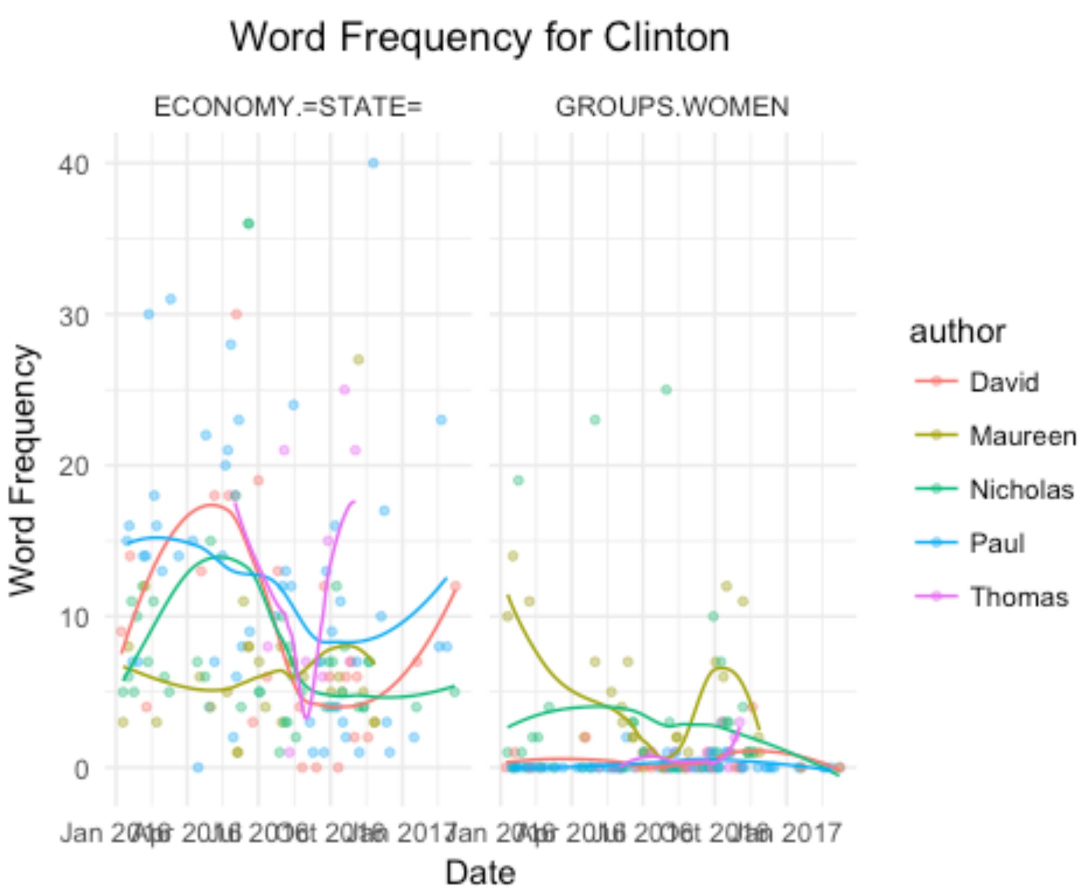




Then I analyze the word frequency of Trump in five columnists articles. For the topics of economu state and group women, the columnists shows difference in their articles. The word frequency of Trump about the group women doesn't have much volatility from Jan, 2016 to Jan, 2017. The word frequency of Trump about economy state shows large volatily. Especially, Paul has the highest word frequency to focus on the Trump economy state.

For the word frequency of Clinton, the group women has larger volatility than Trump, since Clinton support women's right and opportunity. And five columnist are interested in talking about these topic. The word frequency of economy state of Clinton also has larger volatility than Trump, since Clinton has many policy toward the middle class and friendly policy. Especially, Hillary believes our economy depends on a strong middle class and that means we have to raise incomes for hardworking Americans. These plan are attractive and five columnst want to talk about the economy state.


Word Frequency of Trump
ECONOMY.=STATE=
GROUPS.WOMEN
Word Frequency
0
20
40
60
80
2016-01
2016-07
2017-01
Date
author
David
Maureen
Nicholas
Paul
Thomas


Word Frequency for Clinton
ECONOMY.=STATE=
GROUPS.WOMEN
Word Frequency
0
10
20
30
40
Date
author
David
Maureen
Nicholas
Paul
Thomas

So It is likely helpful to identify some major political events, we could find many interesting topic in the graph and text mining about those five columnists in NYT and catch the trend of current political environment.

Project book:

1. Comparing NY Times Columnists.

I start with tokenizing the corpus, removing stop words, stemming and cleaning the data. I also create frequency matrices to explore the text as data.

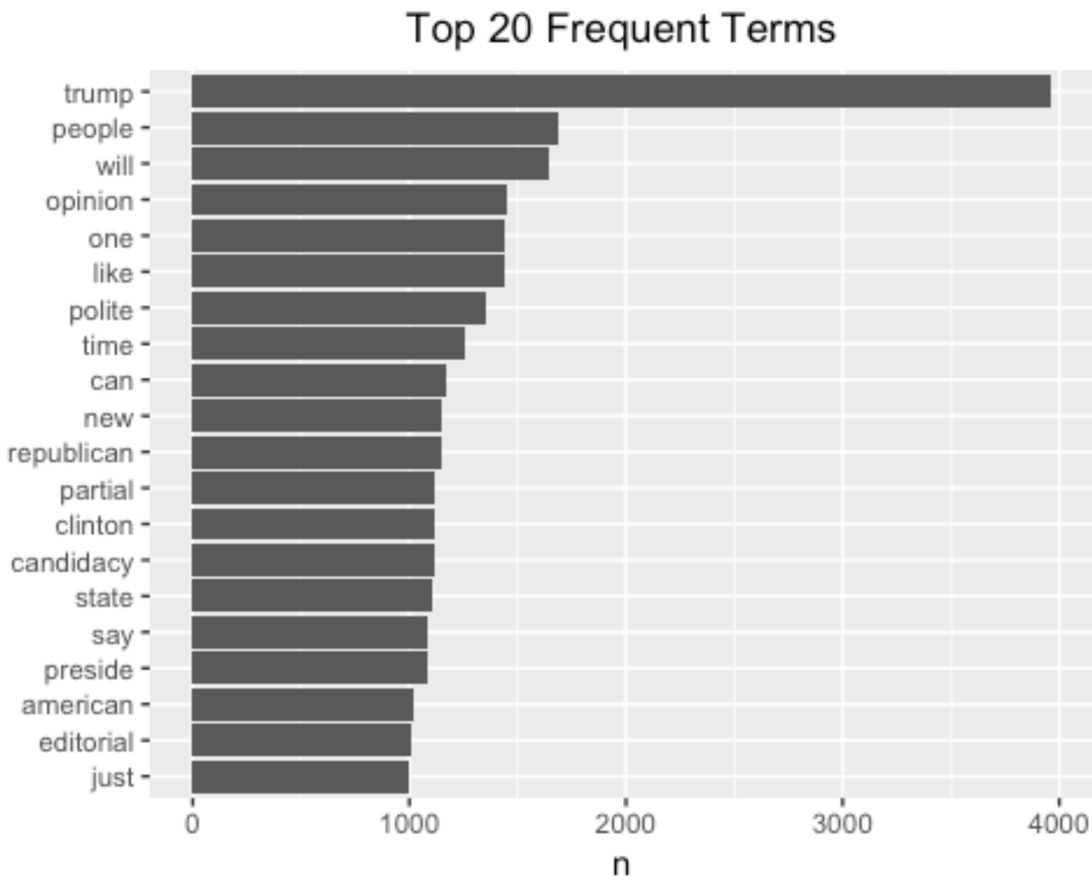
(1) I convert the corpus at first by loading tm corpus and metadata.Then clean the corpus.

(2) Removing stop words, stemming and clean the corpus.

(3) Adding metadata to tidy data.
(4) find most frequent terms

```
library(ggrepel)
library(RColorBrewer)
load("nytimes_oped_corpus.rda")
df <- corpus$documents
source <- DataframeSource(df)
corp <- VCorpus(source)
meta(corp, type="local", tag = "author") <- df$author
meta(corp, type="local", tag = "subject") <- df$subject
clean_corpus <- function(corpus){
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, content_transformer(replace_symbol))
  corpus <- tm_map(corpus, removeWords, c(stopwords("english")))
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  return(corpus)
}

corp_clean <- clean_corpus(corp)
corp_stemmed <- tm_map(corp_clean,stemDocument)
corp_tdm <- TermDocumentMatrix(corp_stemmed)
corp_t <- as.matrix(corp_tdm)
dim(corp_t)

## [1] 15660   547

corp_dtm <- DocumentTermMatrix(corp_stemmed)
corp_d <- as.matrix(corp_dtm)
dim(corp_d)

## [1]   547 15660
```

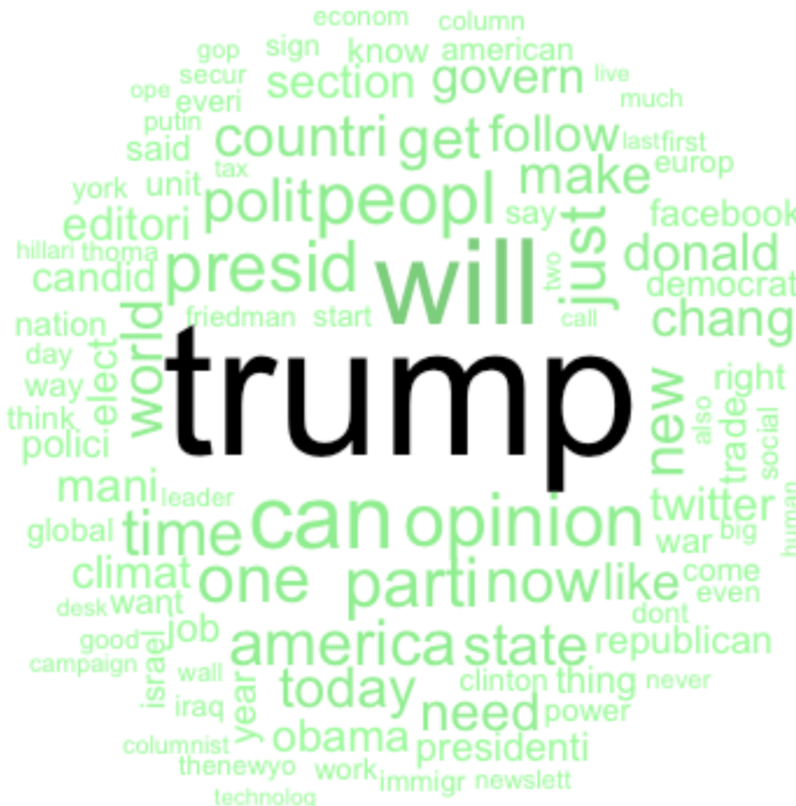
```r
author_p <- tm_filter(corp_stemmed, FUN = function(x) meta(x)[["author"]] ==
"Paul Krugman")
author_d <- tm_filter(corp_stemmed, FUN = function(x) meta(x)[["author"]] ==
"David Brooks")
author_t <- tm_filter(corp_stemmed, FUN = function(x) meta(x)[["author"]] ==
"Thomas L. Friedman")
author_m <- tm_filter(corp_stemmed, FUN = function(x) meta(x)[["author"]] ==
"Maureen Dowd")
author_n <- tm_filter(corp_stemmed, FUN = function(x) meta(x)[["author"]] ==
"Nicholas Kristof")
dtm_p <- tidy(DocumentTermMatrix(author_p))
dtm_p$author <- "Paul Krugman"

dtm_d <- tidy(DocumentTermMatrix(author_d))
dtm_d$author <- "David Brooks"

dtm_t <- tidy(DocumentTermMatrix(author_t))
dtm_t$author <- "Thomas L. Friedman"

dtm_m <- tidy(DocumentTermMatrix(author_m))
dtm_m$author <- "Maureen Dowd"

dtm_n <- tidy(DocumentTermMatrix(author_n))
dtm_n$author <- "Nicholas Kristof"
dtm_all <- rbind(dtm_p, dtm_d, dtm_t, dtm_m, dtm_n)
dtm_all_top <- group_by(dtm_all, term)%>%
              summarise(n = sum(count)) %>%
              top_n(n = 20, wt = n) %>%
              mutate(term = reorder(term, n))

# complete the term
dtm_all_top$term = stemCompletion(dtm_all_top$term, corp_clean)
freq_all <- dtm_all_top %>%
            mutate(term = reorder(term, n)) %>%
            ggplot(aes(term, n),color="bisque4") +
            geom_bar(stat = "identity") +
            xlab(NULL) +
            coord_flip()  +
            theme(plot.title = element_text(hjust = 0.5)) +
            ggtitle("Top 20 Frequent Terms")

freq_all
```



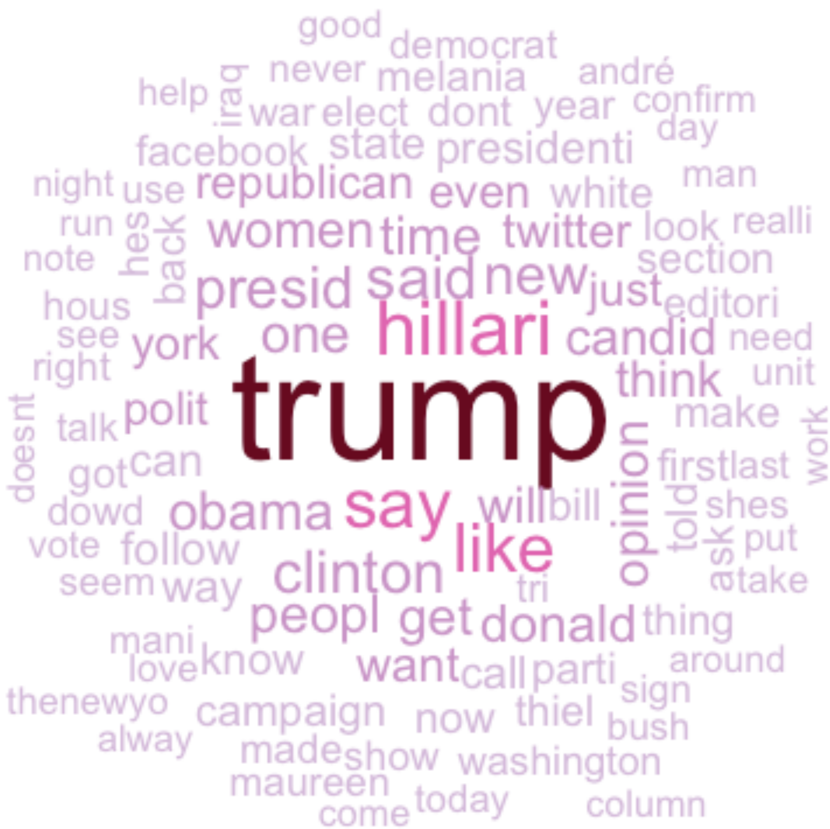

```
sel <- dtm_all[, c("term", "author", "count")]
sel_p <- filter(sel, author == c("Paul Krugman")) %>%
         group_by(term) %>%
        summarise(n = sum(count))
wordcloud(sel_p$term, sel_p$n,
                    random.order = FALSE,
                    random.color = FALSE,
                    colors= c("bisque1","bisque2","bisque3","black"),
                    scale=c(5, .1),
                    max.words= 100)
```


```
sel <- dtm_all[, c("term", "author", "count")]
sel_p <- filter(sel, author == c("David Brooks")) %>%
         group_by(term) %>%
        summarise(n = sum(count))

wordcloud(sel_p$term, sel_p$n,
                    random.order = FALSE,
                    random.color = FALSE,
                    colors= c("dodgerblue1","dodgerblue2","dodgerblue3","dodg
erblue4"),
                    scale=c(4, .2),
                    max.words= 100)
```


```
sel <- dtm_all[, c("term", "author", "count")]
sel_p <- filter(sel, author == c("Thomas L. Friedman")) %>%
         group_by(term) %>%
        summarise(n = sum(count))

wordcloud(sel_p$term, sel_p$n,
                    random.order = FALSE,
                    random.color = FALSE,
                    colors= c("palegreen1","palegreen2","palegreen3","black
"),
                    scale=c(5, .1),
                    max.words= 100)
```




```
sel <- dtm_all[, c("term", "author", "count")]
sel_p <- filter(sel, author == c("Maureen Dowd")) %>%
         group_by(term) %>%
        summarise(n = sum(count))
purple_orange <- brewer.pal(10, "PuRd")
purple_orange <- purple_orange[-(1:2)]

wordcloud(sel_p$term, sel_p$n,
                    random.order = FALSE,
                    random.color = FALSE,
                     colors =purple_orange,
                    scale=c(4, .8),
                    max.words= 100)
```

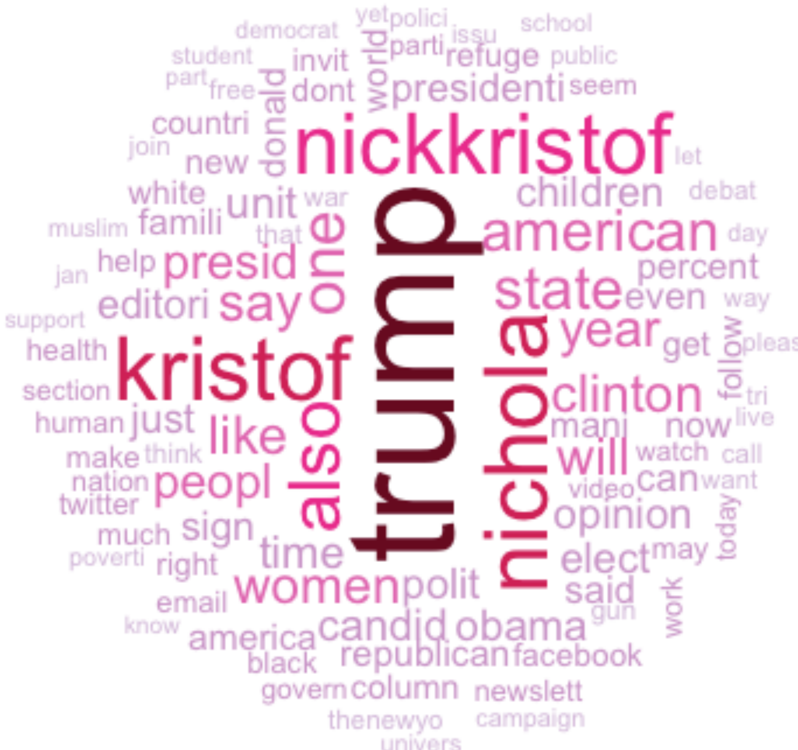


You can also embed plots, for example:

```
sel <- dtm_all[, c("term", "author", "count")]
sel_p <- filter(sel, author == c("Nicholas Kristof")) %>%
         group_by(term) %>%
        summarise(n = sum(count))
blue_orange <- brewer.pal(15, "RdBu")
```

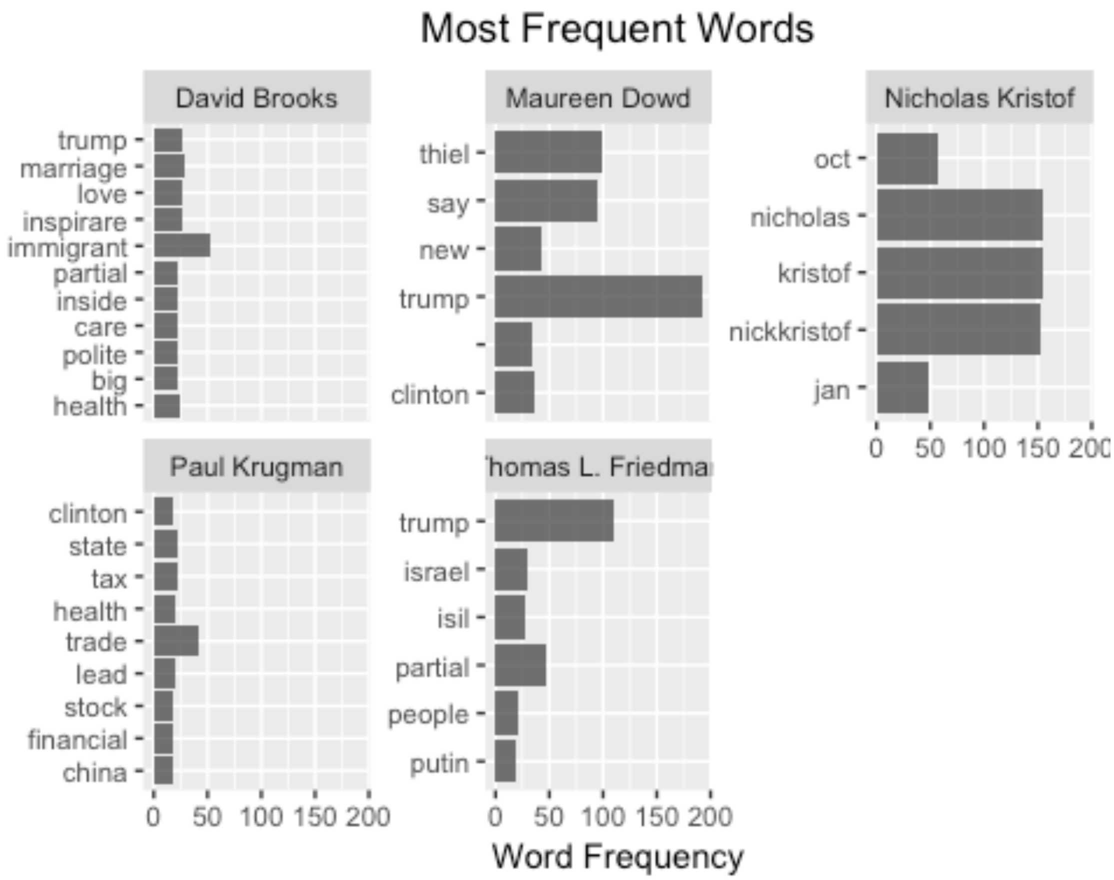
```
## Warning in brewer.pal(15, "RdBu"): n too large, allowed maximum for palett
e RdBu is 11
## Returning the palette you asked for with that many colors
```

```
blue_orange <- purple_orange[-(1:2)]
wordcloud(sel_p$term, sel_p$n,
                    random.order = FALSE,
                    random.color = FALSE,
                    colors = purple_orange,
                    scale=c(4, .2),
                    max.words= 100)
```



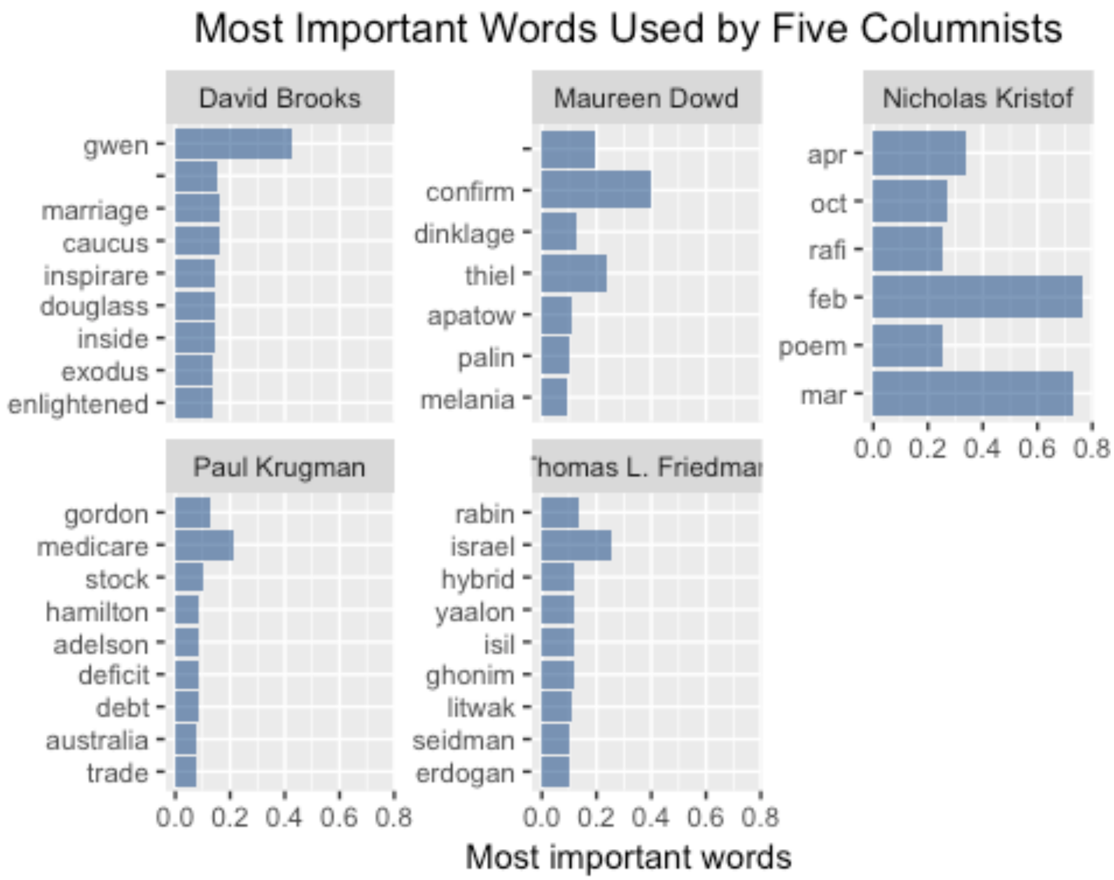
I plot some graphs which put the features of 5 columnists together and get the TF, df, and IDF frequency. I summarize the top 10 most important words unsed by each columnist and complete the term by using stemCompletion, then I plot the term as below to see the most important words.

```
all_tf_idf <-  dtm_all %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))
freq_by_author1 <- all_tf_idf %>%
                  group_by(author) %>%
                  top_n(n = 10, wt = count)
freq_by_author1$term = stemCompletion(freq_by_author1$term, corp_clean)
combine_plot1<-   freq_by_author1%>%
                  ggplot(aes(x = reorder(term, count), y = count)) +
                  geom_bar(stat = "identity", fill = "gray29",alpha=0.8) +
                  facet_wrap(~ author, scales="free_y") +
                  xlab(NULL) +
                  ylab("Word Frequency") +
                  coord_flip() +
                  ggtitle("Most Frequent Words") +
                  theme(plot.title = element_text(hjust = 0.5))
combine_plot1
```



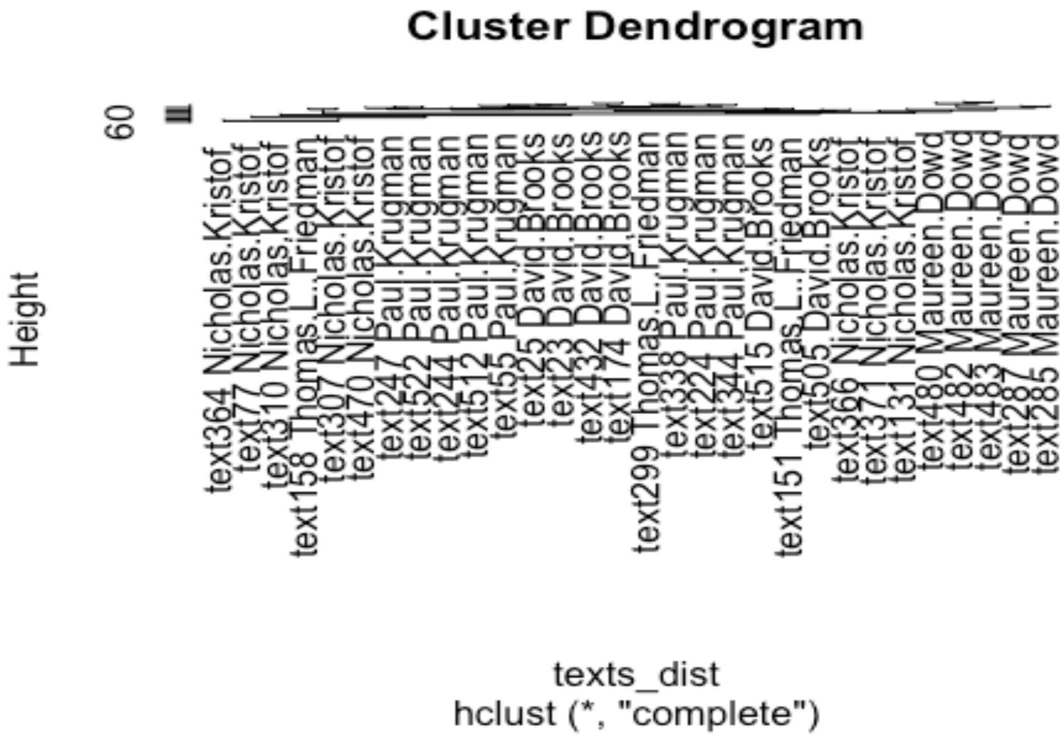
I

```
freq_by_author2 <-all_tf_idf %>%
                 group_by(author) %>%
                 top_n(n = 10, wt = tf_idf)
freq_by_author2$term = stemCompletion(freq_by_author2$term, corp_clean)
combine_plot2 <- freq_by_author2 %>%
                ggplot(aes(x = reorder(term, tf_idf), y = tf_idf)) +
                 geom_bar(stat = "identity",alpha=0.6,fill="dodgerblue4") +
                 coord_flip() +
                 facet_wrap(~ author, scales="free_y") +
                 xlab(NULL) +
                 ylab("Most important words") +
                 ggtitle("Most Important Words Used by Five Columnists") +
                 theme(plot.title = element_text(hjust = 0.5))
combine_plot2
```



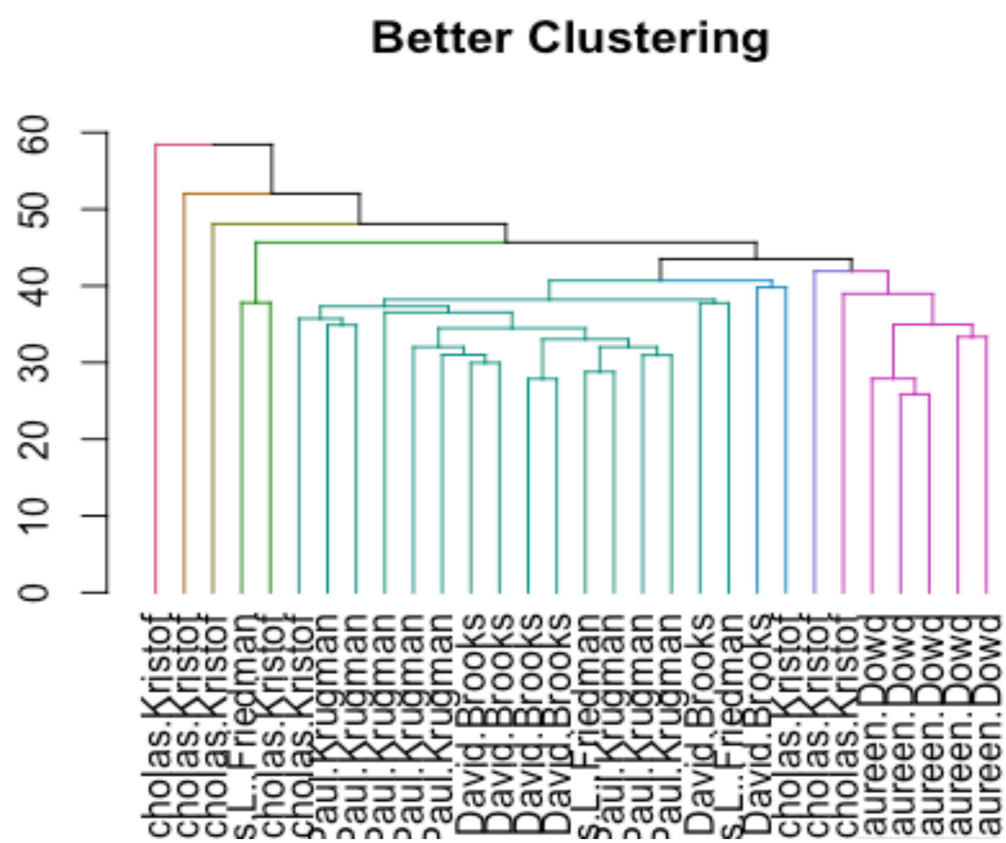
I start to make the dendrogram. First, I make a claer row name of dataframe. As the lecture done, I remove the sparse term by using removeSparseTerms and create dtm dataframe, texts_dist and hc.Then I plot the dendrogram.

```
df_h <- corpus$documents[, c("texts", "author", "subject", "date", "person")]
num <- as.character(row.names(df_h))
au_name <- make.names(df_h$author, unique = FALSE)
df_h$new_name <- paste0(num," ", au_name)
set.seed(123)
sel <- sample(1:547, 30, replace = FALSE)
myReader <- readTabular(mapping=list(content = "texts", id = "new_name"))
corp_h <- VCorpus(DataframeSource(df_h[sel,]), readerControl = list(reader =
myReader))
corp_h <- clean_corpus(corp_h)
corp_h <- tm_map(corp_h,stemDocument)
corp_h_dtm <- DocumentTermMatrix(corp_h)
dtm1 <- removeSparseTerms(corp_h_dtm, sparse = 0.9)
h_df <- as.data.frame(as.matrix(dtm1))
texts_dist <- dist(h_df)
hc <- hclust(texts_dist)
plot(hc)
```



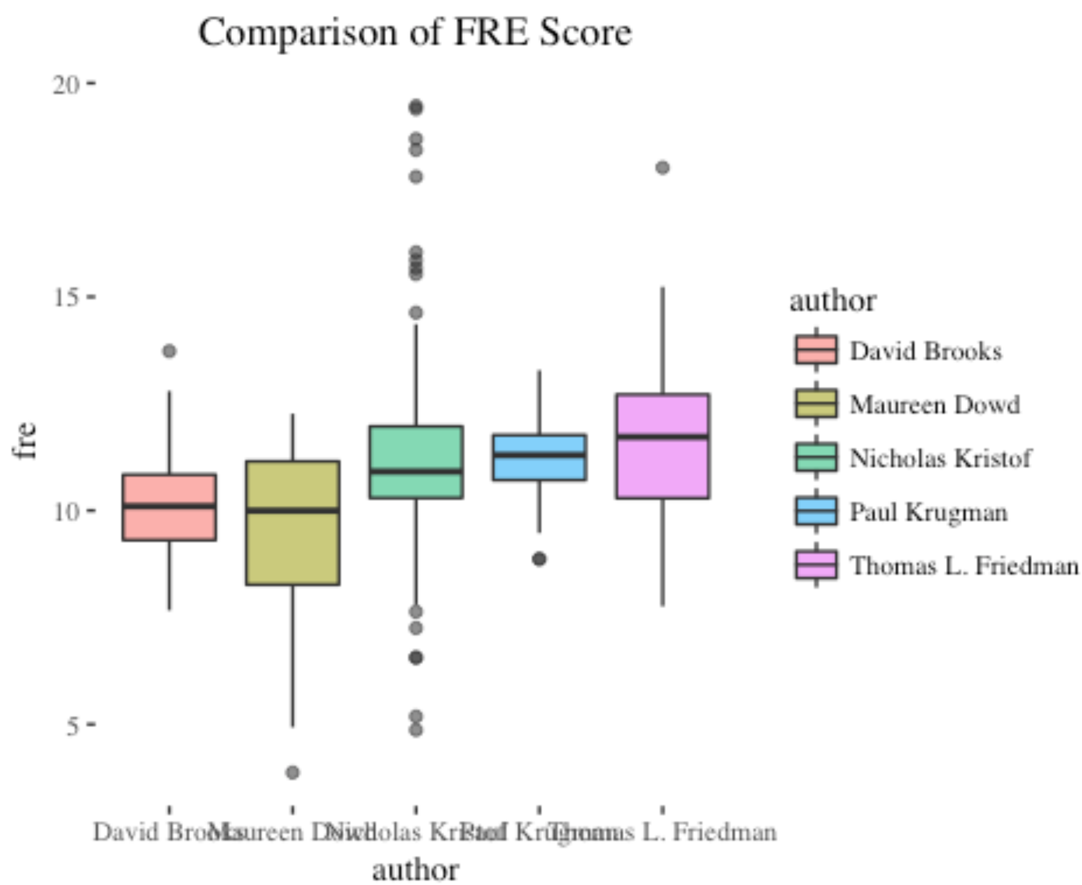
To make dengrogram pretty, I use dendextend package to operate on dendrogram objects. So I change the hierarchical cluster from hclust using as.dendrogram.

```
hc <- hclust(texts_dist)
dend <- as.dendrogram(hc)
# divide into 8 categories
dend %>% set("branches_k_color", k = 8) %>% plot( main = "Better Clustering",
cex = 0.7)
```



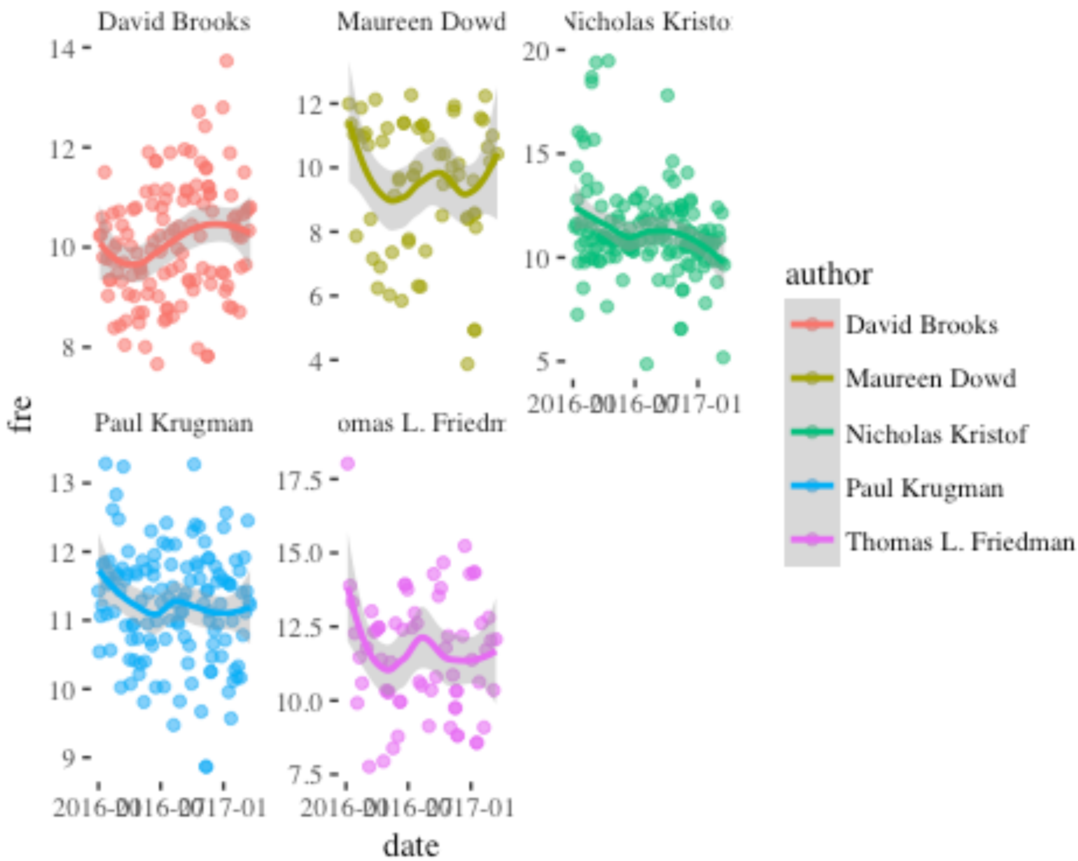
Problem 2.I conver to the quanteda corpus and calculate the Fresch-Kincaid Reading Ease Score. Then I plot with added information.

```
corp_1 <- corpus(df, text_field = "texts")
FRE_all <- textstat_readability(corp_1,
              measure=c('Flesch.Kincaid'))
df$fre <- FRE_all
FRE2 <- df[, c("author", "date", "fre", "subject")]
FRE2$date <- as.Date(FRE2$date)
FRE_box <- ggplot(data = FRE2) +
            geom_boxplot(aes(x = author, y = fre, fill= author), alpha = 0.6)
 +
            theme_tufte() +
            ggtitle("Comparison of FRE Score") +
            theme(plot.title = element_text(hjust = 0.5))

FRE_box
```




```
FRE_plot <- ggplot(data = FRE2) +
            geom_point(aes(x = date, y = fre, color = author), alpha = 0.6) +
            geom_smooth(aes(x = date, y = fre, color = author)) +
            facet_wrap(~ author, scales="free_y") +
            theme_tufte()
FRE_plot
```

```
## `geom_smooth()` using method = 'loess'
```



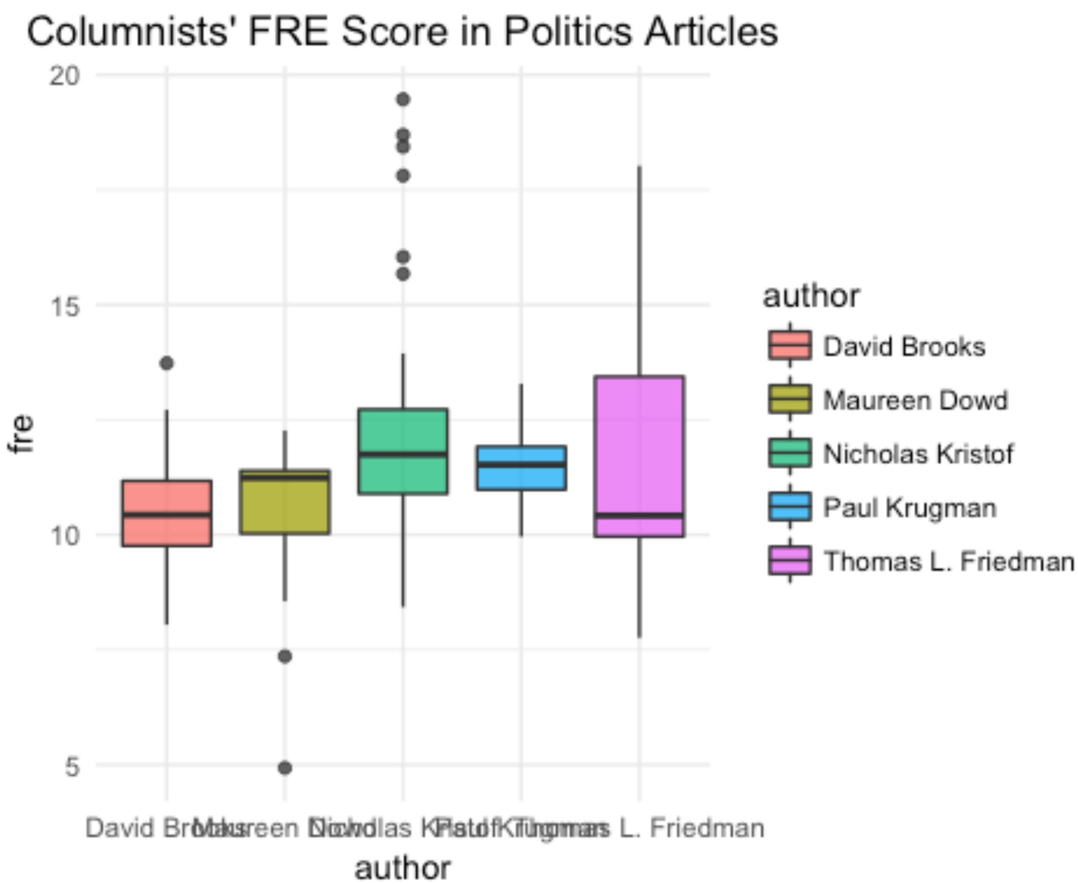

```r
subjects <- gsub( " *\\(.*?\\) *", "", df$subject) # Remove parentheses
subjects <- strsplit(subjects, ";") # Split by ';' into a list
subjects <- lapply(subjects, FUN=trimws) # Remove whitespace
subjectlist <- unique(unlist(subjects)) # Make into a list, remove whitespace
top10subjects <- rownames(sort(table(unlist(subjects)),
decreasing=TRUE)[2:11])

subjects <- lapply(subjects, FUN=
function(A,B){top10subjects[match(A,top10subjects)]})
subjects <- lapply(subjects, function(x) x[!is.na(x)])
top10subjects
```

```
##  [1] "US PRESIDENTIAL CANDIDATES 2016" "US PRESIDENTIAL CANDIDATES 2012"
##  [3] "US REPUBLICAN PARTY"             "POLITICS"
##  [5] "POLITICAL PARTIES"               "US PRESIDENTS"
##  [7] "CAMPAIGNS & ELECTIONS"           "RELIGION"
##  [9] "US DEMOCRATIC PARTY"             "US PRESIDENTIAL CANDIDATES 2008"
```

```r
x <- 1:547
 for(i in x){
        if(c("US PRESIDENTIAL CANDIDATES 2016") %in% c(subjects[[i]]) == TRU
E)
                         {FRE2[i, "US PRESIDENTIAL CANDIDATES 2016"] <- "YES"
                 } else {
                   FRE2[i, "US PRESIDENTIAL CANDIDATES 2016"] <- "NO"
```

```
                }
        if(c("US PRESIDENTIAL CANDIDATES 2012") %in% c(subjects[[i]]) == TRU
E)
                        {FRE2[i, "US PRESIDENTIAL CANDIDATES 2012"] <- "YES"
                } else {
                  FRE2[i, "US PRESIDENTIAL CANDIDATES 2012"] <- "NO"
                }
        if(c("US REPUBLICAN PARTY") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "US REPUBLICAN PARTY"] <- "YES"
                } else {
                  FRE2[i, "US REPUBLICAN PARTY"] <- "NO"
                }
        if(c("POLITICS") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "POLITICS"] <- "YES"
                } else {
                  FRE2[i, "POLITICS"] <- "NO"
                }
        if(c("POLITICAL PARTIES") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "POLITICAL PARTIES"] <- "YES"
                } else {
                  FRE2[i, "POLITICAL PARTIES"] <- "NO"
                }
        if(c("CAMPAIGNS & ELECTIONS") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "CAMPAIGNS & ELECTIONS"] <- "YES"
                } else {
                  FRE2[i, "CAMPAIGNS & ELECTIONS"] <- "NO"
                }
        if(c("RELIGION") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "RELIGION"] <- "YES"
                } else {
                  FRE2[i, "RELIGION"] <- "NO"
                }
        if(c("US DEMOCRATIC PARTY") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "US DEMOCRATIC PARTY"] <- "YES"
                } else {
                  FRE2[i, "US DEMOCRATIC PARTY"] <- "NO"
                }
        if(c("US PRESIDENTIAL CANDIDATES 2008") %in% c(subjects[[i]]) == TRU
E)
                        {FRE2[i, "US PRESIDENTIAL CANDIDATES 2008"] <- "YES"
                } else {
                  FRE2[i, "US PRESIDENTIAL CANDIDATES 2008"] <- "NO"
                }
        if(c("US PRESIDENTS") %in% c(subjects[[i]]) == TRUE)
                        {FRE2[i, "US PRESIDENTS"] <- "YES"
                } else {
                  FRE2[i, "US PRESIDENTS"] <- "NO"
                }
 }
politic_fre <- filter(FRE2, `POLITICS` == "YES") %>%
```

```
                ggplot() +
                geom_boxplot(aes(x = author, y = fre, fill=author),alpha=0.8)
+
                theme_minimal() +
                ggtitle("Columnists' FRE Score in Politics Articles") +
                theme(plot.title = element_text(hjust = 0.5))
politic_fre
```



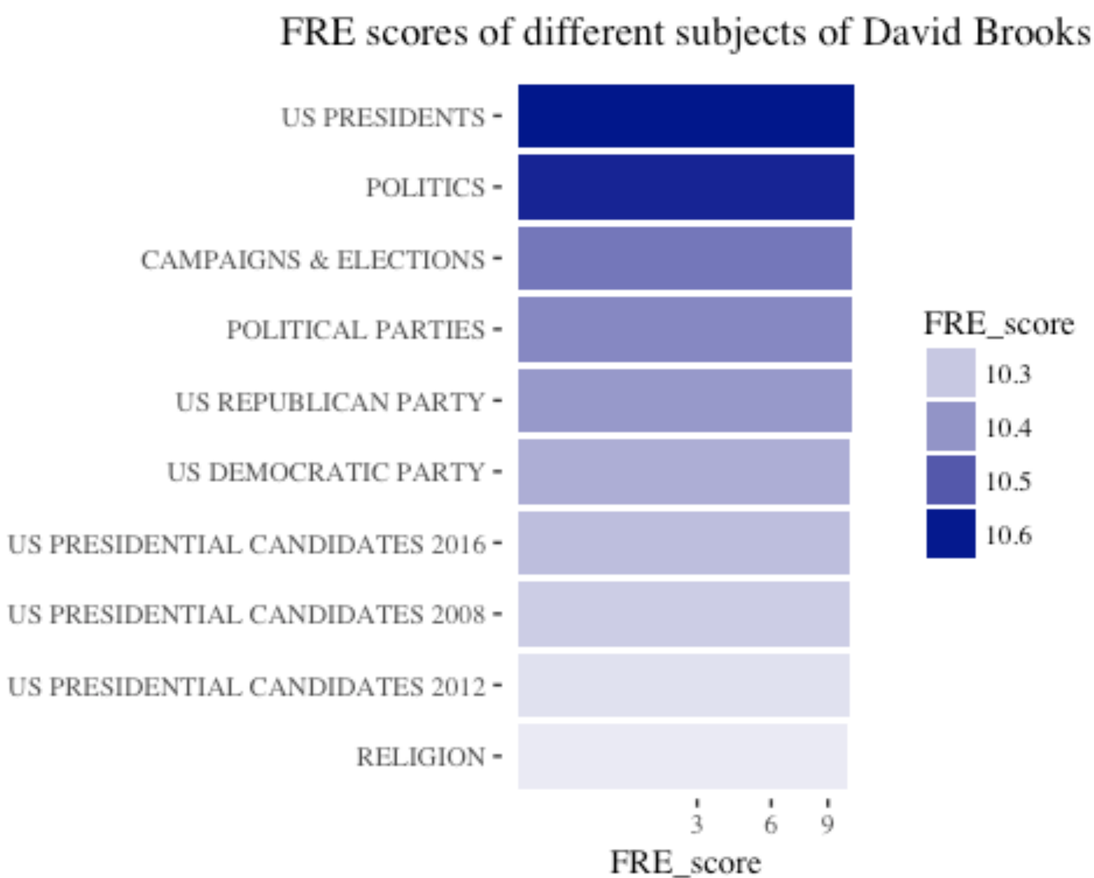
Then I use the mean FRE score of David Brooks to show the word complexity.

```
df_d <- filter(FRE2, author == "David Brooks")
fre_sub1 <- df_d%>%
         filter(`US PRESIDENTIAL CANDIDATES 2016` == "YES") %>%
         summarise(ave1 = mean(fre))
fre_sub2 <- df_d %>%
         filter(`US PRESIDENTIAL CANDIDATES 2012` == "YES") %>%
         summarise(ave2 = mean(fre))
fre_sub3 <- df_d %>%
         filter(`US REPUBLICAN PARTY` == "YES") %>%
         summarise(ave3 = mean(fre))
fre_sub4 <- df_d %>%
         filter(`POLITICS` == "YES") %>%
         summarise(ave4 = mean(fre))
fre_sub5 <- df_d %>%
```

```r
         filter(`POLITICAL PARTIES` == "YES") %>%
         summarise(ave5 = mean(fre))
fre_sub6 <- df_d %>%
         filter(`US PRESIDENTS` == "YES") %>%
         summarise(ave6 = mean(fre))
fre_sub7 <- df_d %>%
         filter(`CAMPAIGNS & ELECTIONS` == "YES") %>%
         summarise(ave7 = mean(fre))
fre_sub8 <- df_d %>%
         filter(`RELIGION` == "YES") %>%
         summarise(ave8 = mean(fre))
fre_sub9 <- df_d %>%
         filter(`US DEMOCRATIC PARTY` == "YES") %>%
         summarise(ave9 = mean(fre))
fre_sub10 <- df_d %>%
         filter(`US PRESIDENTIAL CANDIDATES 2008` == "YES") %>%
         summarise(ave10 = mean(fre))
a <- as.character(c("US PRESIDENTIAL CANDIDATES 2016", "US PRESIDENTIAL
CANDIDATES 2012",
       "US REPUBLICAN PARTY", "POLITICS", "POLITICAL PARTIES",
       "US PRESIDENTS", "CAMPAIGNS & ELECTIONS", "RELIGION",
       "US DEMOCRATIC PARTY", "US PRESIDENTIAL CANDIDATES 2008"))
b <- c(fre_sub1, fre_sub2, fre_sub3, fre_sub4, fre_sub5,
       fre_sub6, fre_sub7, fre_sub8, fre_sub9, fre_sub10)
b <- round(as.numeric(b), 2)
sub_d <- as.data.frame(cbind(a,b))
colnames(sub_d) <- c("subject", "FRE_score")
sub_d$FRE_score <- as.numeric(as.character(sub_d$FRE_score))
sub_d_plot <- ggplot(sub_d) +
        geom_histogram(aes(x = reorder(subject,FRE_score) ,
                           y = FRE_score, alpha = FRE_score),
                       stat = "identity", fill = "blue4") +
        coord_flip()+
        scale_y_sqrt() +
        theme_tufte() +
        xlab(NULL) +
        ggtitle("FRE scores of different subjects of David Brooks") +
        theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```r
sub_d_plot
```



3.I identify the words the columnists that are associated with the candidates.And then I analyze how the tone of the texts differs between these two candidates and across the columnists, and describe the patterns.I check whether major political events influenced the sentiment of how columnists wrote about the two candidates. In details, I load RID dictionary to meaure emotions.Then I make DFM into data frame to plot with ggplot.

```
df$trump_article <- grepl("TRUMP", df$person, fixed=TRUE)
df$clinton_article <- grepl("CLINTON", df$person, fixed=TRUE)
df_tr <- filter(df, trump_article == TRUE)
df_cl <- filter(df, clinton_article == TRUE)
corp_tr <- corpus(df_tr,text_field = "texts")
corp_cl <- corpus(df_cl, text_field = "texts")
cname <- file.path("~", "Desktop", "dictionaries","positive-words.txt")
dname <-  file.path("~", "Desktop", "dictionaries","negative-words.txt")
pos <- read.table(cname, as.is=T)
neg <- read.table(dname, as.is=T)
myDict <- dictionary(list(positive = pos, negative = neg))
sentiment <- function(words=c("really great good stuff bad")){
  require(quanteda)
  tok <- quanteda::tokenize(words)
  pos.count <- sum(tok[[1]]%in%pos[,1])
  cat("\n positive words:",tok[[1]][which(tok[[1]]%in%pos[,1])],"\n")
  neg.count <- sum(tok[[1]]%in%neg[,1])
```

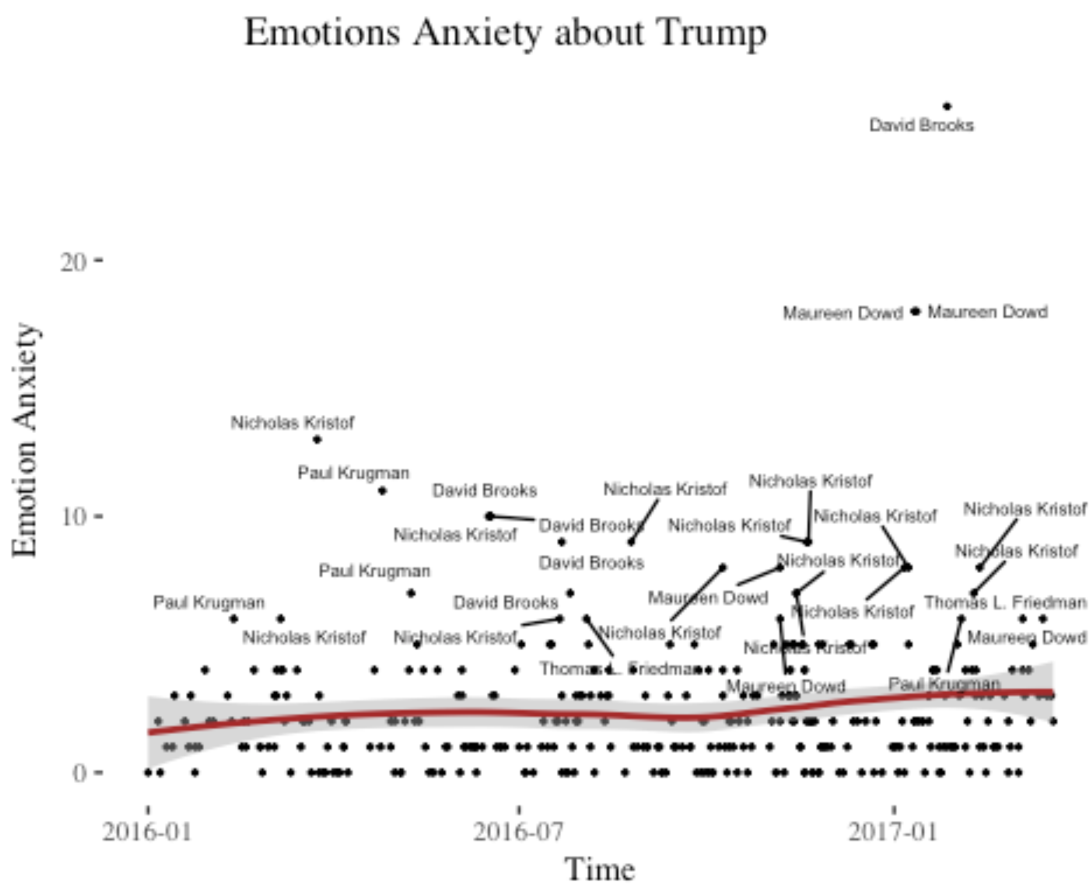
```
  cat("\n negative words:",tok[[1]][which(tok[[1]]%in%neg[,1])],"\n")
  out <- (pos.count - neg.count)/(pos.count+neg.count)
  cat("\n Tone of Document:",out)
}
sent_tr <- sentiment(corp_tr$documents$texts)
##  positive words: like admire applaud lead popular support admiration win g
enius good benefits educated faith lead educated smartest compassion reform r
eform like work top right right like secure steady welcome silent hopeful
##  negative words: resigned crushed destruction meaningless mirage insane te
rrible endanger incapable incompetent incapable savage trauma suicide incompe
tent corrupt conservative conservative dispute anger pathetically
##  Tone of Document: 0.1764706

sent_tr

sent_cl <- sentiment(corp_cl$documents$texts)
##  positive words: leading amazing leading like ready lead win favor strikin
g lean consistently progressive celebration creative stability happy trust tr
usted leads greatest energetic creative sensitive available genius perfectly
benefits reforms like lean clear improving richer solid supporting lead happy

##  negative words: crushing crash stagnant conservative conservative conserv
ative crisis instability lack suspicious skeptical risk oppose
##  Tone of Document: 0.48

sent_cl

ename <- file.path("~", "Desktop", "RID.CAT")
RID_dictionary <- dictionary(file = ename,
                             format = "wordstat")

dtm_rid_tr <- dfm(corp_tr, dictionary = RID_dictionary)
dtm_rid_cl <- dfm(corp_cl, dictionary = RID_dictionary)

library(reshape)

##
## Attaching package: 'reshape'

## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast

## The following objects are masked from 'package:tidyr':
##
##     expand, smiths

## The following object is masked from 'package:dplyr':
##
##     rename
```

```
## The following object is masked from 'package:qdap':
##
##     condense
```

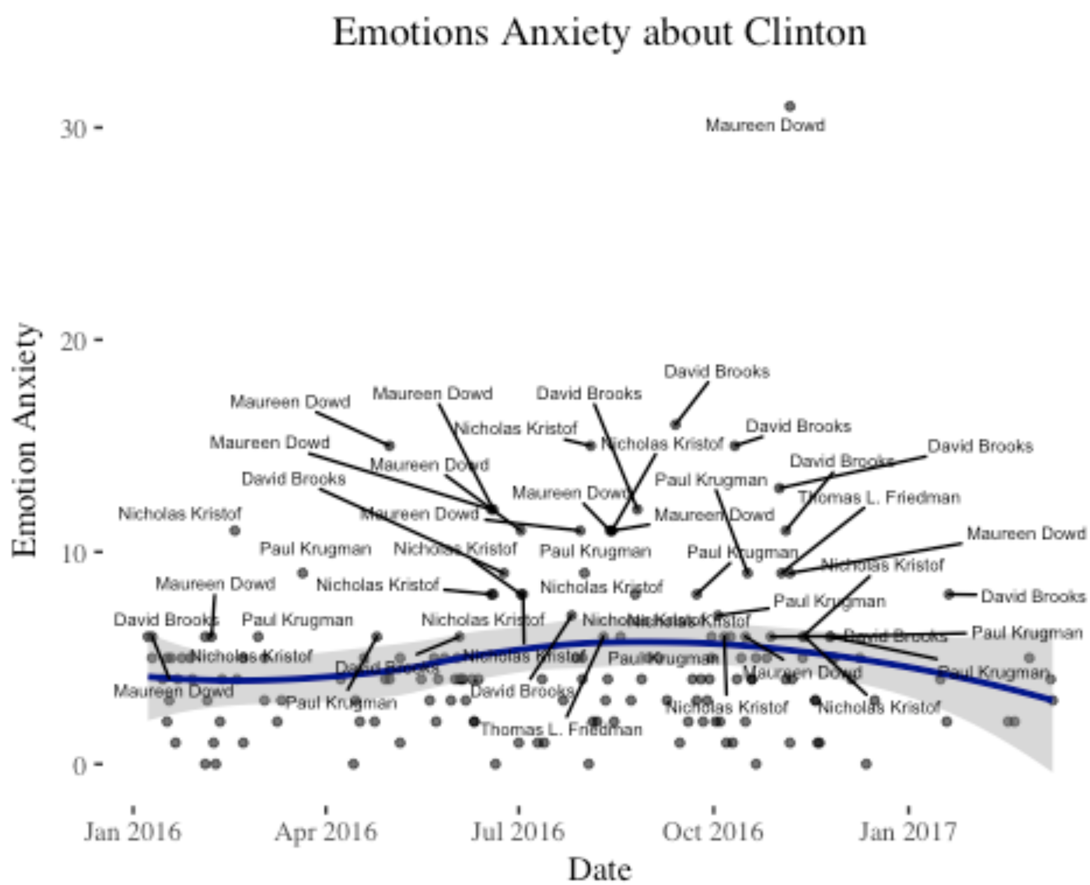
```
RIDdf_tr <- melt(as.matrix(dtm_rid_tr))
RIDdf_tr$author <- df_tr$author
RIDdf_tr$date <- as.Date(df_tr$date)
RIDdf_tr <- as_data_frame(RIDdf_tr)

RIDdf_cl <- melt(as.matrix(dtm_rid_cl))
RIDdf_cl$author <- df_cl$author
RIDdf_cl$date <- as.Date(df_cl$date)
RIDdf_cl <- as_data_frame(RIDdf_cl)
```

```
library(ggrepel)

anxiety_tr <- ggplot(filter(RIDdf_tr, features=="EMOTIONS.ANXIETY._"),
                  aes(x=date, y=value)) +
                  geom_point(size =0.5) +
                  ylab("Emotion Anxiety") +
                  xlab("Time") +
                  theme_tufte() +
                  geom_smooth(color = "brown") +
                  geom_text_repel(data=filter(RIDdf_tr,
                  features=="EMOTIONS.ANXIETY._", value > 5),
                  aes(label = author), size=2) +
                  ggtitle("Emotions Anxiety about Trump") +
                  theme(plot.title = element_text(hjust = 0.3))
anxiety_tr
```

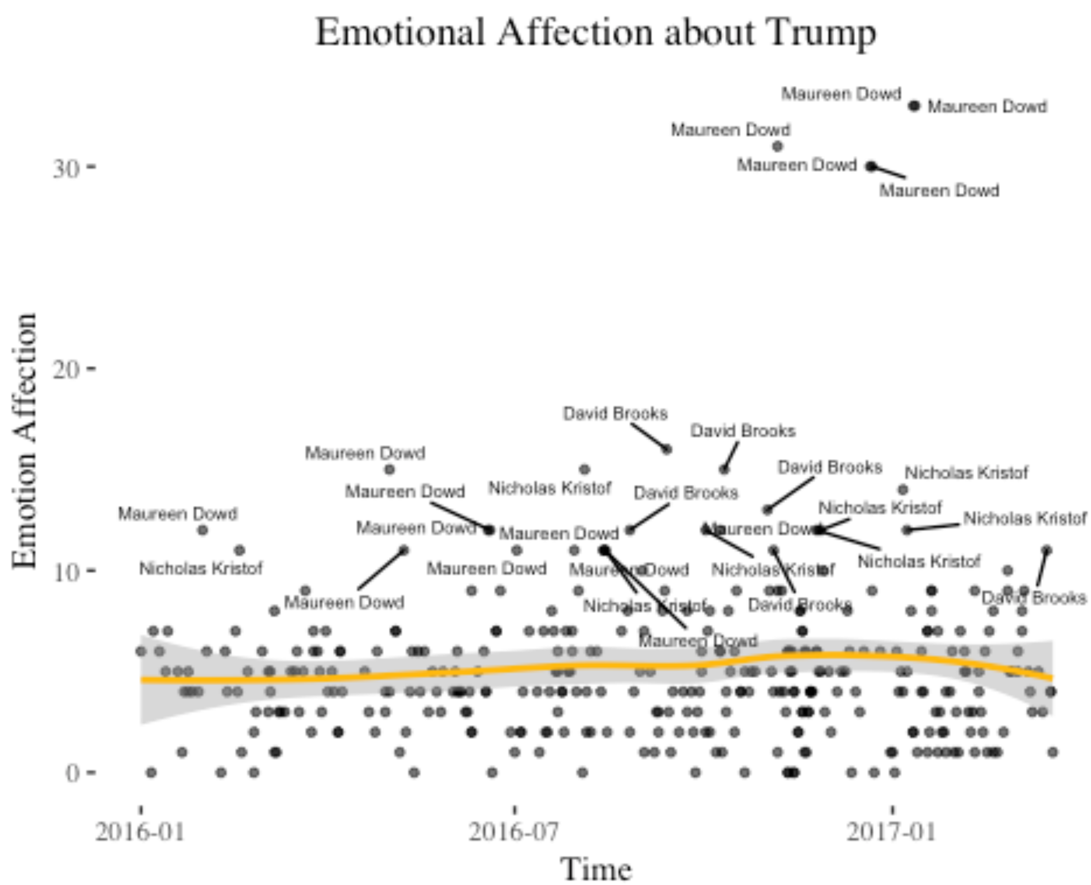
```
## `geom_smooth()` using method = 'loess'
```



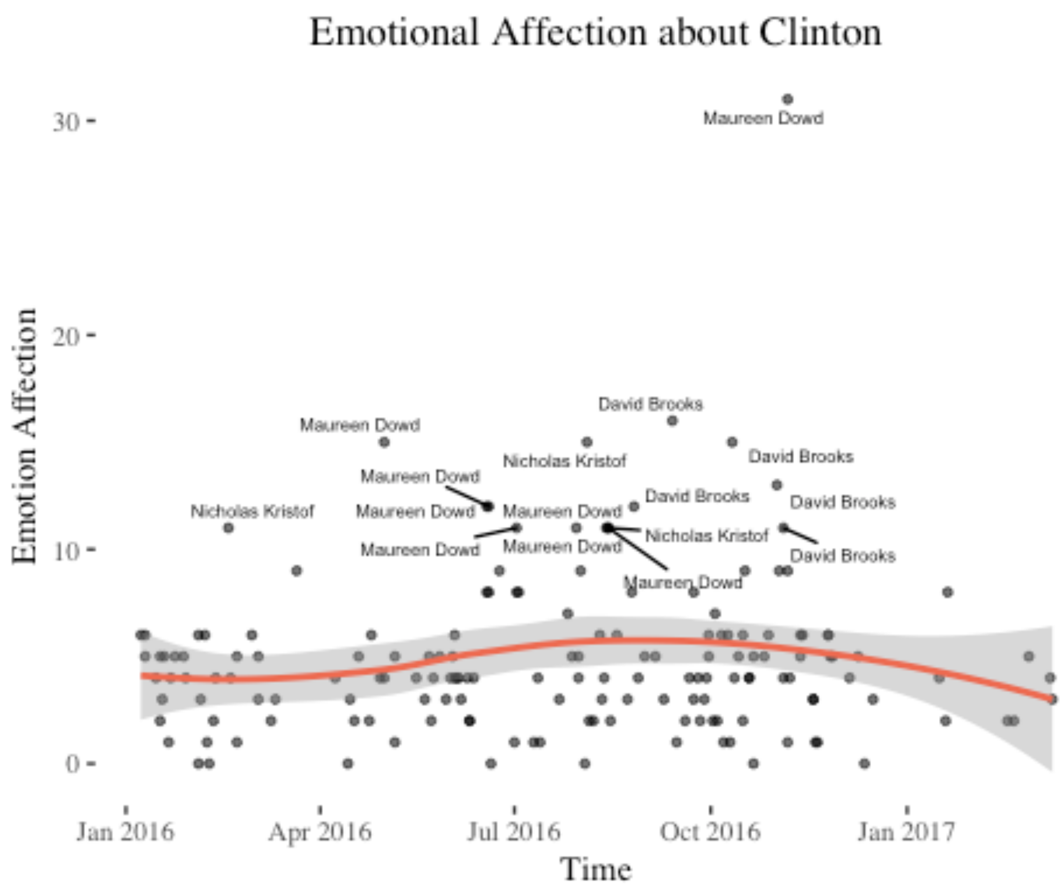

```
anxiety_cl <- ggplot(filter(RIDdf_cl, features=="EMOTIONS.AFFECTION._"),
                    aes(x=date, y=value)) +
                    geom_point(size = 1, alpha = 0.6) +
                    ylab("Emotion Anxiety") +
                    xlab("Date") +
                    theme_tufte() +
                    geom_smooth(color = "darkblue") +
                    geom_text_repel(data=filter(RIDdf_cl,
                    features=="EMOTIONS.AFFECTION._", value > 5),
                    aes(label = author), size=2) +
                    ggtitle("Emotions Anxiety about Clinton") +
                    theme(plot.title = element_text(hjust = 0.5))
anxiety_cl

## `geom_smooth()` using method = 'loess'
```




```
# plot trump's emotion affection considered by columnists

affection_tr <- ggplot(filter(RIDdf_tr, features=="EMOTIONS.AFFECTION._"),
                  aes(x=date, y=value)) +
                  geom_point(size = 1, alpha = 0.6) +
                  ylab("Emotion Affection") +
                  xlab("Time") +
                  theme_tufte() +
                  geom_smooth(color = "darkgoldenrod1",size=1) +
                  geom_text_repel(data=filter(RIDdf_tr,
                  features=="EMOTIONS.AFFECTION._", value > 10),
                  aes(label = author), size=2) +
                  ggtitle("Emotional Affection about Trump") +
                  theme(plot.title = element_text(hjust = 0.5))
affection_tr

## `geom_smooth()` using method = 'loess'
```




```
affection_cl<- ggplot(filter(RIDdf_cl, features=="EMOTIONS.AFFECTION._"),
                  aes(x=date, y=value)) +
                  geom_point(size = 1, alpha = 0.6) +
                  ylab("Emotion Affection") +
                  xlab("Time") +
                  theme_tufte() +
                  geom_smooth(color = "coral2") +
                  geom_text_repel(data=filter(RIDdf_cl,
                  features=="EMOTIONS.AFFECTION._", value > 10),
                  aes(label = author), size=2) +
                  ggtitle("Emotional Affection about Clinton") +
                  theme(plot.title = element_text(hjust = 0.5))
affection_cl

## `geom_smooth()` using method = 'loess'
```




```
hname <- file.path("~", "Desktop", "LaverGarry.cat")
LG_dictionary <- dictionary(file=hname,
                            format = "wordstat")
dtm_LG_tr <- dfm(corp_tr, dictionary=LG_dictionary,
                 groups = c("author","date"))
dtm_LG_cl <- dfm(corp_cl, dictionary=LG_dictionary,
                 groups = c("author","date"))
LG_tidy_tr <- tidy(dtm_LG_tr)
LG_tidy_tr <- separate(LG_tidy_tr, document, c("author","date"), extra = "mer
ge")
LG_tidy_tr$date = substr(LG_tidy_tr$date,nchar(LG_tidy_tr$date) - 9,nchar(LG_
tidy_tr$date))
LG_tidy_tr$date = as.Date(LG_tidy_tr$date)

LG_tidy_cl <- tidy(dtm_LG_cl)
LG_tidy_cl <- separate(LG_tidy_cl, document, c("author","date"), extra = "mer
ge")
LG_tidy_cl$date = substr(LG_tidy_cl$date,nchar(LG_tidy_cl$date) - 9,nchar(LG_
tidy_cl$date))
LG_tidy_cl$date = as.Date(LG_tidy_cl$date)
unique(LG_tidy_tr$term)
```

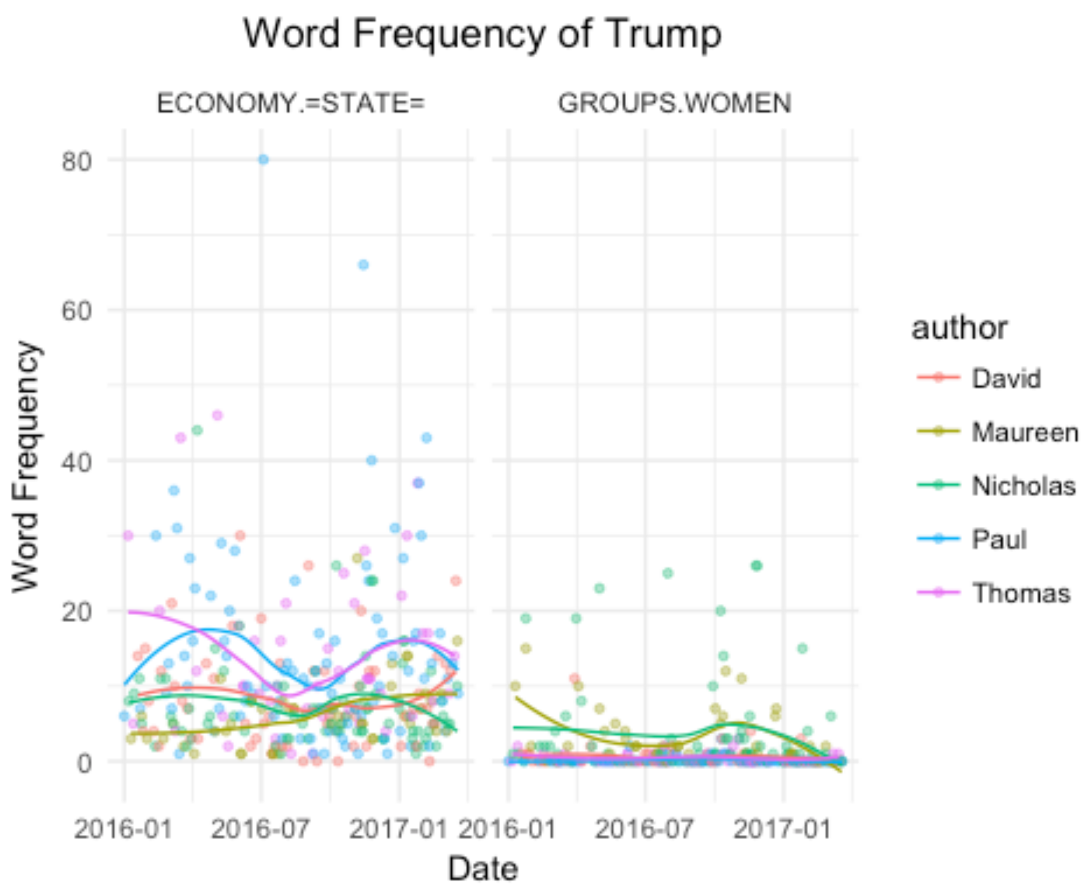
```
##  [1] "CULTURE.CULTURE-HIGH"           "CULTURE.CULTURE-POPULAR"
##  [3] "CULTURE.SPORT"                  "CULTURE._"
##  [5] "ECONOMY.+STATE+"                "ECONOMY.=STATE="
##  [7] "ECONOMY.-STATE-"                "ENVIRONMENT.CON ENVIRONMENT"
##  [9] "ENVIRONMENT.PRO ENVIRONMENT"    "GROUPS.ETHNIC"
## [11] "GROUPS.WOMEN"                   "INSTITUTIONS.CONSERVATIVE"
## [13] "INSTITUTIONS.NEUTRAL"           "INSTITUTIONS.RADICAL"
## [15] "LAW_AND_ORDER.LAW-CONSERVATIVE" "LAW_AND_ORDER.LAW-LIBERAL"
## [17] "RURAL._"                        "URBAN._"
## [19] "VALUES.CONSERVATIVE"            "VALUES.LIBERAL"
```

```
LG_tidy_tr[, c("term", "count")]
```

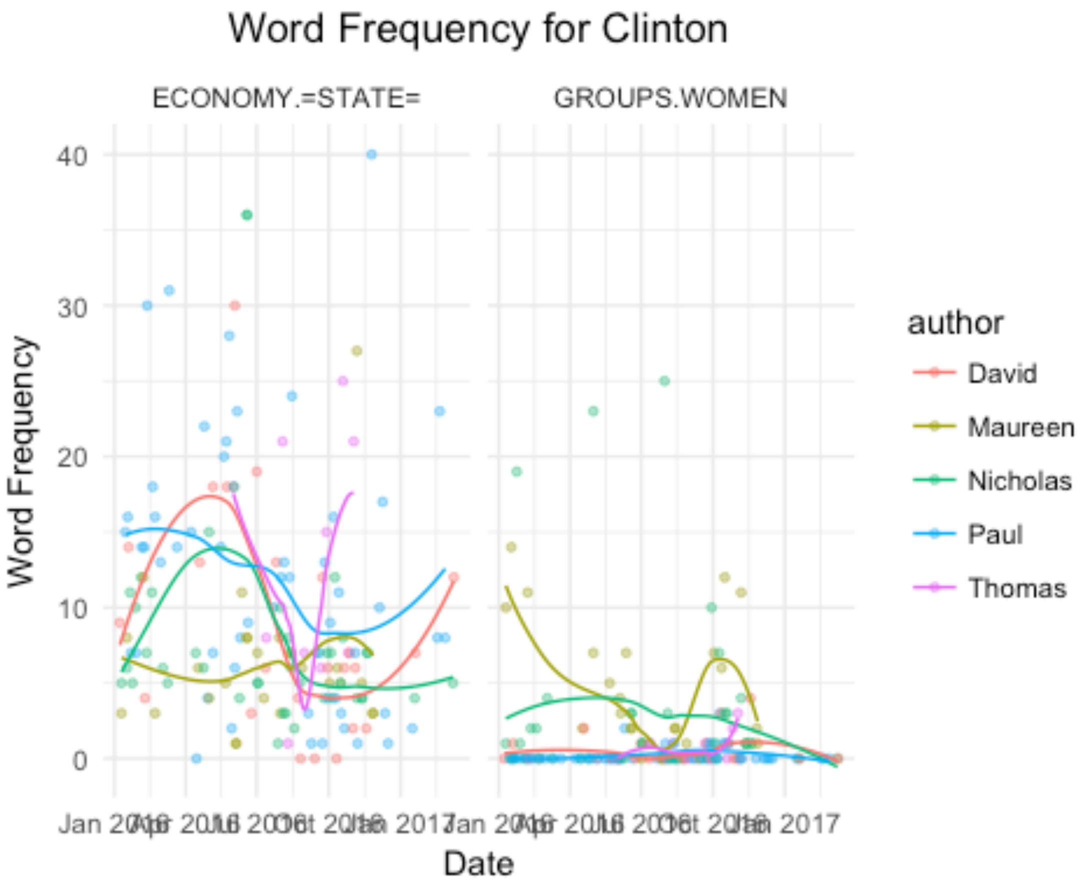
```
## # A tibble: 6,360 × 2
##                   term count
##                  <chr> <dbl>
## 1  CULTURE.CULTURE-HIGH     2
## 2  CULTURE.CULTURE-HIGH     0
## 3  CULTURE.CULTURE-HIGH     1
## 4  CULTURE.CULTURE-HIGH     1
## 5  CULTURE.CULTURE-HIGH     1
## 6  CULTURE.CULTURE-HIGH     0
## 7  CULTURE.CULTURE-HIGH     0
## 8  CULTURE.CULTURE-HIGH     0
## 9  CULTURE.CULTURE-HIGH     1
## 10 CULTURE.CULTURE-HIGH     0
## # ... with 6,350 more rows
```

```
gl_tr <- ggplot(filter(LG_tidy_tr,
       term %in% c("GROUPS.WOMEN" ,"ECONOMY.=STATE=")),
       aes(x = date, y = count, color = author, group = author)) +
       facet_wrap(~ term) +
       geom_point(size=1, alpha=0.4) +
       ylab("Word Frequency") +
       xlab("Date") +
       geom_smooth(se=F,size=0.5) +
       theme_minimal() +
       ggtitle("Word Frequency of Trump") +
       theme(plot.title = element_text(hjust = 0.5))
 gl_tr
```

```
## `geom_smooth()` using method = 'loess'
```




```
gl_cl <- ggplot(filter(LG_tidy_cl,
       term %in% c("GROUPS.WOMEN" ,"ECONOMY.=STATE=")),
       aes(x = date, y = count, color = author, group = author)) +
       facet_wrap(~ term) +
       geom_point(size=1,alpha = 0.4) +
       ylab("Word Frequency") +
       xlab("Date") +
       geom_smooth(se=F,size=0.5) +
       theme_minimal() +
       ggtitle("Word Frequency for Clinton") +
       theme(plot.title = element_text(hjust = 0.5))
gl_cl

## `geom_smooth()` using method = 'loess'
```



Word Frequency for Clinton
ECONOMY.=STATE=
GROUPS.WOMEN
Word Frequency
Date
author
David
Maureen
Nicholas
Paul
Thomas