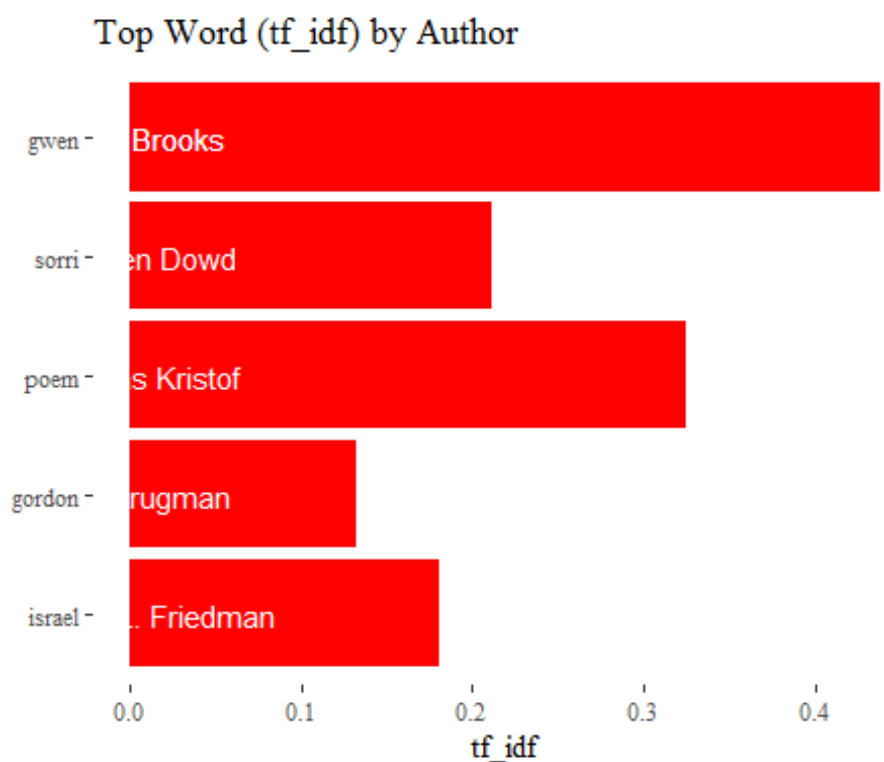
Assignment 3

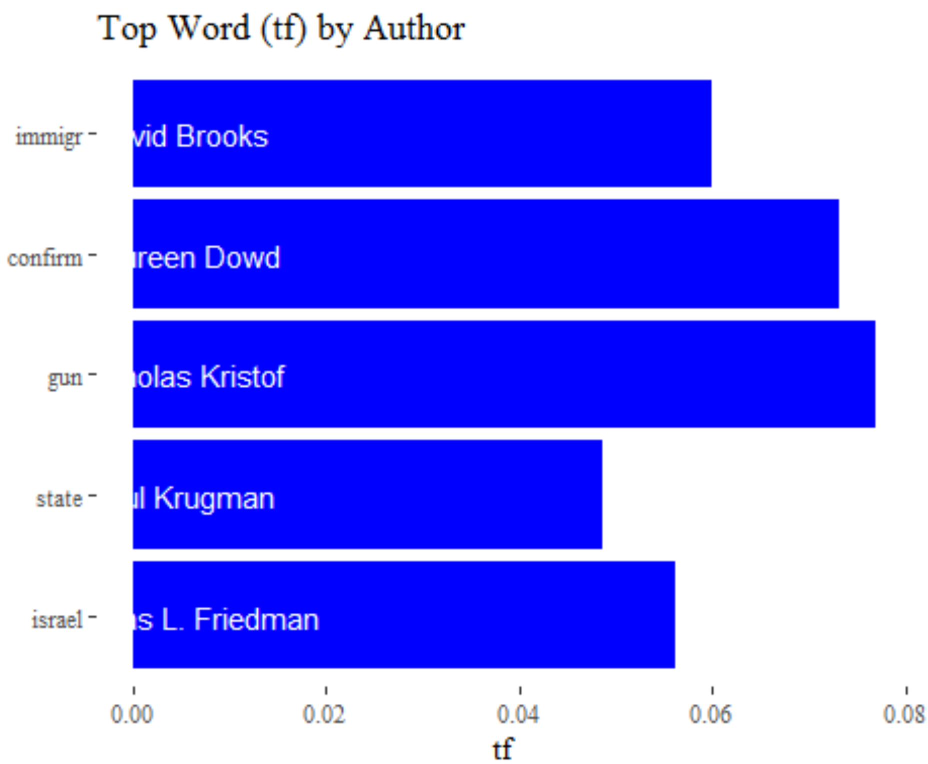
Brandon Wolff

March 24, 2017

Assignment 3

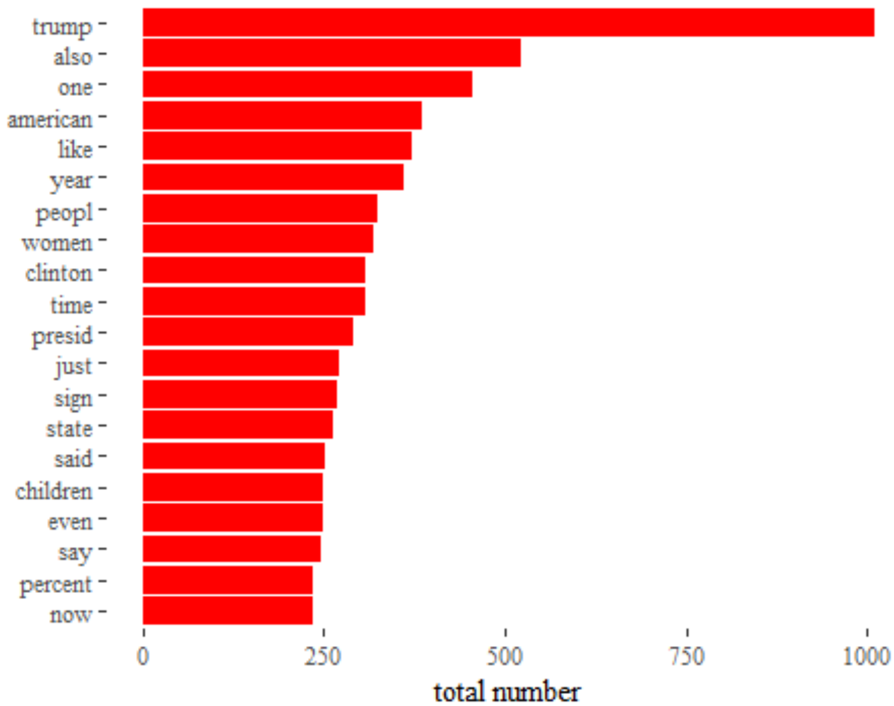


Above is a plot of each authors most frequent term according to the tf-idf measure. TF-IDF is a combination of the term frequency (TF) and the inverse document frequency (IDF) into a single numerical value. The resulting measurement decreases the weight for commonly used words and increases the weight for rare words (not in many documents). We can notice that for the authors David Brooks and Paul Krugman the most used terms seem to be the names gwen and gordon. For Nicholas Kristof the term is poem etc. But maybe we should look at the term frequency alone to see if it may be a better (give us more information) than tf-idf.

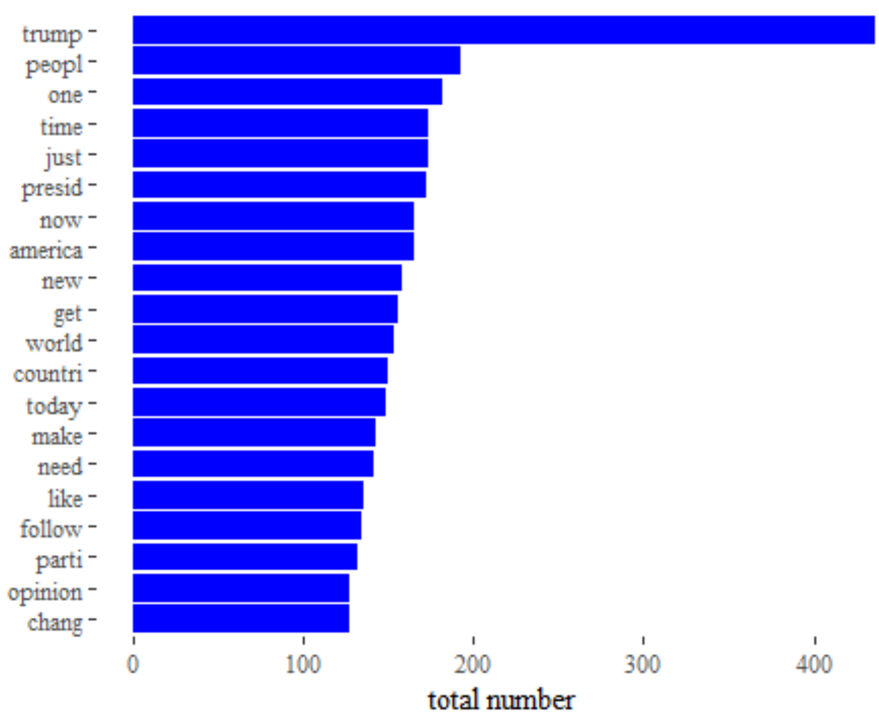


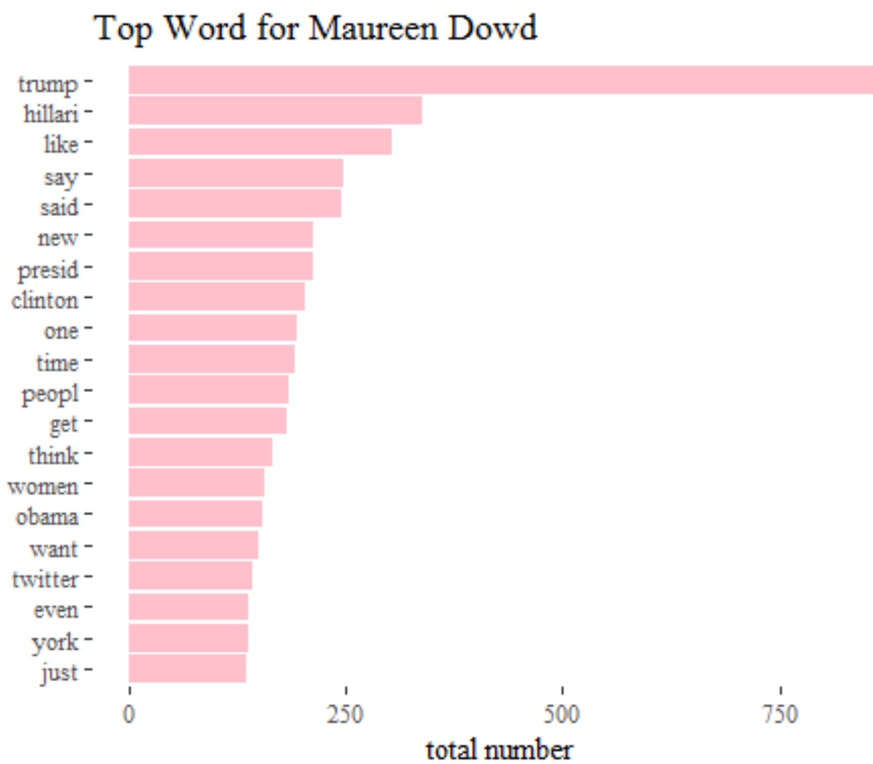
The plot above measuring the top word based on term frequency does appear to give us more information about the topics the authors seem to write about. We see David Brooks often uses the word immigration/immigrant and might typically write about immigration. Nicholas Kristof most often uses the term gun, maybe writing about gun control. Thomas Friedman most often uses the term Israel, potentially writing about Israeli often. Maureen Dowd most often uses the term confirm, and Paul Krugman most often uses the term state.

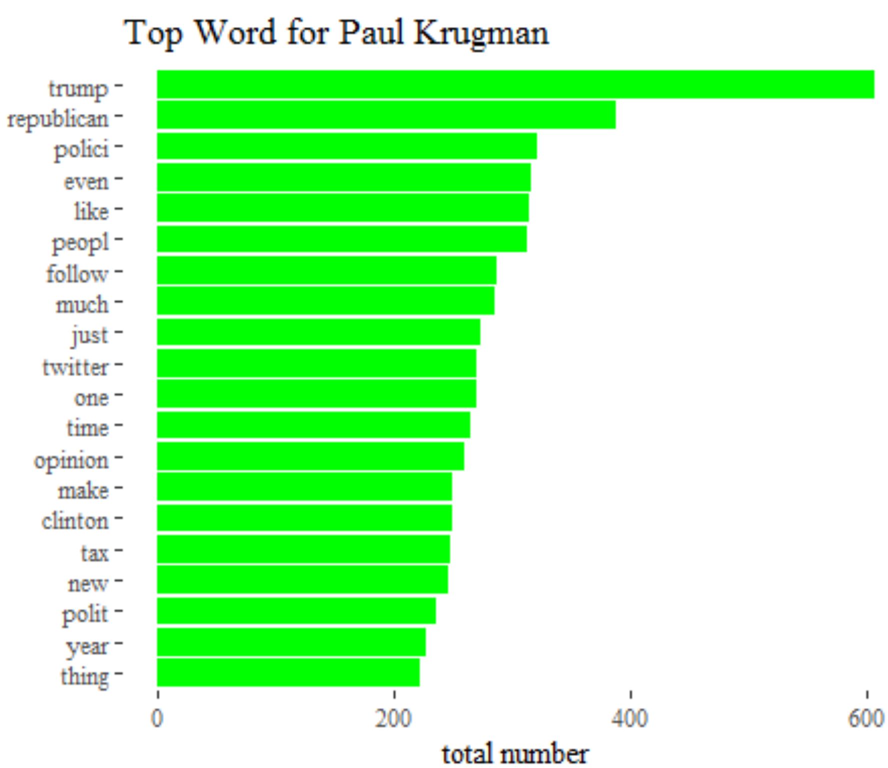
Top Word for Nicholas Kristof

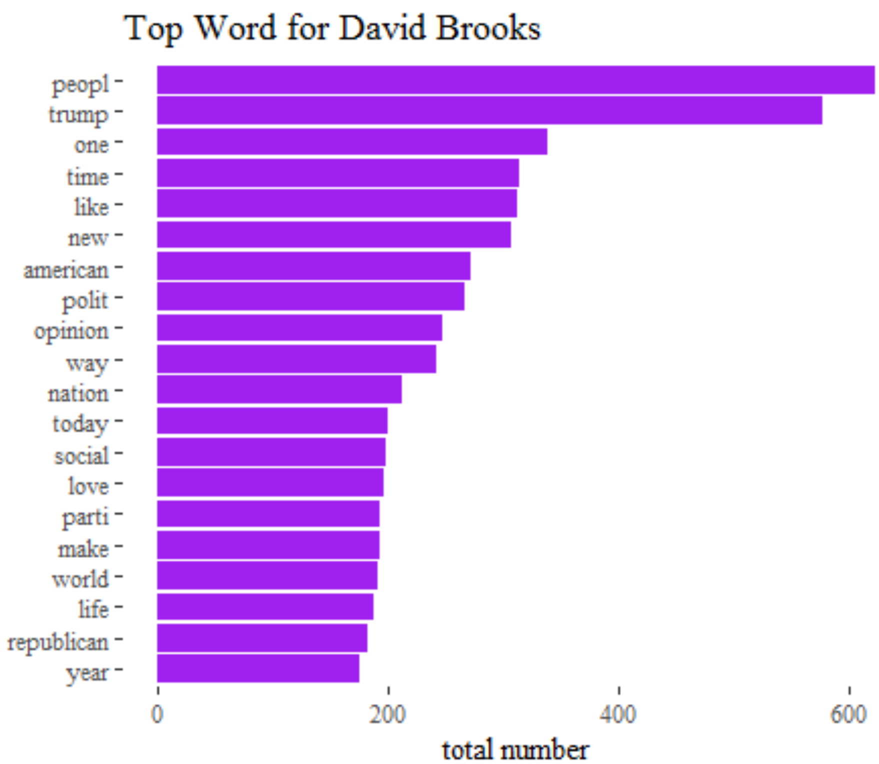


Top Word for Thomas L. Friedman

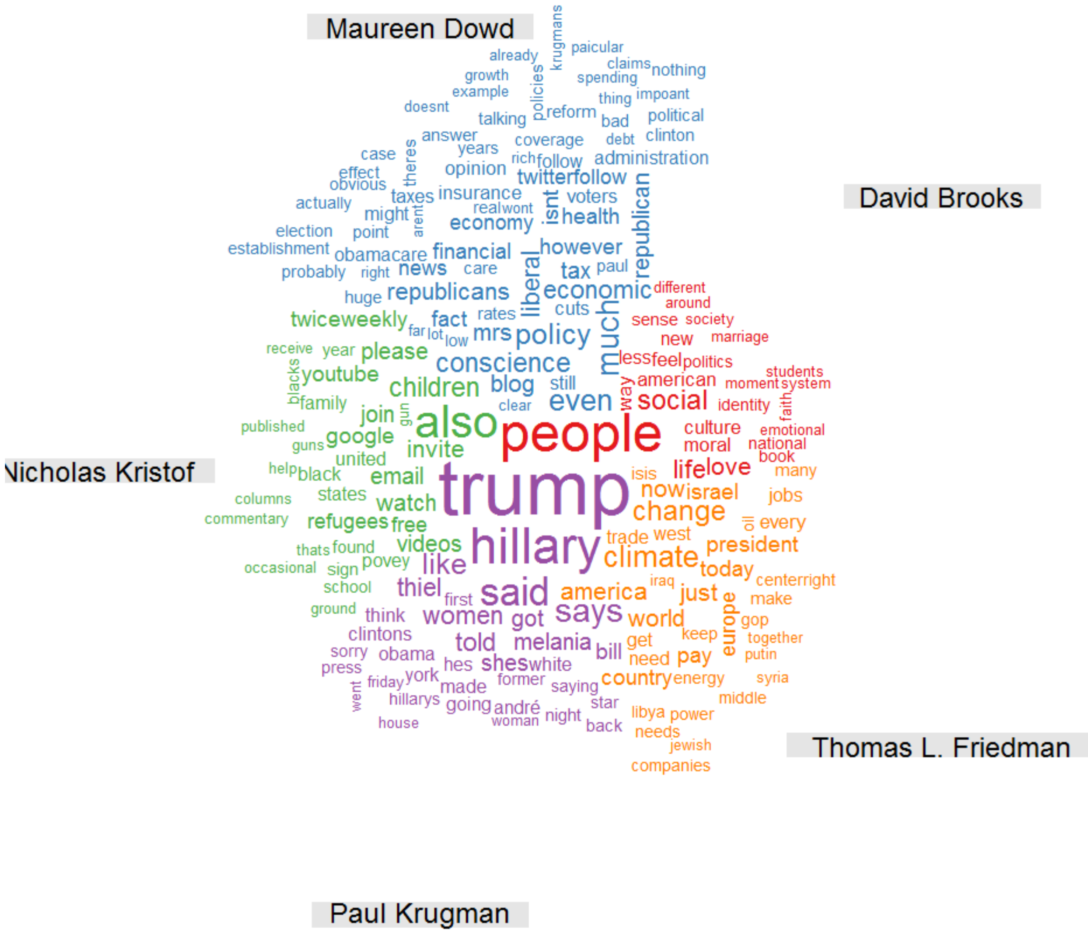



Top Word for Maureen Dowd
trump
hillari
like
say
said
new
presid
clinton
one
time
peopl
get
think
women
obama
want
twitter
even
york
just
0
250
500
750
total number


Top Word for Paul Krugman
trump
republican
polici
even
like
peopl
follow
much
just
twitter
one
time
opinion
make
clinton
tax
new
polit
year
thing
0
200
400
600
total number

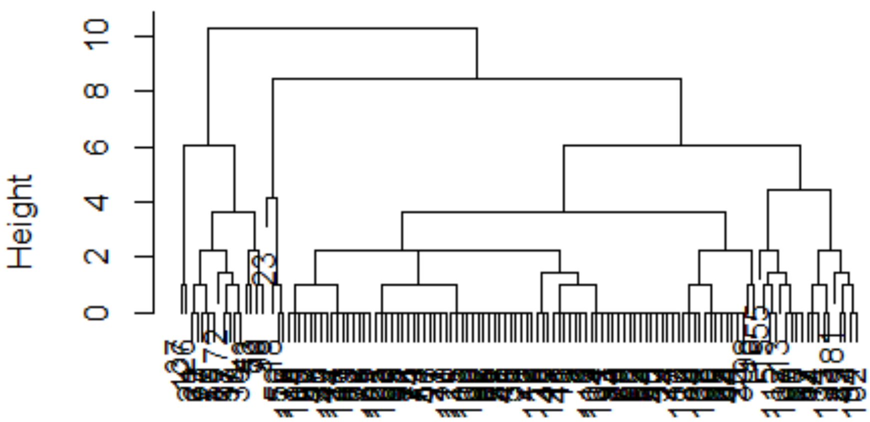


Above is a list of the top 20 words used by each individual author. We can very easily notice that the top term for all but the author David Brooks are "trump" and it is actually the second most used term by Brooks as well. We can conclude that all of these authors often speak of Trump which is not very surprising with the interest of the general public upon the topic. Nicolas Kristof and Maureen Dowd often use the terms "Clinton" and "women". Maybe Kristoff and Dowd also often talked about Clinton and women overall in comparison to the other three authors. Overall all the others do have very similar words as their top 20 and all seem to speak a lot about the election and politics.



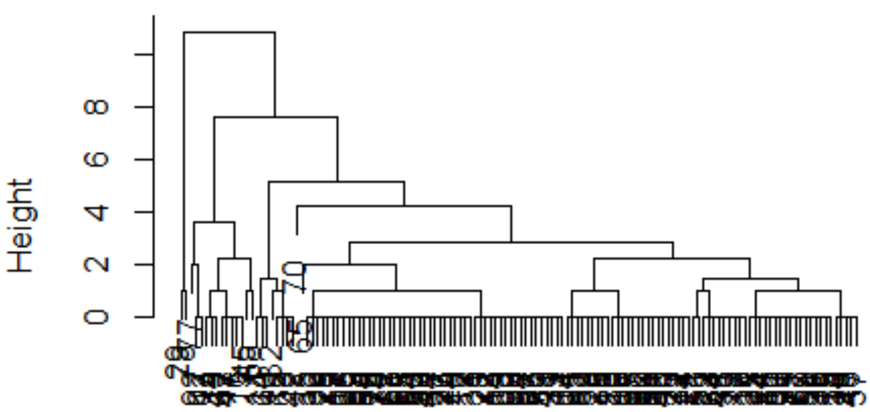
Above is a word cloud of top word counts by each other author. In a word cloud the larger the word appears the more frequently it is used. We can easily notice the large purple words Trump and Hillary. This means that Paul Krugman very often uses these two words. Paul Krugman must often write on the topics of the election and the candidates. WE can also notice the words people and social in red (David Brooks). It appears that Brooks often writes about social issues or "the people" of the Unites States. We see in orange the words climate and change (Thomas Friedman). This shows that Friedman most likely writes often about climate change. When looking at the green (Nicholas Kristof) and blue (Maureen Dowd) terms it is hard to pick out a specific topic.

David Brooks

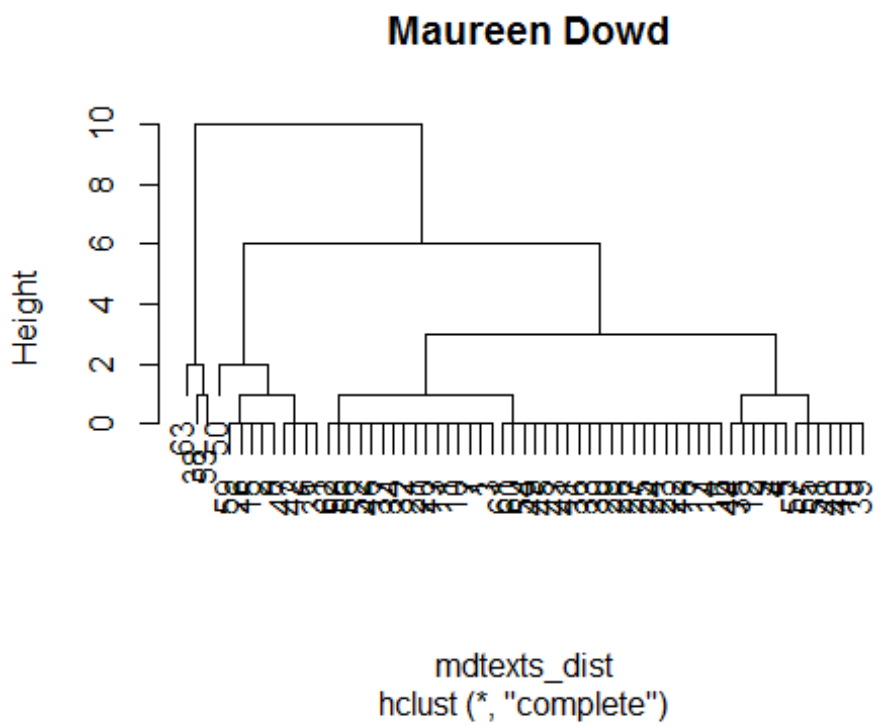


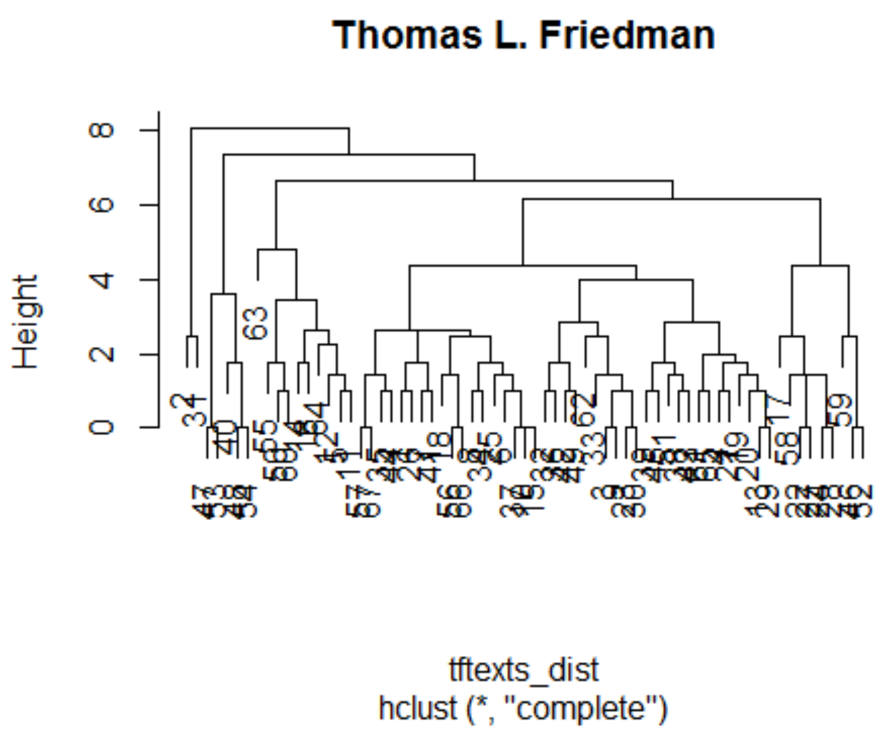
texts_dist
hclust (*, "complete")

Paul Krugman



pktexts_dist
hclust (*, "complete")





Above we see a dendrogram for each other which show us the differences of the articles the authors write and how similar some articles the authors write are. When looking at the differences between the dendrograms we see there is a lot of differentiation between articles written by Thomas L. Friedman. WE can also note that there does not seem to be much differentiation in articles written by Maureen Dowd and Paul Krugman. Nicholas Kristof was not included due to the fact I could not make the plot work properly.

1 bonus

```
## # A tibble: 20 × 1
##     `Top 20 subjects David Brooks`
##                              <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2              US REPUBLICAN PARTY
## 3                         POLITICS
## 4             CAMPAIGNS & ELECTIONS
## 5  US PRESIDENTIAL CANDIDATES 2012
## 6                 POLITICAL PARTIES
## 7                     US PRESIDENTS
## 8                      CONSERVATISM
## 9        US PRESIDENTIAL ELECTIONS
## 10                        ELECTIONS
## 11                         RELIGION
## 12 US PRESIDENTIAL CANDIDATES 2008
## 13             US DEMOCRATIC PARTY
## 14   HEADS OF GOVERNMENT ELECTIONS
## 15               POLITICAL DEBATES
## 16            POLITICAL CANDIDATES
## 17                 MUSLIMS & ISLAM
## 18                TAXES & TAXATION
## 19         INTERNATIONAL RELATIONS
## 20                      LIBERALISM
```

```
## # A tibble: 20 × 1
##    ` top 20 subjects Paul Krugman`
##                              <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2              US REPUBLICAN PARTY
## 3                         POLITICS
## 4             CAMPAIGNS & ELECTIONS
## 5  US PRESIDENTIAL CANDIDATES 2012
## 6                 POLITICAL PARTIES
## 7                     US PRESIDENTS
## 8                      CONSERVATISM
## 9                         RELIGION
## 10       US PRESIDENTIAL ELECTIONS
## 11                        ELECTIONS
## 12 US PRESIDENTIAL CANDIDATES 2008
## 13             US DEMOCRATIC PARTY
## 14   HEADS OF GOVERNMENT ELECTIONS
## 15            POLITICAL CANDIDATES
## 16               POLITICAL DEBATES
```

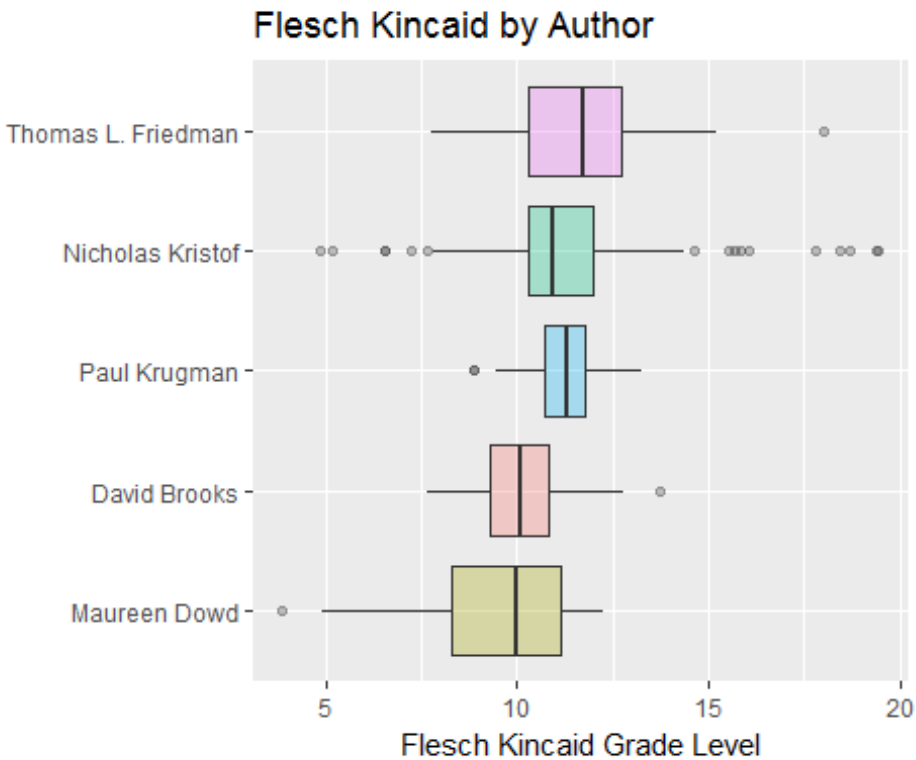
```
## 17                  MUSLIMS & ISLAM
## 18                      LIBERALISM
## 19                 TAXES & TAXATION
## 20         INTERNATIONAL RELATIONS

## # A tibble: 20 × 1
##    `Top 20 subjects Nicholas Kristof`
##                                 <chr>
## 1     US PRESIDENTIAL CANDIDATES 2016
## 2                 US REPUBLICAN PARTY
## 3                            POLITICS
## 4                       US PRESIDENTS
## 5     US PRESIDENTIAL CANDIDATES 2012
## 6                   POLITICAL PARTIES
## 7               CAMPAIGNS & ELECTIONS
## 8                            RELIGION
## 9                        CONSERVATISM
## 10    US PRESIDENTIAL CANDIDATES 2008
## 11                          ELECTIONS
## 12          US PRESIDENTIAL ELECTIONS
## 13                US DEMOCRATIC PARTY
## 14      HEADS OF GOVERNMENT ELECTIONS
## 15                  POLITICAL DEBATES
## 16                    MUSLIMS & ISLAM
## 17               POLITICAL CANDIDATES
## 18                           CHILDREN
## 19                         LIBERALISM
## 20            INTERNATIONAL RELATIONS

## # A tibble: 20 × 1
##     `Top 20 subjects Maureen Dowd`
##                              <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2              US REPUBLICAN PARTY
## 3                POLITICAL PARTIES
## 4                         POLITICS
## 5            CAMPAIGNS & ELECTIONS
## 6                     CONSERVATISM
## 7  US PRESIDENTIAL CANDIDATES 2012
## 8                    US PRESIDENTS
## 9                         RELIGION
## 10                TAXES & TAXATION
## 11                   ECONOMIC NEWS
## 12                       ELECTIONS
## 13         INTERNATIONAL RELATIONS
## 14             US DEMOCRATIC PARTY
## 15                  WEALTHY PEOPLE
## 16                  FOREIGN POLICY
## 17                         TAX LAW
## 18                     IMMIGRATION
```

```
## 19                       LIBERALISM
## 20                  MUSLIMS & ISLAM
```

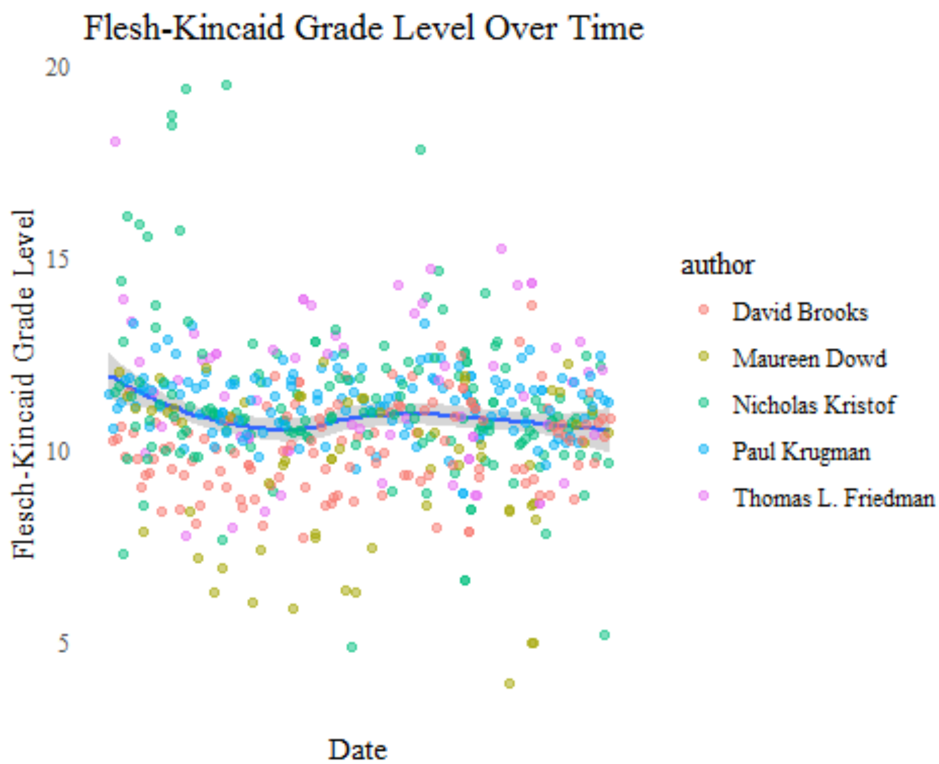
```
## # A tibble: 20 × 1
##    `Top 20 subjects Thomas L. Friedman`
##                                   <chr>
## 1       US PRESIDENTIAL CANDIDATES 2016
## 2                  US REPUBLICAN PARTY
## 3                    POLITICAL PARTIES
## 4                             POLITICS
## 5                CAMPAIGNS & ELECTIONS
## 6                         CONSERVATISM
## 7       US PRESIDENTIAL CANDIDATES 2012
## 8                        US PRESIDENTS
## 9                    TAXES & TAXATION
## 10                            RELIGION
## 11             INTERNATIONAL RELATIONS
## 12                 US DEMOCRATIC PARTY
## 13                       ECONOMIC NEWS
## 14                           ELECTIONS
## 15      US PRESIDENTIAL CANDIDATES 2008
## 16                      WEALTHY PEOPLE
## 17                      FOREIGN POLICY
## 18                             TAX LAW
## 19                         IMMIGRATION
## 20                          LIBERALISM
```

When looking above at the top 20 articles by each author we can notice that the top subject by all authors is "US Presidential Candidates 2016". The other subjects do have some slight variation but overall all of the top subjects by all of the authors are very similar and all seem to concentrate on politics. Again this is not surprising with the fact their election just recently took place and was very publicized.

2

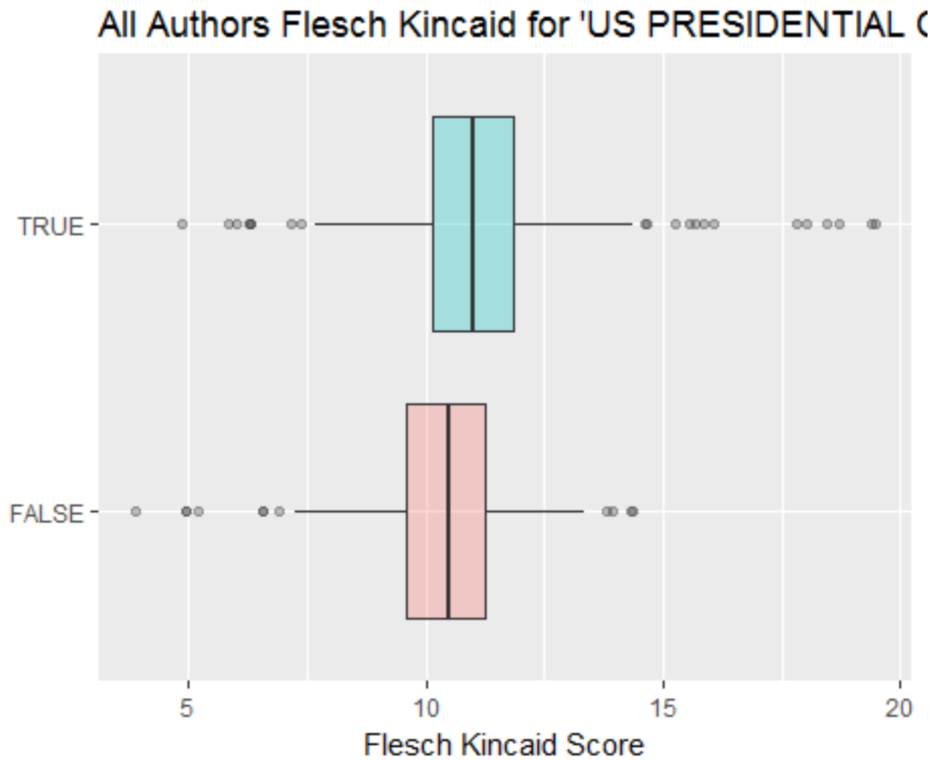


The visualization above is boxplot of each authors Flesch Kincaid Grade Level. It can be noted that Thomas Friedman has the highest average Flesch_Kincaid Grade Level with a mean around 11.5. We also see that Maureen Dowd has the lowest average Flesch Kincaid Grade Level with a mean around 10.

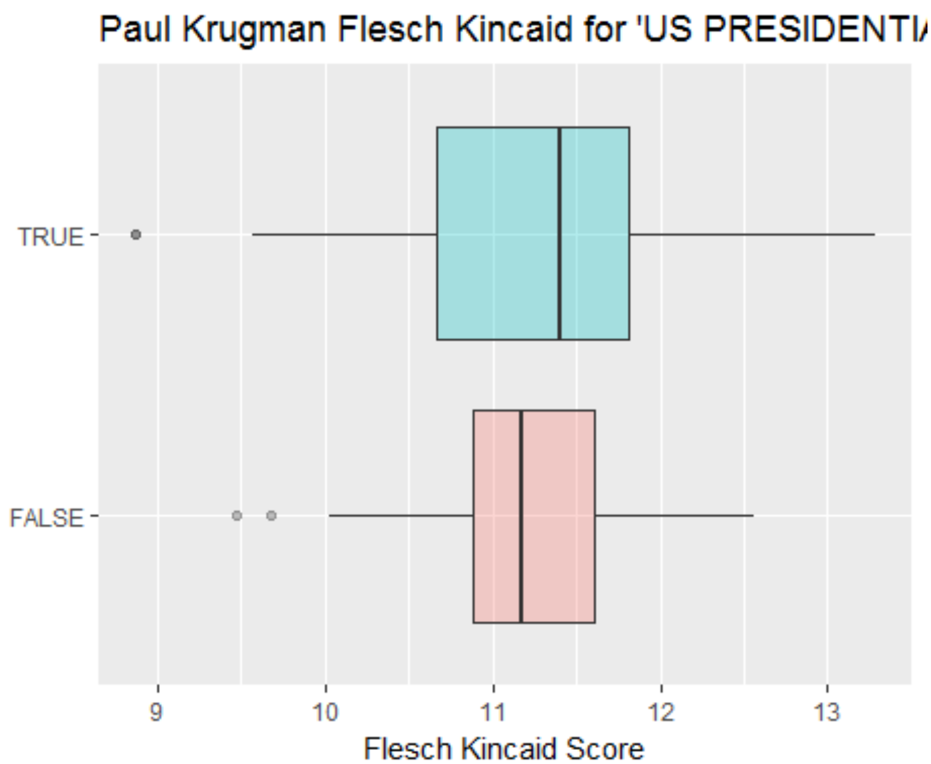


We can see from the graph above that overall the distribution of Flesch Kincaid grade level is pretty consistent. We also see that there are a number of green (Kristof) and purple (Friedman) dots way above the average and there are a number of yellow (Dowd) dots below the average.

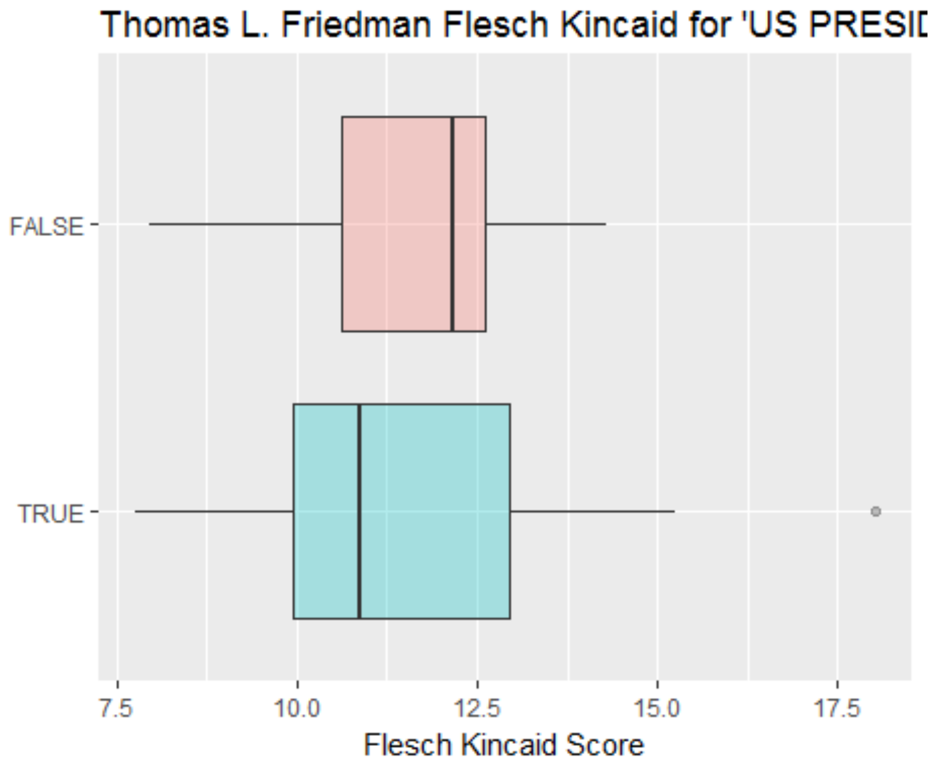
```
##               author    x.mean    x.count
## 1       David Brooks 10.085517 126.000000
## 2       Maureen Dowd  9.494868  63.000000
## 3   Nicholas Kristof 11.232045 157.000000
## 4       Paul Krugman 11.226747 134.000000
## 5 Thomas L. Friedman 11.560675  67.000000
```



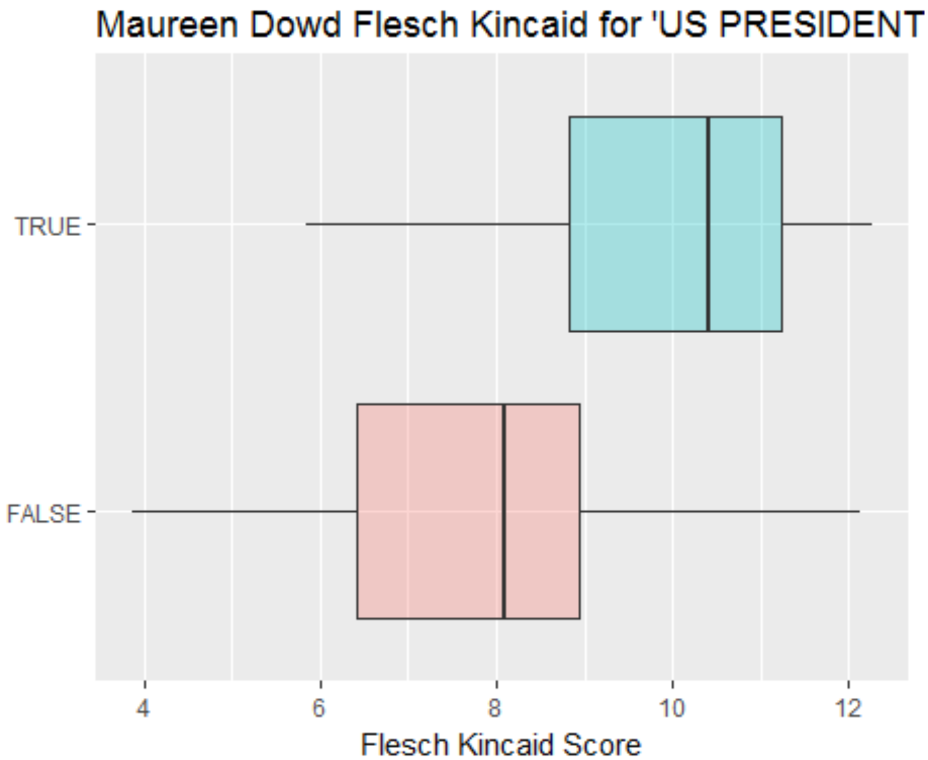
Above is boxplots of the Flesch-Kincaid grade level for the most popular subject (US Presidential Candidates 2016) among all of the authors. WE see that when writing about this subject the average grade level is around 11.5 which is higher than the average grade level of articles not about the US Presidential Candidates of 2016. Now we should examine this subjects grade level for each author individually to see how they may differ.



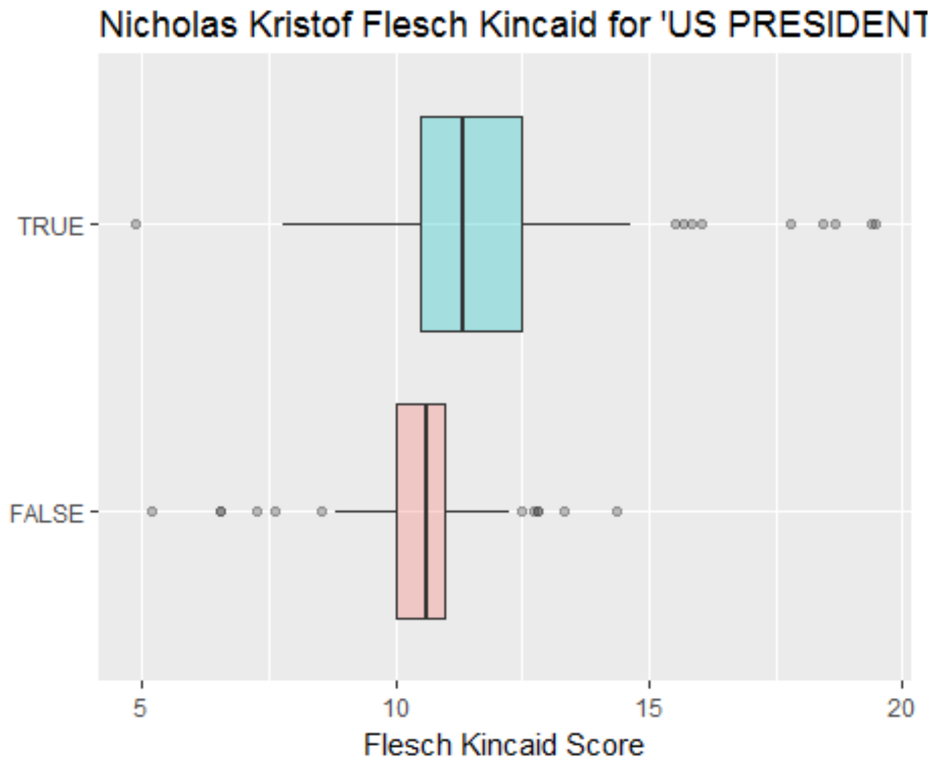
Pual Krugman has an average Flech-Kincaid grade level of about 12 for the subject US Presidential Candidates of 2016 and only a score of a little over 11 for all other articles. When writing about the US Presidential candidates of 2016 Paul seems to write at a slightly higher grade level.



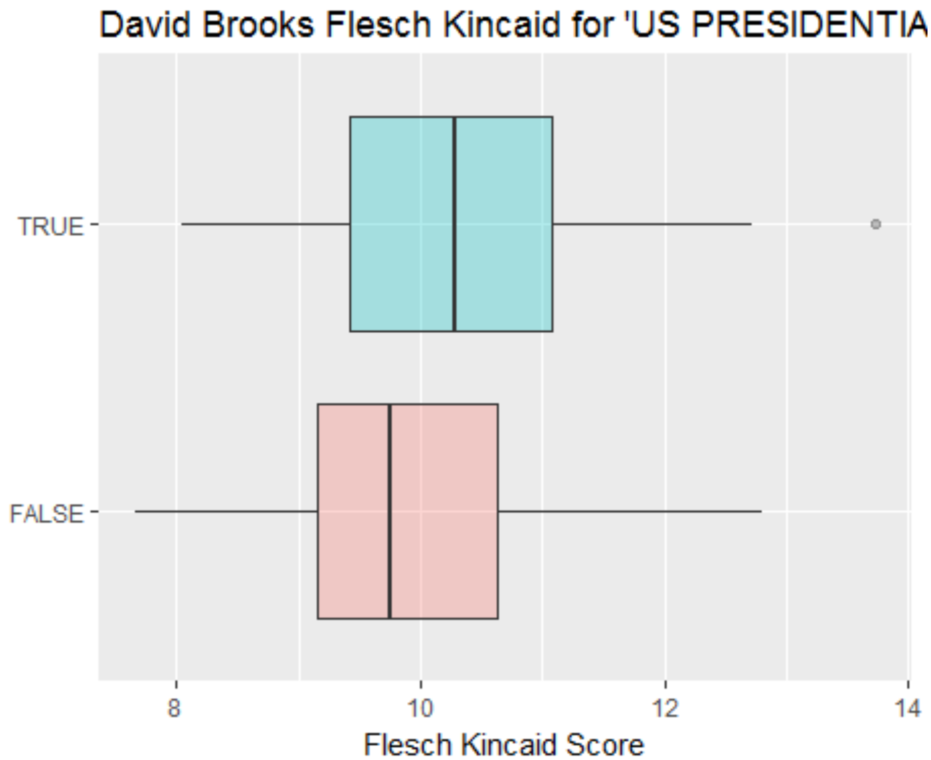
Thomas L. Friedman has an average Flech-Kincaid grade level of only about 11 for the subject US Presidential Candidates of 2016 and about 12 for all other articles. When writing about the US Presidential candidates of 2016 Friedman seems to write at a slightly lower grade level.



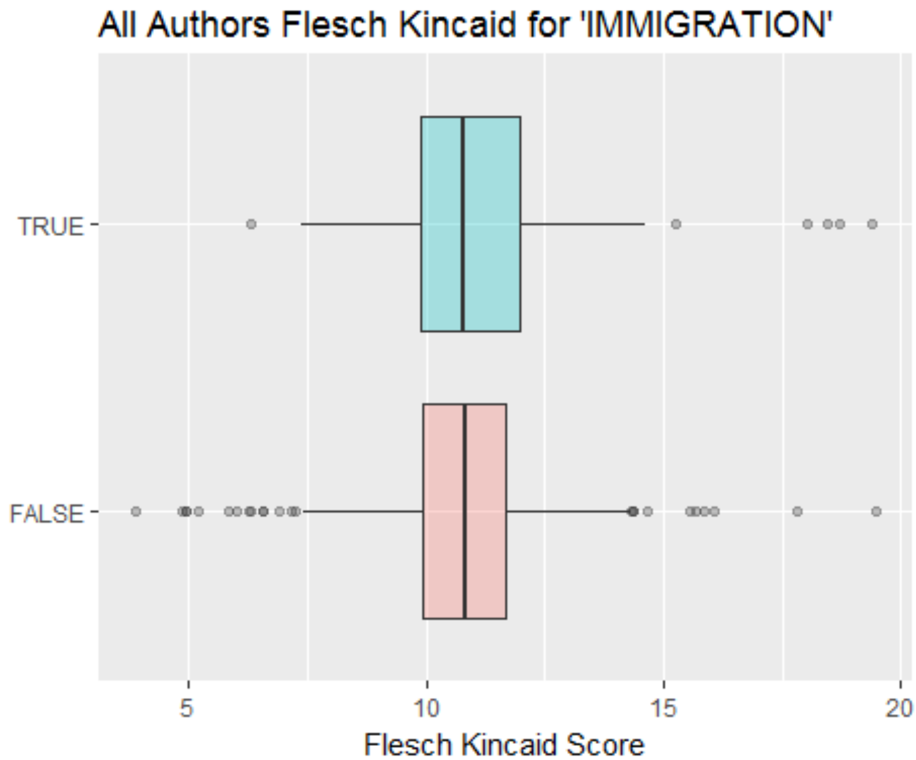
Maureen Dowd has an average Flech-Kincaid grade level of about 10.5 for the subject US Presidential Candidates of 2016 and only about 8 for all other articles. Maureen Dowd has a lower grade level compared to the other authors but just like most other authors when writing about the US Presidential candidates of 2016 Dowd seems to write at a slightly higher grade level.



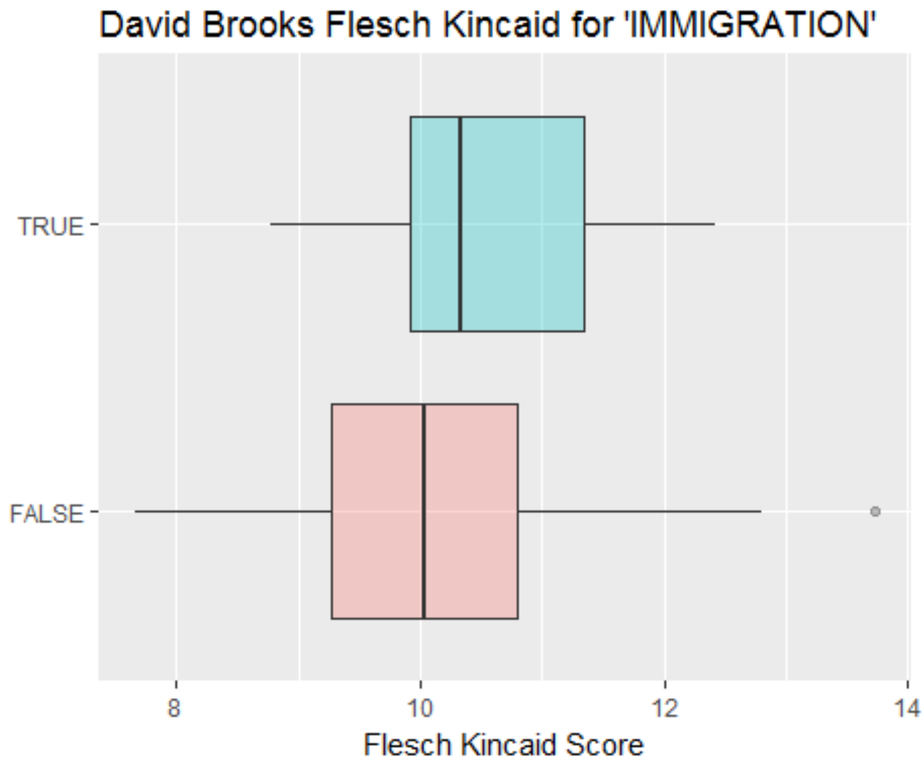
Nicholas Kristof has an average Flech-Kincaid grade level of about 11 for the subject US Presidential Candidates of 2016 and only about 10.5 for all other articles. When writing about the US Presidential candidates of 2016 Kristof seems to write at a slightly higher grade level. Kristoff and Krugman have very similar grade levels on average.



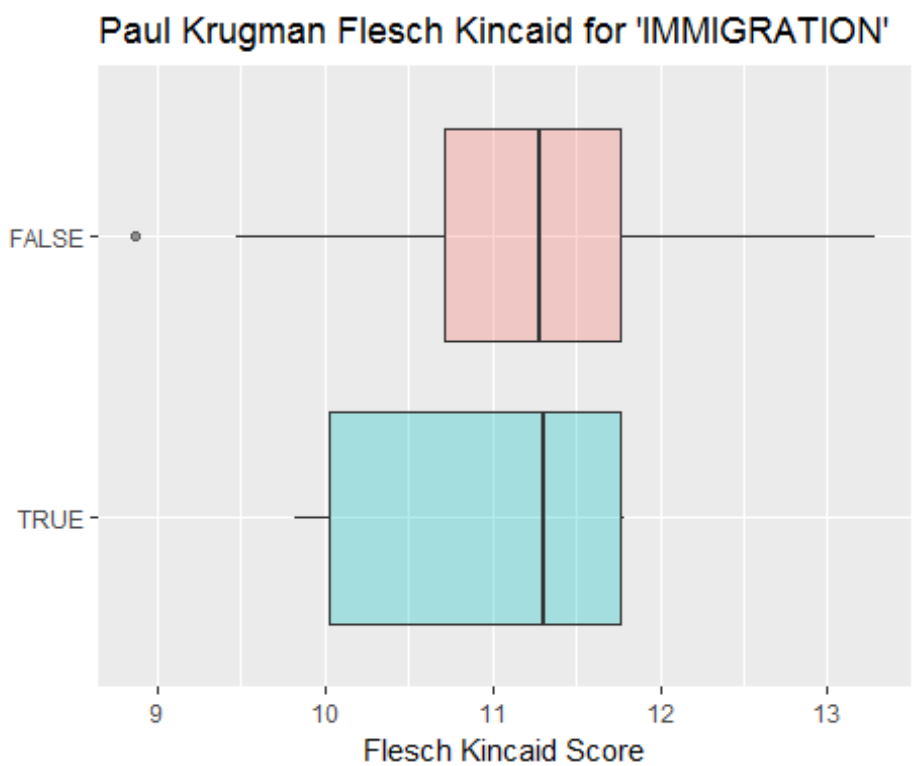
David Brooks has an average Flech-Kincaid grade level of about 10.5 for the subject US Presidential Candidates of 2016 and only about 9.5 for all other articles. When writing about the US Presidential candidates of 2016 Brooks seems to write at a slightly higher grade level. Brooks seems to have higher grade levels than Dowd but slightly lower than Kristoff, Friedman, and Krugman on average.



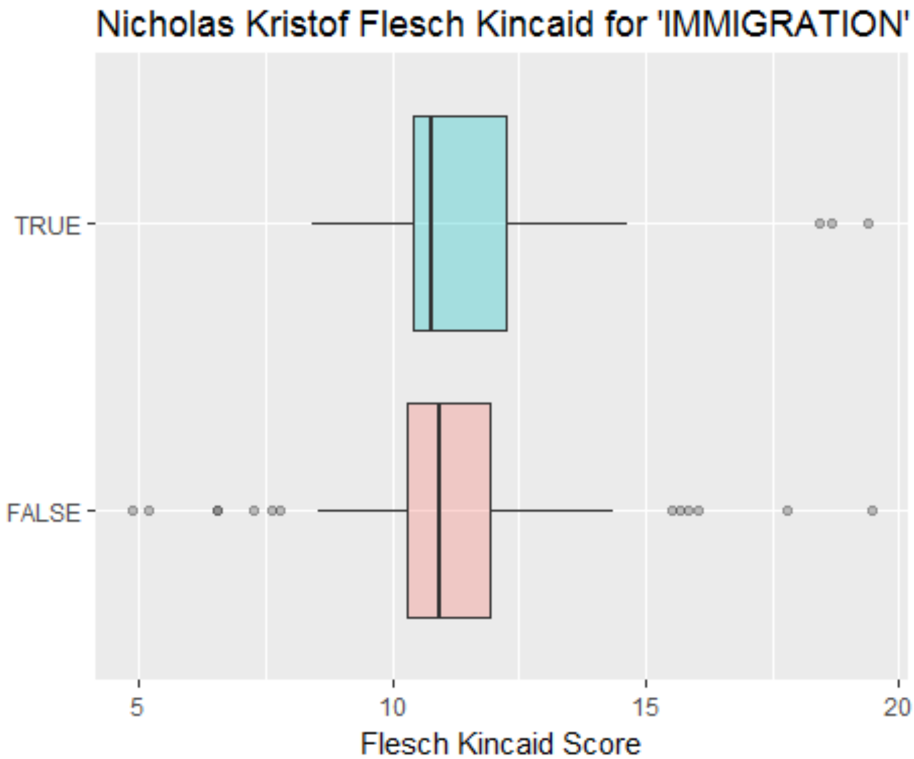
Above is boxplots of the Flesch-Kincaid grade level for another popular subject (immigration) among all of the authors. We see that when writing about this subject the average grade level is around 11 which is nearly identical to the average grade level of articles not about Immigration. Now we should examine this subjects grade level for each author individually to see how they may differ.



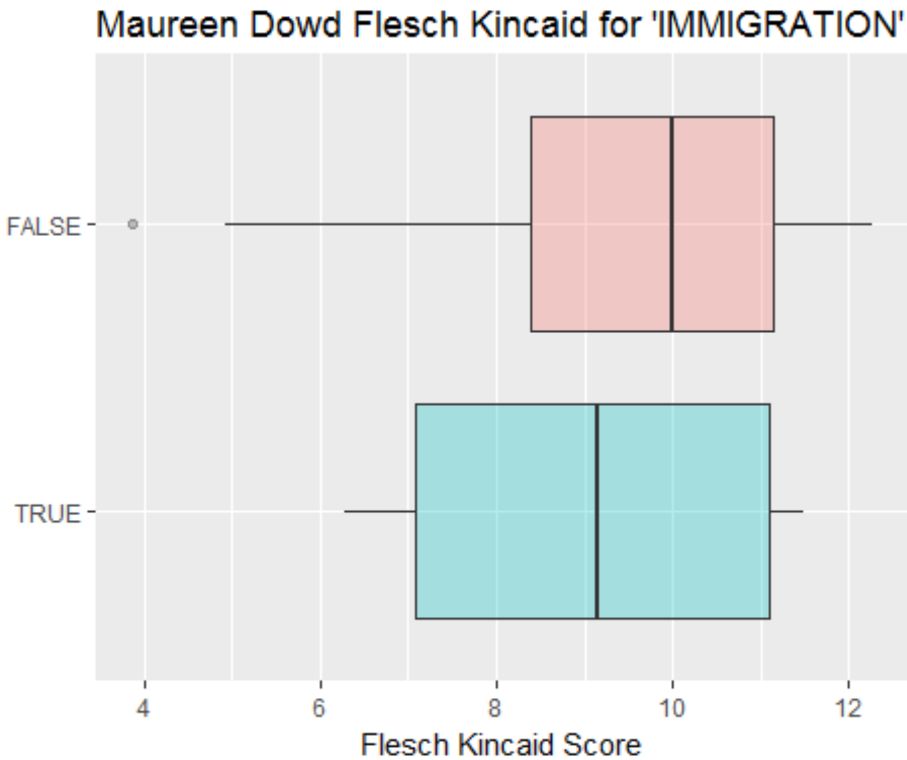
David Brooks has an average Flech-Kincaid grade level of about 10.5 for the subject Immigration and only a score of a little over 10 for all other articles. When writing about the Immigration Brooks seems to write at a slightly higher grade level.



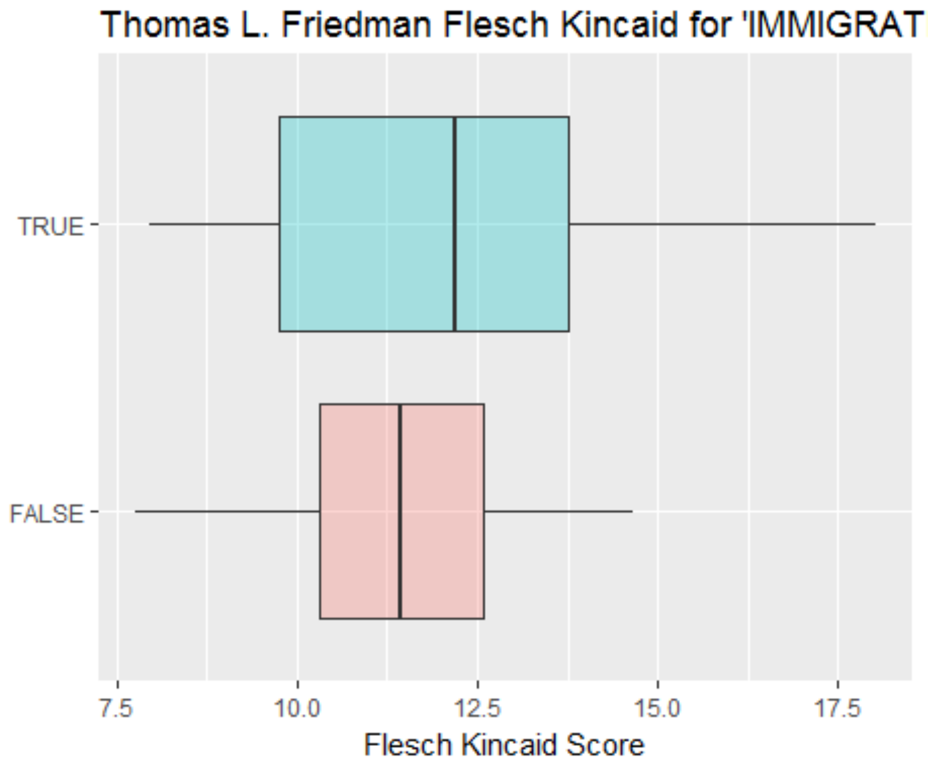
Paul Krugman has an average Flech-Kincaid grade level of about 11.5 for the subject Immigration and also a score of about 11.5 for all other articles. When writing about the Immigration Brooks seems to write at athe same grade level as he would on other subjects, on average.



Nicholas Kristof has an average Flech-Kincaid grade level of about 11 for the subject Immigration and also a score of about 11 for all other articles. When writing about the Immigration Kristof seems to write at a the same grade level as he would on other subjects, on average. Kristof and Brooks both seem to match the overall by all others and seem to have a similar grade level when and when not talking about the subject Immigration.

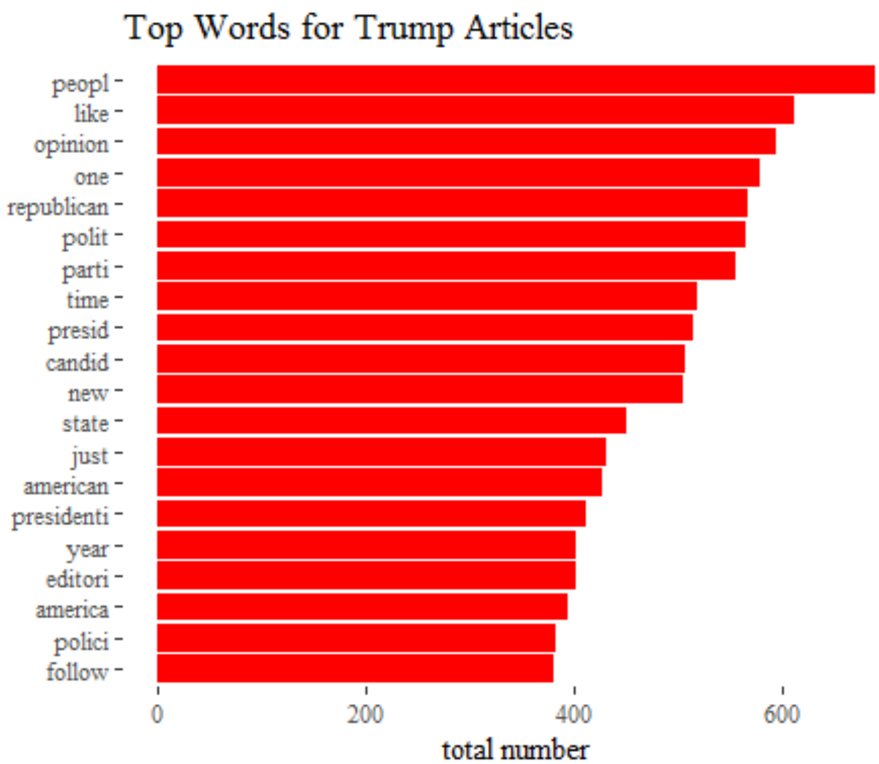


Maureen Dowd has an average Flech-Kincaid grade level of only about 9 for the subject Immigration and about 10 for all other articles. Maureen Dowd has a lower grade level compared to the other authors. Dowd also has a lower average grade level when writing about the subject Immigration in comparison to articles not on the subject.

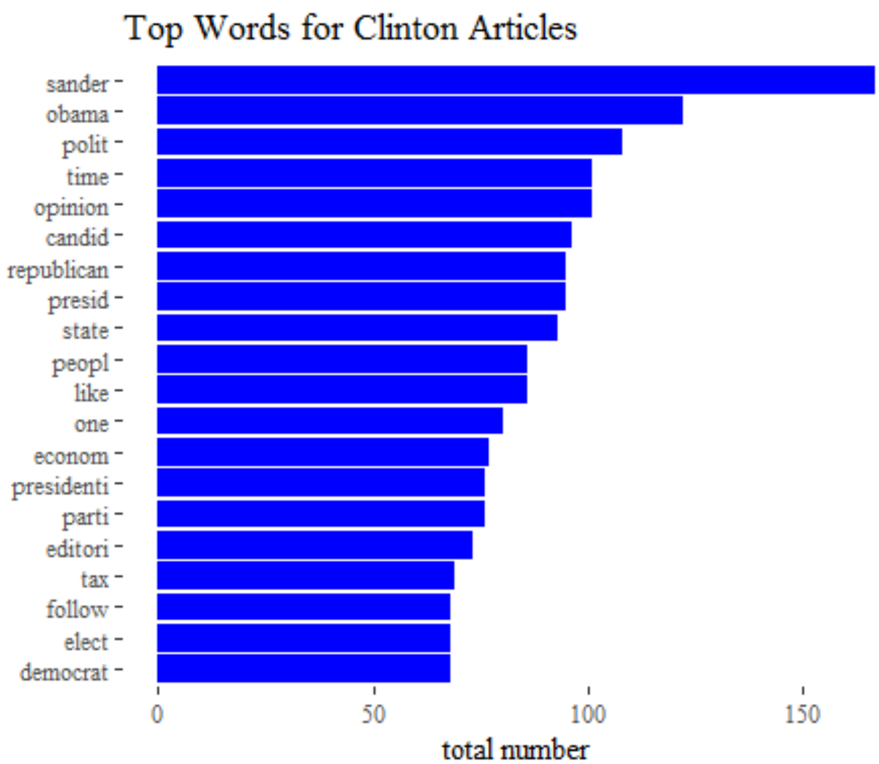


Thomas Friedman has an average Flech-Kincaid grade level of about 12 for the subject Immigration and only a score of a little over 11 for all other articles. When writing about the Immigration Friedman seems to write at a slightly higher grade level similar to the author Brooks.

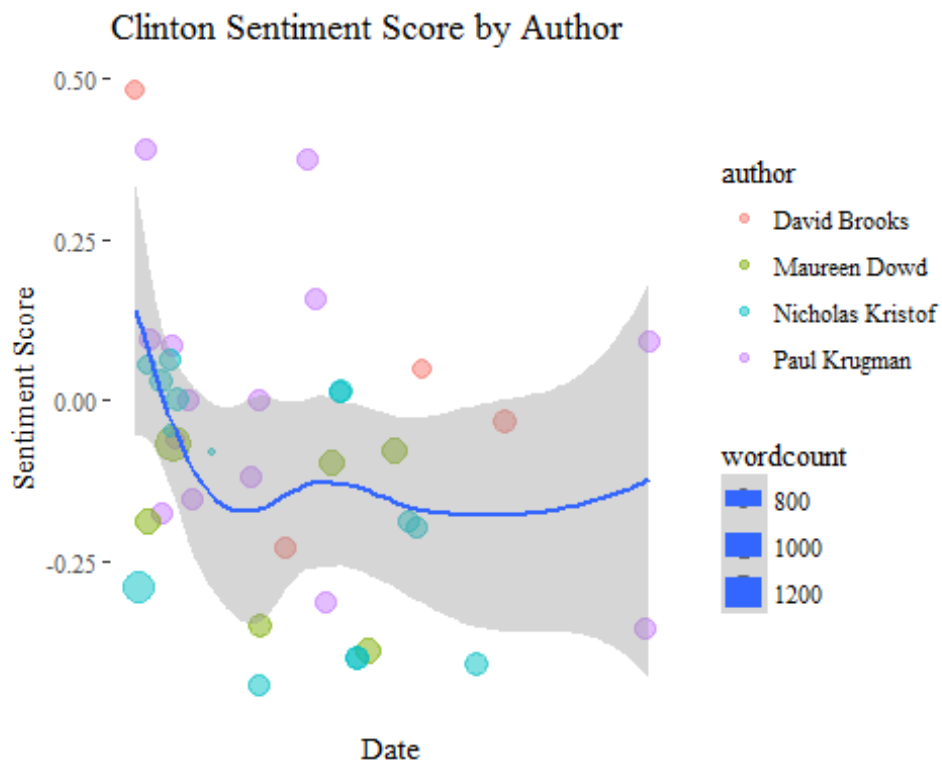
3



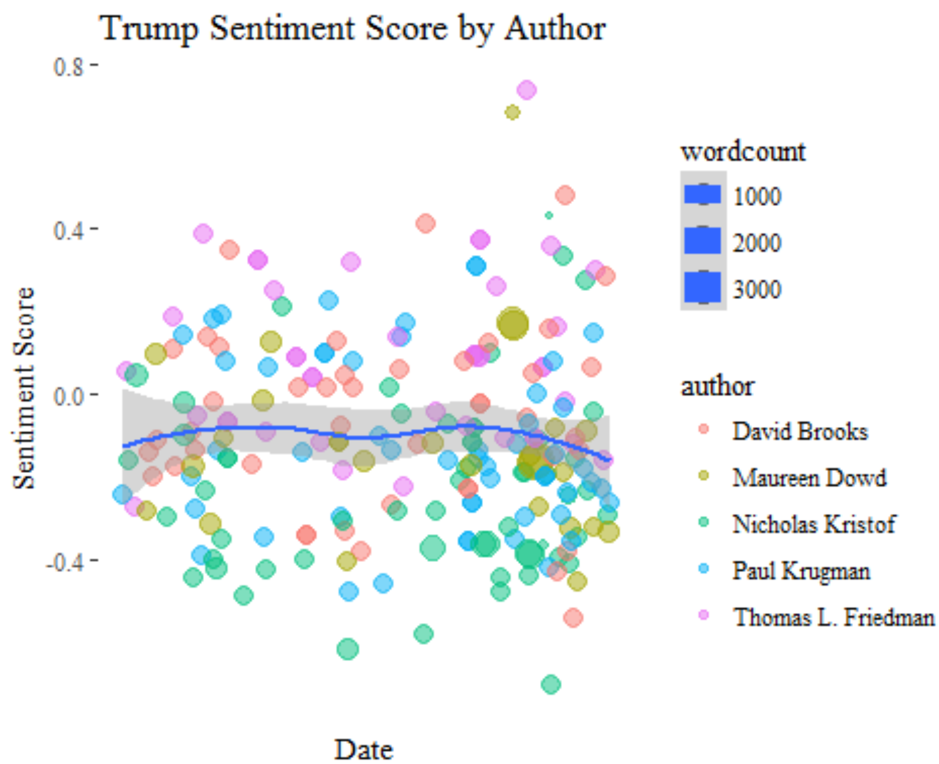
The graph above is a list of the top 20 words used in articles written about Trump that do not speak about Clinton. There are a total of 222 articles and the most common word used is people and it is used over 600 times. The second most used word is like which is surprising because of how many individuals do not agree with Trumps views and do not typically like him. But this could be due to the support Trump had prior to his Presidency when more individuals in the US did "like" Trump. Now lets take a look at the most common used words for articles about Clinton.



The graph above is a list of the top 20 words used in articles written about Clinton that do not speak about Trump. There are a total of 39 articles and the most common two words used are Sanders (over 150 times) and Obama (over 100 times). It appears that when writing about Clinton they often refer to Bernie Sanders and or Obama. This could be due to the fact that when writing about Clinton and her views they compare her views to that of Obama and Sanders or because they speak of the competition between Sanders and Clinton when they were running against one another. It is also worth noting that the term "republican" is in the top 10 words while the term "democrat" is the 20th word when Clinton is herself a Democrat and the term "democrat" doesn't appear in the top 20 for Trump.



The graph above is the sentiment score for articles about Hillary Clinton and not Trump over time. The sentiment analysis counted up all of the positive words in each article and gave each word a value of 1 and then added up all the negative words and gave them a value of -1. Then we took the positive values and minused the negative values, last we divide that by the positive values plus the negative values. This results in the sentiment score a positive score means the article was positive and a negative score means the article was negative. We see by the gray area around the blue line there is a large confidence interval because we have such a small number of articles (39). We can also notice that over time the sentiment for Clinton articles appears to steadily drop with the exception of a small hump where the sentiment scores raise. This actually happens at the time Hillary Clinton won the Democratic Nomination and could be the reason for the small hump. Clinton articles started with a positive sentiment but quickly drop to a negative sentiment and remain that way. We also see the author Friedman is not in the graph because he did not write just about Clinton in an article. The author Krugman writes about Clinton throughout the entire time frame while all other articles stopped writing about Clinton earlier.



The graph above is the sentiment score for articles about Donald Trump and not Clinton over time. We can notice that unlike Clinton articles Trump articles were always negative overtime and remain around the same overtime. It is also easy to notice that there are many more articles about Trump and the number of articles does not decrease over time but actually appears to increase. WE can see Thomas Friedman (Purple) who did not write about Clinton seems to have a number of positive sentiment reviews unlike the other remaining authors.

```
## [1] -0.0809698
```

After looking at the visualizations above on the sentiment scores I feel it is important to also view the average sentiment score for Clinton and Trump articles. Above is the mean sentiment score for all Clinton articles and below is the mean sentiment score for all Trump articles. We see that both have a negative sentiment but we see trump has a more negative average in comparison to Clinton. It is also important to keep in mind the fact there are only 39 Clinton articles in comparison to 222 Trump articles.

```
## [1] -0.1003955
```

Now Lets take a look at the average sentiment scores for Clinton and Trump articles for each individual authors in order to see how they compare to one another and between writing about Trump or Clinton. We will start with Trump articles.

```
## [1] -0.0598379
```

Above we see the mean sentiment score of Trump articles written by David Brooks. He appears to have a slightly less negative score than the overall mean sentiment score for all authors.

```
## [1] -0.2305594
```

Above is the mean sentiment score of Trump articles written by Nicholas Kristof.He appears to have a more negative score than the overall mean sentiment score for all authors.

```
## [1] -0.1252047
```

Above is the mean sentiment score of Trump articles written by Paul Krugman. He appears to have a mean sentiment score very similar to the overall mean sentiment score for all authors.

```
## [1] -0.1227132
```

Above is the mean sentiment score of Trump articles written by Maureen Dowd. She also appears to have a mean sentiment score very similar to the overall mean sentiment score for all authors just as Paul Krugman did. Dowd and Krugman may have very similar views of Trump.

```
## [1] 0.08380603
```

Lastly, above is the mean sentiment score of Trump articles written by Maureen Dowd. She also appears to have a mean sentiment score very similar to the overall mean sentiment score for all authors just as Paul Krugman did. Dowd and Krugman may have very similar views of Trump. Now lets take a look at the average sentiment score for Clinton articles for each author.

```
## [1] 0.06669739
```

Above we see the mean sentiment score of Clinton articles written by David Brooks. He appears to have a positive score in comparison to the negative overall mean sentiment score for all authors. David Brooks might have a positive view of Clinton.

```
## [1] -0.1517392
```

Above is the mean sentiment score of Clinton articles written by Nicholas Kristof. He appears to have a more negative score than the overall mean sentiment score for all authors and he also had a more negative score for Trump. It appears that Kristof uses more negative words than the other authors.

```
## [1] 0.001378833
```

Above we see the mean sentiment score of Clinton articles written by Paul Krugman. He appears to have a positive score in comparison to the negative overall mean sentiment score for all authors, just like David Brooks. Paul Krugman might also have a positive view of Clinton but his average sentiment score is very close to zero.

```
## [1] -0.1946379
```

Above is the mean sentiment score of Clinton articles written by Maureen Dowd. Se appears to have a more negative score than the overall mean sentiment score for all authors and she has the most negative score for Clinton potentially being the reason for the overall mean negative sentiment score for Clinton.

Project Plan

1

First I create a corpus using the text and meta data.

```
data_nyt <- corpus$documents
text <- data_nyt[c(1)]

df_source <- DataframeSource(text)

# Convert df_source to a corpus: df_corpus

df_corpus <- VCorpus(df_source)

# Examine df_corpus
df_corpus

## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 547
```

Next I use a funtion to clean my corpus and I also decided to remove some specific terms. I decided to remove these specific terms because they are unimportant and seem to show up very often.

```
new_stops <- c("nichola", "nickkristof", "kristof", "say", "can",
"will", stopwords("en"))


#clean text
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("en")))
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, removeWords, c("nicholas", "nickkristof",
```

```
"kristof", "say", "can", "will", "jan", "feb", "mar", "apr", "may",
"jun", "jul", "aug", "sep", "oct", "nov", "dec"))
  return(corpus)
}

# Apply customized function
nyt <- clean_corpus(df_corpus)

# example
```

```
nyt[[15]]$meta
```

```
##   author       : character(0)
##   datetimestamp: 2017-03-30 02:29:01
##   description  : character(0)
##   heading      : character(0)
##   id           : 15
##   language     : en
##   origin       : character(0)
```

Next I stem the corpus.

```
library(SnowballC)
# Stem all words
nyt_stemmed <- tm_map(nyt, stemDocument)
```

After steming the corpus I readd the meta data.

```
# Add meta data
meta(nyt_stemmed, type="local", tag="author") <- data_nyt$author
meta(nyt_stemmed, type="local", tag="subject") <- data_nyt$subject
meta(nyt_stemmed, type="local", tag="heading") <- data_nyt$heading
meta(nyt_stemmed, type="local", tag="person") <- data_nyt$person
meta(nyt_stemmed, type="local", tag="origin") <- data_nyt$geographic
meta(nyt_stemmed, type="local", tag="date") <- data_nyt$date
```

```
nyt_stemmed[[15]]$meta
```

```
##   author       : David Brooks
##   datetimestamp: 2017-03-30 02:29:01
##   description  : character(0)
##   heading      : A Little Reality on Immigration
##   id           : 15
##   language     : en
##   origin       : MEXICO (95%); UNITED STATES (94%); LATIN AMERICA
(90%); GUATEMALA (79%)
```

```
##   subject      : IMMIGRATION (95%); US REPUBLICAN PARTY (90%);
ILLEGAL IMMIGRANTS (90%); US PRESIDENTIAL CANDIDATES 2016 (90%); US
PRESIDENTIAL CANDIDATES 2012 (90%); EDITORIALS & OPINIONS (90%); PUBLIC
POLICY (89%); CRIME RATES (88%); BORDER CONTROL (78%); POLITICAL
PARTIES (78%); TERRITORIAL & NATIONAL BORDERS (78%); HISPANIC AMERICANS
(76%); RESEARCH INSTITUTES (70%); CRIMINAL OFFENSES (69%); VIOLENT
CRIME (69%); TERRORIST ATTACKS (66%); TERRORISM (64%); HIGH SCHOOLS
(62%); VIOLENT CRIME STATISTICS (61%); SUICIDE BOMBINGS (60%)
##   person       : DONALD TRUMP (92%); RONALD REAGAN (89%)
##   date         : 2016-02-19
```

Below I create a dtm, dtm matrix, tdm, and tdm matrix just incase I wanted/needed to use them.

```
# Create the dtm from the corpus
nyt_dtm <- DocumentTermMatrix(nyt_stemmed)

# Print out nyt_dtm data
nyt_dtm

## <<DocumentTermMatrix (documents: 547, terms: 15505)>>
## Non-/sparse entries: 183752/8297483
## Sparsity           : 98%
## Maximal term length: 44
## Weighting          : term frequency (tf)

# Convert nyt_dtm to a matrix
nyt_mD <- as.matrix(nyt_dtm)

# Print the dimensions of nyt_m
dim(nyt_mD)

## [1]   547 15505

# Create a TDM
nyt_tdm <- TermDocumentMatrix(nyt_stemmed)

# Print tdm data
nyt_tdm

## <<TermDocumentMatrix (terms: 15505, documents: 547)>>
## Non-/sparse entries: 183752/8297483
## Sparsity           : 98%
## Maximal term length: 44
## Weighting          : term frequency (tf)

# Convert nyt_tdm to a matrix
nyt_mT <- as.matrix(nyt_tdm)

# Print the dimensions of the matrix
dim(nyt_mT)
```

```
## [1] 15505   547
```

Next I tidy the tdm into a dataframe and tidy the stemmed corpus into a dataframe. I do this in order to join the two into one dataframe for analysis.

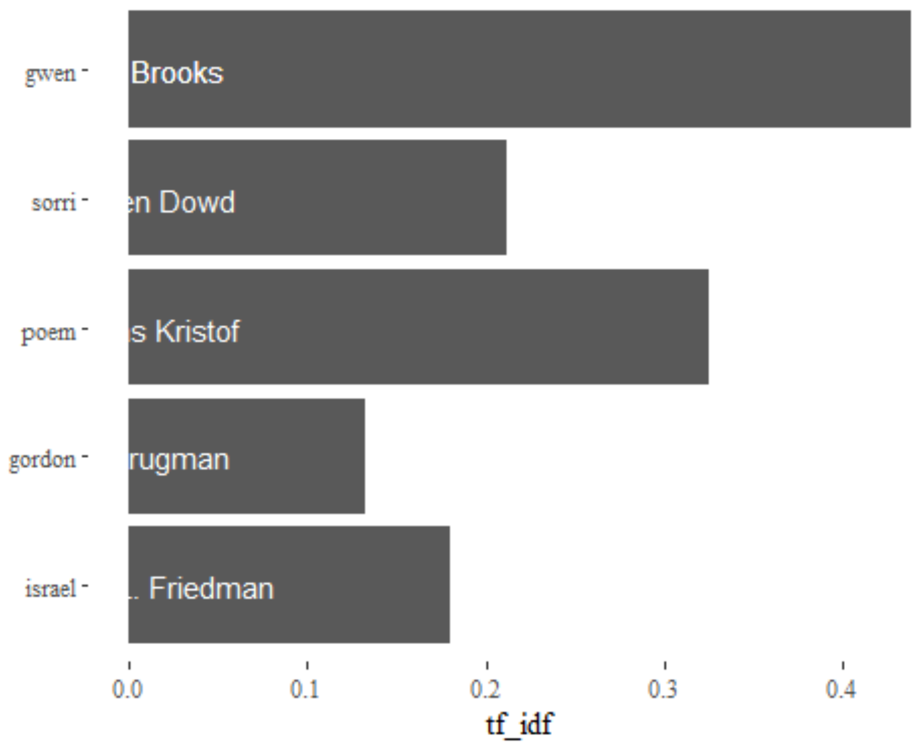
```
# tidy for ggplot
library(dplyr)
library(tidytext)


nyt_td <- tidy(nyt_stemmed)


author <- nyt_td$author

meta(nyt_stemmed, "author", type = "local") <- author

nyt_td2 <- tidy(nyt_tdm)

nyt_td$document <- row.names(nyt_td)
nyt_td3 <- right_join(nyt_td, nyt_td2, by = "document")
```

I take the new dataframe and make a tf-idf to run analysis using the tf-idf and the term frequencey becuase one might be more beneficial then the other for this analysis.

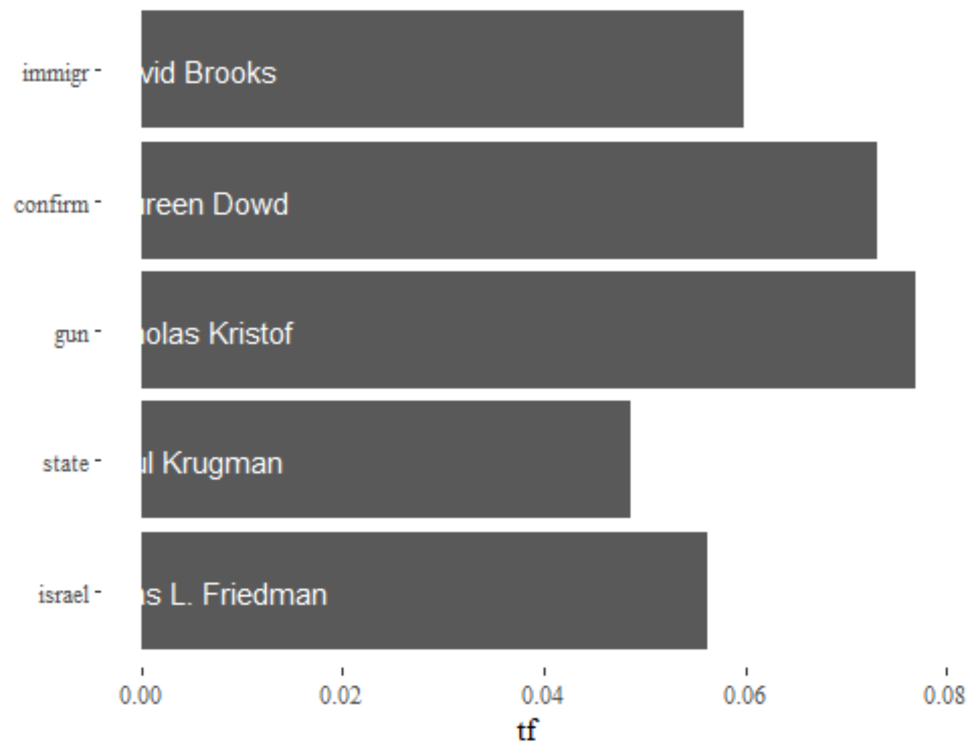
```
nyt_tf_idf <-  nyt_td3 %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))
```

Next I create a graph to look at the most common term per author according ot the tf-idf and the tf. I included both in the polished graphs because I felt it was important to visualize the difference between the two. I used a different color for each to easily differentiate them.

```
library(ggplot2)
library(ggthemes)

nyt_tf_idf %>% group_by(author) %>%
                top_n(n = 1, wt = tf_idf)  %>%
ggplot(aes(x = reorder(term, desc(author)), y = tf_idf)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label=author, x=term, y=0.005), color="white") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```




```
nyt_tf_idf %>% group_by(author) %>%
                top_n(n = 1, wt = tf)  %>%
ggplot(aes(x = reorder(term, desc(author)), y = tf)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label=author, x=term, y=0.005), color="white") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```

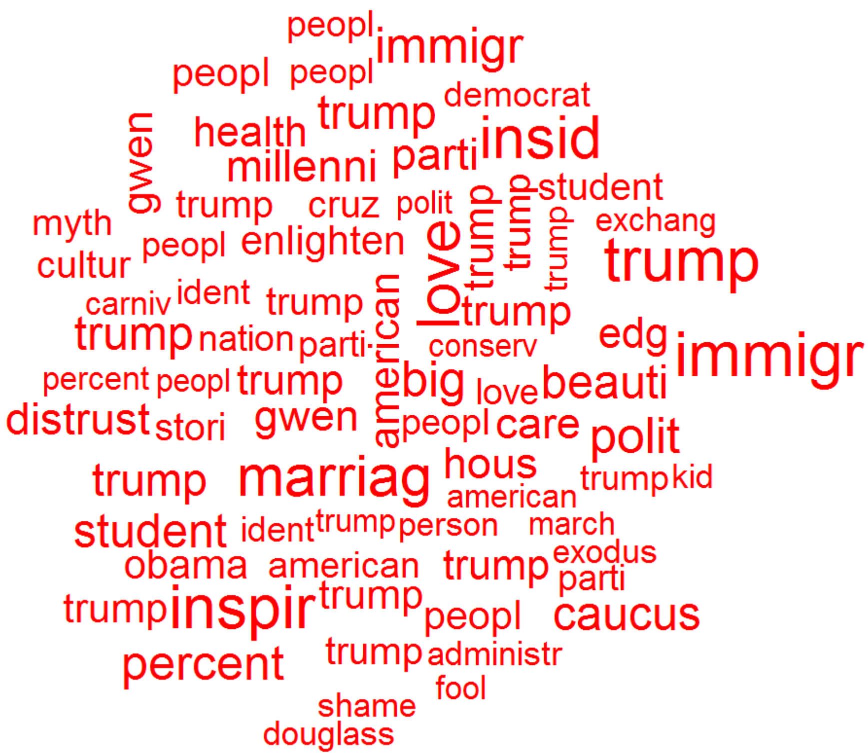


Next I create a dataframe for each individual author and then create a corpus for each and clean and stem said corpuses. I also create a tdm and a tf-idf for each author in order to create word clouds for each. After all of this the wordclouds had repeated terms because they were not by author but the words would print by article, this was problematic so I had to figure out how to solve this.

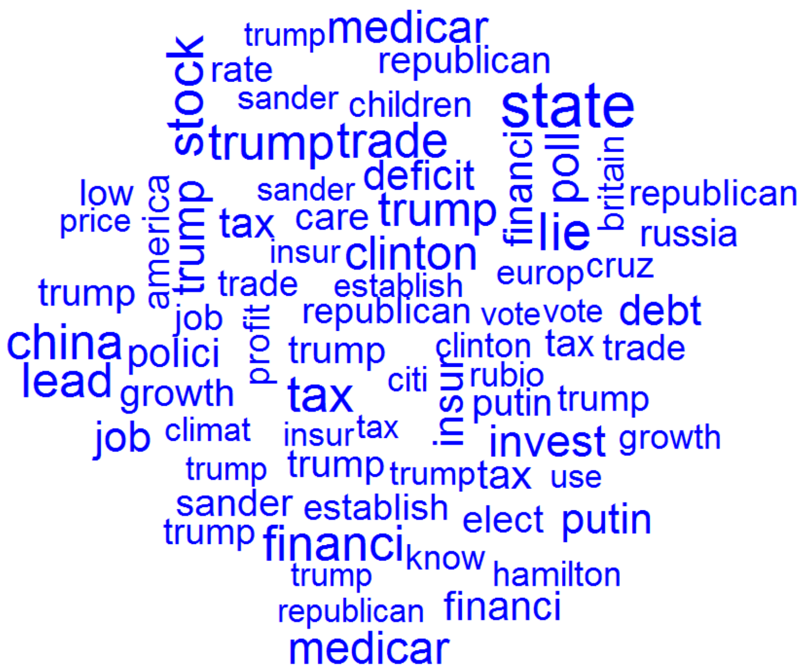
```
library(wordcloud)

db <- filter(text, author == "David Brooks")
nk <- filter(text, author == "Nicholas Kristof")
pk <- filter(text, author == "Paul Krugman")
md <- filter(text, author == "Maureen Dowd")
tf <- filter(text, author == "Thomas L. Friedman")

df_sourcedb <- DataframeSource(db)
df_corpusdb <- VCorpus(df_sourcedb)
df_sourcenk <- DataframeSource(nk)
df_corpusnk <- VCorpus(df_sourcenk)
df_sourcepk <- DataframeSource(pk)
df_corpuspk <- VCorpus(df_sourcepk)
df_sourcemd <- DataframeSource(md)
df_corpusmd <- VCorpus(df_sourcemd)
df_sourcetf <- DataframeSource(tf)
df_corpustf <- VCorpus(df_sourcetf)

cdb <- clean_corpus(df_corpusdb)
```

```r
cnk <- clean_corpus(df_corpusnk)
cpk <- clean_corpus(df_corpuspk)
cmd <- clean_corpus(df_corpusmd)
ctf <- clean_corpus(df_corpustf)

db_stemmed <- tm_map(cdb, stemDocument)
nk_stemmed <- tm_map(cnk, stemDocument)
pk_stemmed <- tm_map(cpk, stemDocument)
md_stemmed <- tm_map(cmd, stemDocument)
tf_stemmed <- tm_map(ctf, stemDocument)


db_tdm <- TermDocumentMatrix(db_stemmed)
nk_tdm <- TermDocumentMatrix(nk_stemmed)
pk_tdm <- TermDocumentMatrix(pk_stemmed)
md_tdm <- TermDocumentMatrix(md_stemmed)
tf_tdm <- TermDocumentMatrix(tf_stemmed)

db_td <- tidy(db_tdm)
nk_td <- tidy(nk_tdm)
pk_td <- tidy(pk_tdm)
md_td <- tidy(md_tdm)
tf_td <- tidy(tf_tdm)


db_tf_idf <-  db_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))


nk_tf_idf <-  nk_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))

pk_tf_idf <-  pk_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))

md_tf_idf <-  md_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))

tf_tf_idf <-  tf_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))

# Set seed - to make your word cloud reproducible
set.seed(1234)
```

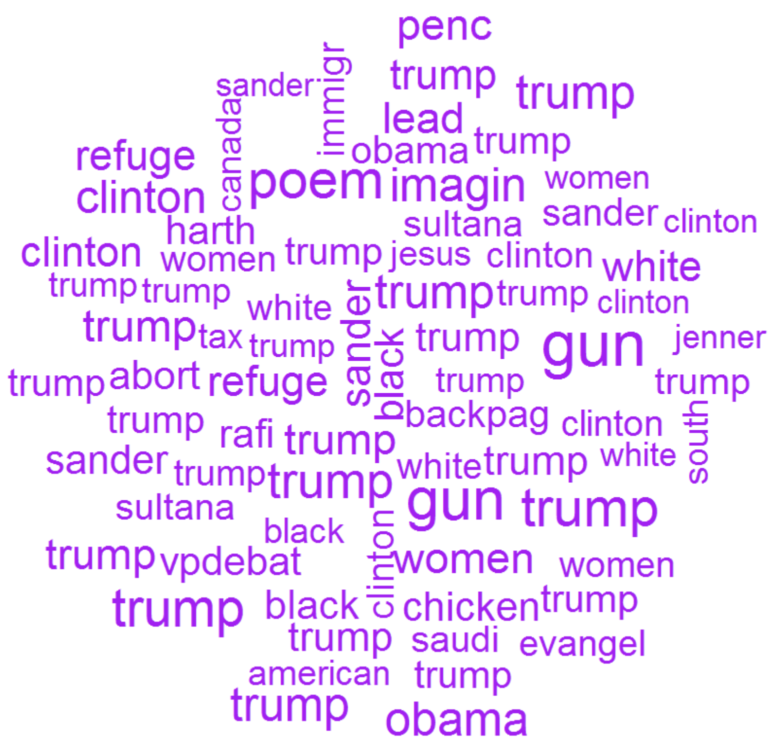
```r
# Create a wordcloud for the values in word_freqs
wordcloud(db_tf_idf$term, db_tf_idf$tf,
          max.words = 75, colors = "red")
```



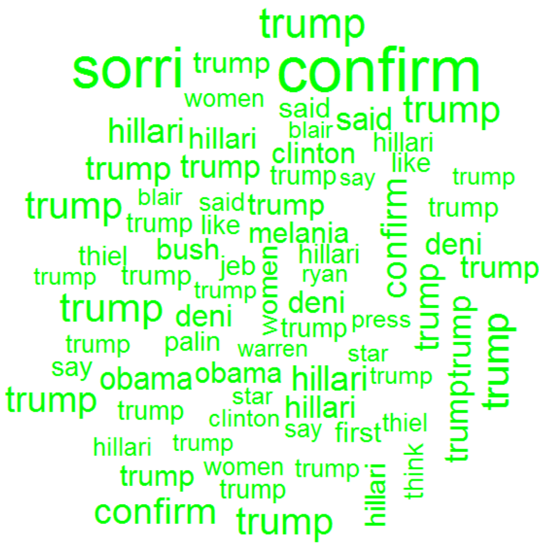

```r
wordcloud(pk_tf_idf$term, pk_tf_idf$tf,
          max.words = 75, colors = "blue")
```



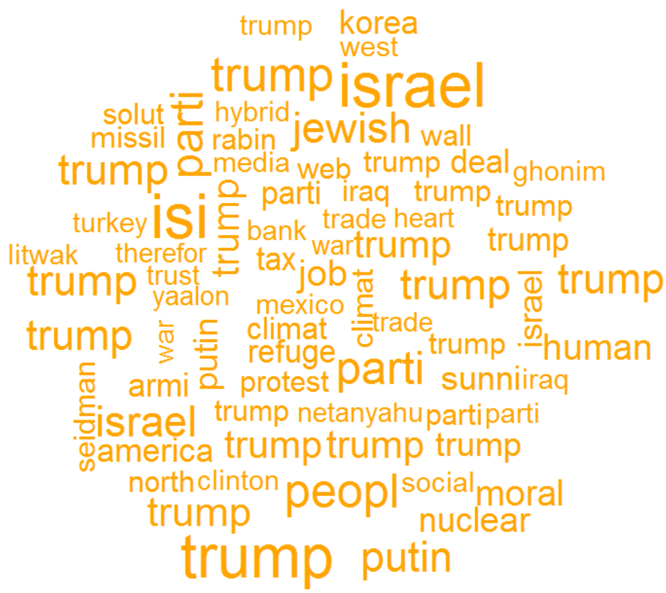

```r
wordcloud(nk_tf_idf$term, nk_tf_idf$tf,
          max.words = 75, colors = "purple")
```




```r
wordcloud(md_tf_idf$term, md_tf_idf$tf,
          max.words = 75, colors = "green")
```




```
wordcloud(tf_tf_idf$term, tf_tf_idf$tf,
          max.words = 75, colors = "orange")
```

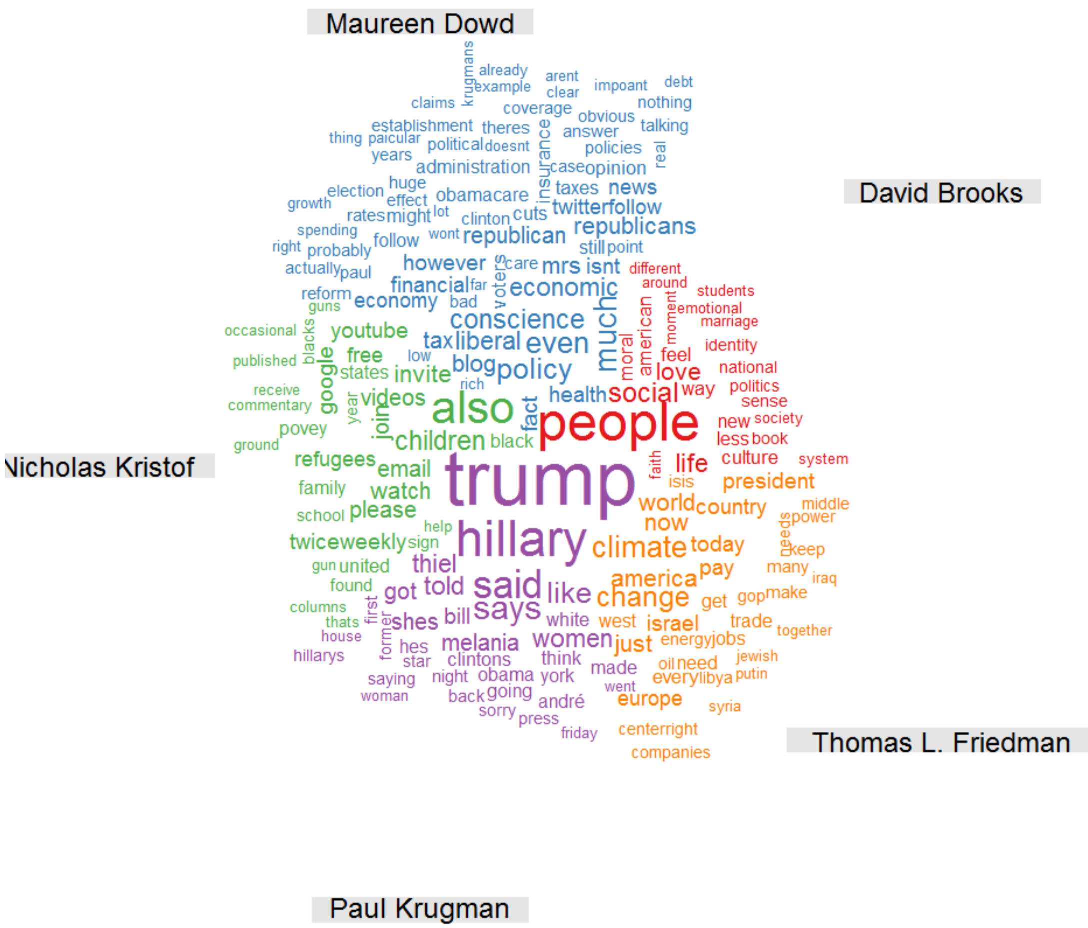


Now I create a vector including a vector with all authors separated then create a corpus and then a tdm of said corpus. I also separate each author into columns. Finally I create a comparison word cloud to compare each author, this was included in the final plots because it seemed informative and took alot of time/effort. I enlarged the plot to fit all words I wanted and also used the clor spectrum I used because I felt it was easy to differentiate the authors. I also added a title for each author to identify them.

```
# Wordcloud by author

library(wordcloud)

dbt <- data_nyt$texts[data_nyt$author == "David Brooks"]
pkt <- data_nyt$texts[data_nyt$author == "Paul Krugman"]
nkt <- data_nyt$texts[data_nyt$author == "Nicholas Kristof"]
mdt <- data_nyt$texts[data_nyt$author == "Maureen Dowd"]
```

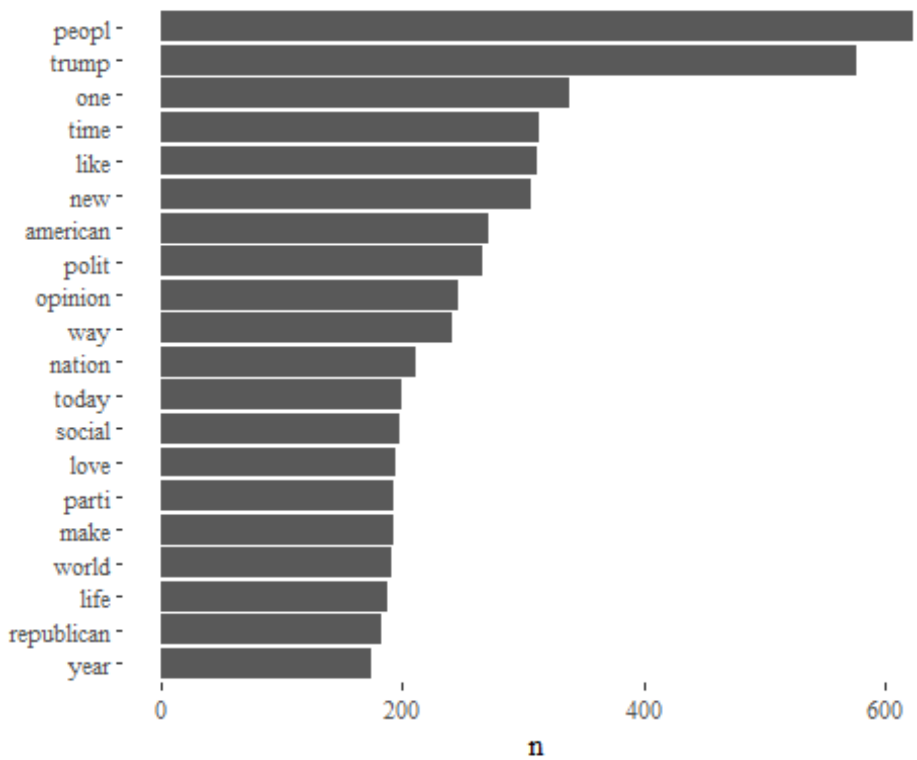
```r
tft <- data_nyt$texts[data_nyt$author == "Thomas L. Friedman"]

clean_byauthor <- function(x)
{
  x = tolower(x)
  x = gsub("rt", "", x)
  x = gsub("@\\w+", "", x)
  x = gsub("[[:punct:]]", "", x)
  x = gsub("[[:digit:]]", "", x)
  x = gsub("http\\w+", "", x)
  x = gsub("[ |\t]{2,}", "", x)
  x = gsub("^ ", "", x)
  x = gsub(" $", "", x)
  return(x)
}

dbt_clean <- clean_byauthor(dbt)
pkt_clean <- clean_byauthor(pkt)
nkt_clean <- clean_byauthor(nkt)
mdt_clean <- clean_byauthor(mdt)
tft_clean <- clean_byauthor(tft)

dbt <- paste(dbt_clean, collapse=" ")
pkt <- paste(pkt_clean, collapse=" ")
nkt <- paste(nkt_clean, collapse=" ")
mdt <- paste(mdt_clean, collapse=" ")
tft <- paste(tft_clean, collapse=" ")

# create a single vector
author_vector <- c(dbt, pkt, nkt, mdt, tft)

author_vector <- removeWords(author_vector,c(stopwords("english"),
"Paul Krugman", "David Brooks", "Thomas L. Friedman", "Maureen Dowd",
"Nicholas Kristof", "nickkristof", "say", "can", "will", "jan", "feb",
"mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"))

# create corpus
author_corp <- Corpus(VectorSource(author_vector))

# create term-document matrix
author_tdm <- TermDocumentMatrix(author_corp)

# convert as matrix
author_tdm <- as.matrix(author_tdm)

# add column names
colnames(author_tdm) <- c("David Brooks", "Maureen Dowd", "Nicholas
Kristof", "Paul Krugman", "Thomas L. Friedman")
```

```r
comparison.cloud(author_tdm, random.order=FALSE,
colors = brewer.pal(8, "Set1"),
title.size=1.5, max.words = 200)
```

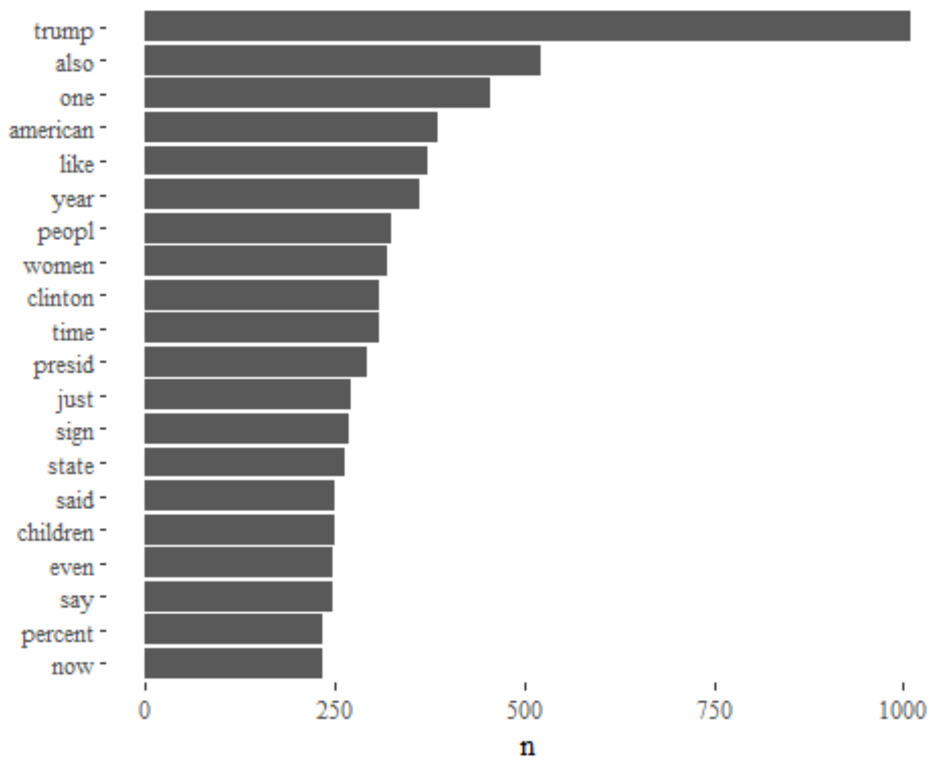


I also include the top 20 terms by count for each author in the fina plots becuase I believe it is more clear then the word cloud. I used a different color for each becuase it makes it easier to notice they are all different by author. I decided not to join them all as one plot because the word cloud was already very busy and I felt it was better to keep them seperate and not overwhelm the reader.

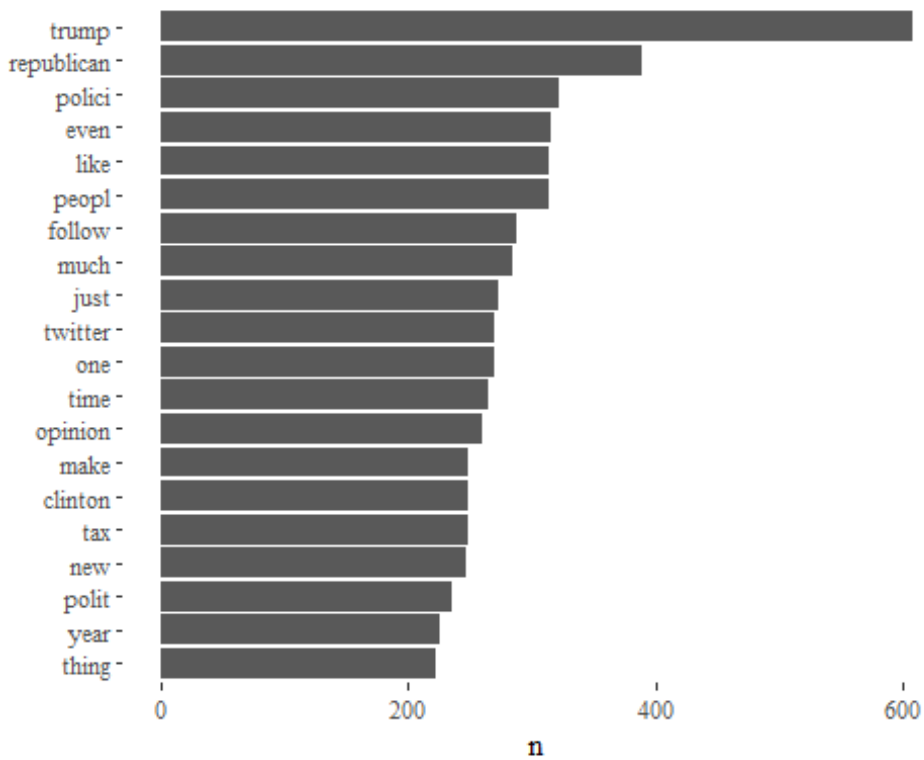
```r
db_td %>%      group_by(term) %>%
                 summarise(n = sum(count)) %>%
                 top_n(n = 20, wt = n)  %>%
                 mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```



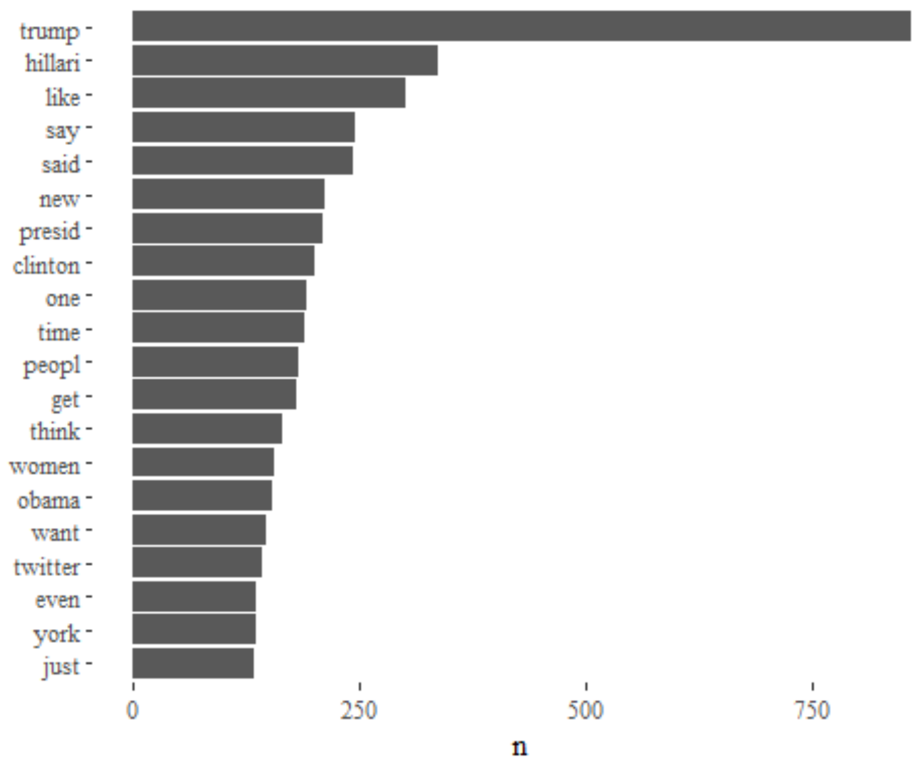

```
nk_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```



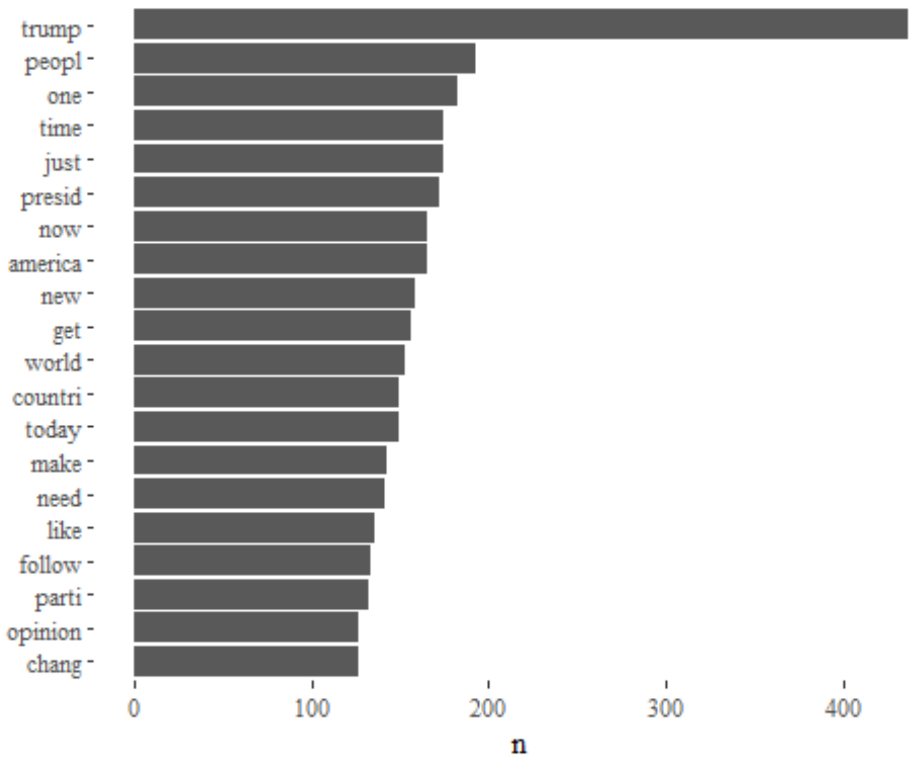

```r
pk_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```




```r
md_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```

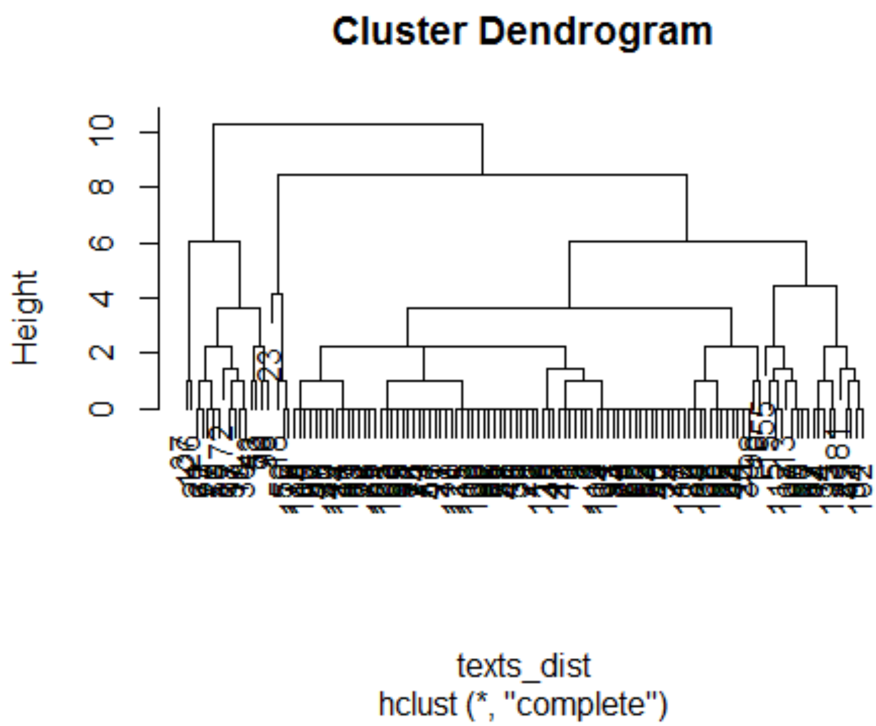



```
tf_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```

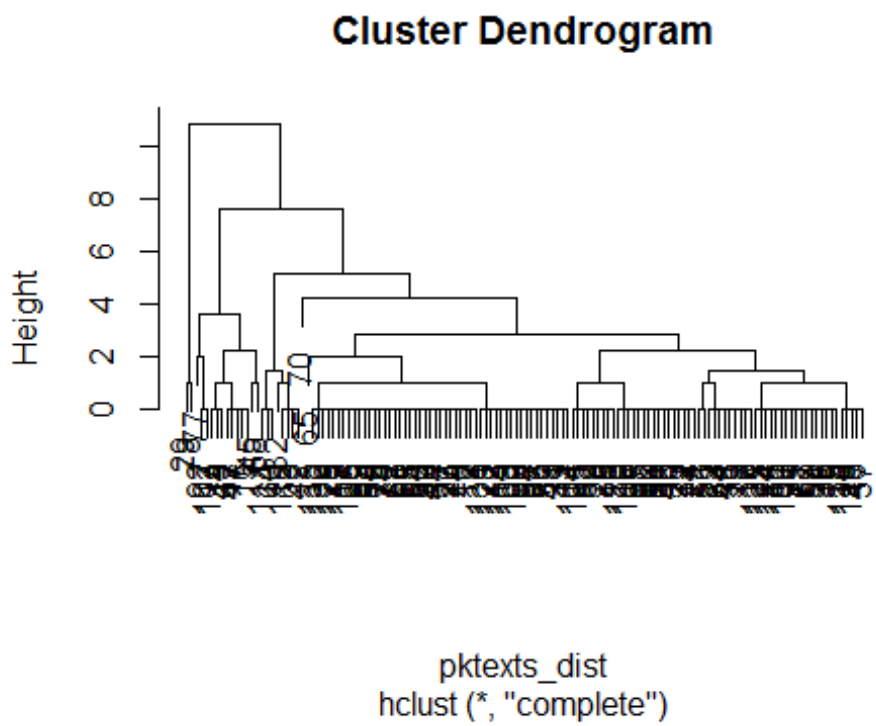


I also included the dendrograms for each author because I felt it was a nice way to look at the differences between authors in a more abstract way. I added a title for each to show what author we are analyzing. I again kept the graphs separate to make it more visually pleasing and easier to interpret. I was unable to included a dendrogram of the author Kristof because it would not run and I was unable to figure out the issue. The code is included below but hashed out inorder to knit the file.

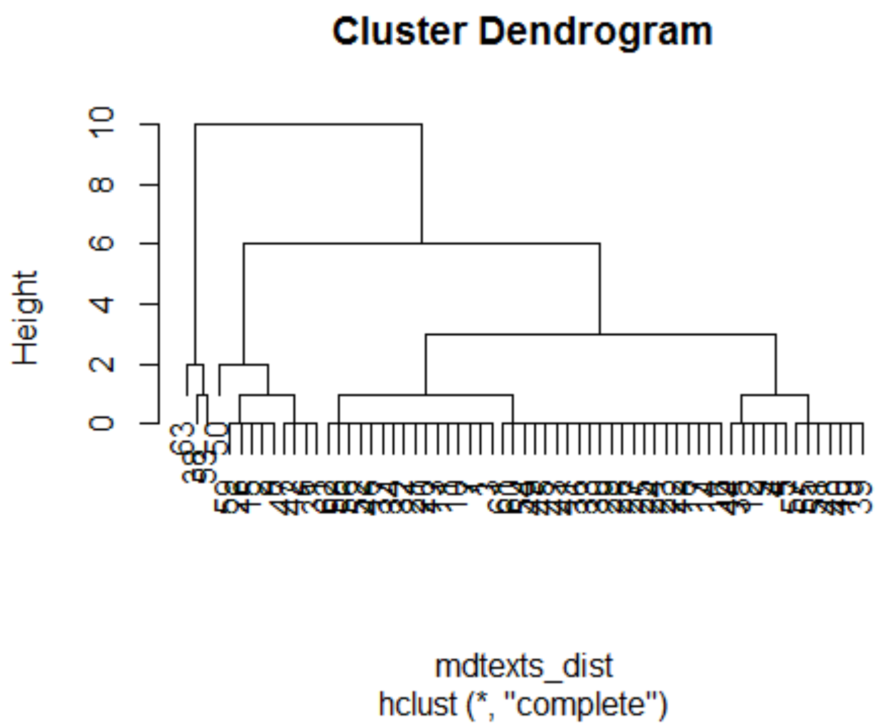
```
# dendograms by author

db_dtm <- DocumentTermMatrix(db_stemmed)

dtm1 <- removeSparseTerms(db_dtm, sparse = 0.01)
   # Remove most sparse terms
dtm_m <- as.matrix(dtm1) # Create tdm_m
dtm_df <- as.data.frame(dtm_m) # Create tdm_df
texts_dist <- dist(dtm_df) # Create texts_dist
hc <- hclust(texts_dist) # Create hc

# Plot the dendrogram
plot(hc)
```



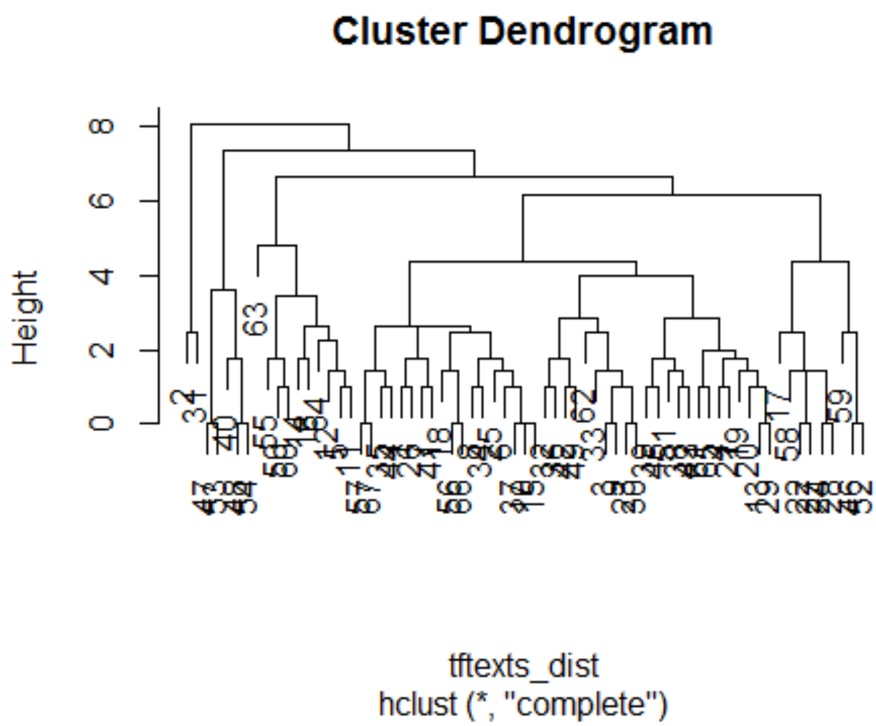

```
pk_dtm <- DocumentTermMatrix(pk_stemmed)

pkdtm1 <- removeSparseTerms(pk_dtm, sparse = 0.01)
   # Remove most sparse terms
pkdtm_m <- as.matrix(pkdtm1) # Create tdm_m
pkdtm_df <- as.data.frame(pkdtm_m) # Create tdm_df
pktexts_dist <- dist(pkdtm_df) # Create texts_dist
pkhc <- hclust(pktexts_dist) # Create hc

# Plot the dendrogram
plot(pkhc)
```




```
nk_dtm <- DocumentTermMatrix(nk_stemmed)

nkdtm1 <- removeSparseTerms(nk_dtm, sparse = 0.01)
   # Remove most sparse terms
nkdtm_m <- as.matrix(nkdtm1) # Create tdm_m
nkdtm_df <- as.data.frame(nkdtm_m) # Create tdm_df
nktexts_dist <- dist(nkdtm_df) # Create texts_dist
#nkhc <- hclust(nktexts_dist) # Create hc

# Plot the dendrogram
#plot(nkhc)


md_dtm <- DocumentTermMatrix(md_stemmed)

mddtm1 <- removeSparseTerms(md_dtm, sparse = 0.01)
   # Remove most sparse terms
mddtm_m <- as.matrix(mddtm1) # Create tdm_m
mddtm_df <- as.data.frame(mddtm_m) # Create tdm_df
mdtexts_dist <- dist(mddtm_df) # Create texts_dist
mdhc <- hclust(mdtexts_dist) # Create hc

# Plot the dendrogram
plot(mdhc)
```




```
tf_dtm <- DocumentTermMatrix(tf_stemmed)

tfdtm1 <- removeSparseTerms(tf_dtm, sparse = 0.01)
   # Remove most sparse terms
tfdtm_m <- as.matrix(tfdtm1) # Create tdm_m
tfdtm_df <- as.data.frame(tfdtm_m) # Create tdm_df
tftexts_dist <- dist(tfdtm_df) # Create texts_dist
tfhc <- hclust(tftexts_dist) # Create hc

# Plot the dendrogram
plot(tfhc)
```



I did the bonus! For the bonus part of question one I calculated and printed the top 20 subjects of each individual author. I felt this was the clearest way to examine the differences between the authors because there are so many subjects and they also overlap (used in same article).

1 bonus

```
#seperate out subjects and find major topics by author
meta(df_corpusdb, type="local", tag="subject") <- data_nyt$subject

db_corpus <- corpus(df_corpusdb)


sub_db <- gsub( " *\\(.*?\\) *", "", db_corpus$documents$subject) #
Remove parentheses
sub_db <- strsplit(sub_db, ";")  # Split by ';' into a list
sub_db <- lapply(sub_db, FUN=trimws) # Remove whitespace
sub_dblist <- unique(unlist(sub_db)) # Make into a list, remove
whitespace
top10sub_db <- rownames(sort(table(unlist(sub_db)),
decreasing=TRUE)[2:21])
sub_db <- lapply(sub_db, FUN=
function(A,B){top10sub_db[match(A,top10sub_db)]})
sub_db <- lapply(sub_db, function(x) x[!is.na(x)])
sub_db1 <- tidy(top10sub_db)
library(reshape)
```

```
David_Brooks_top_20_subjects <- rename(sub_db1, c(x="Top 20 subjects
David Brooks"))

David_Brooks_top_20_subjects

## # A tibble: 20 × 1
##     `Top 20 subjects David Brooks`
##                              <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2              US REPUBLICAN PARTY
## 3                         POLITICS
## 4             CAMPAIGNS & ELECTIONS
## 5  US PRESIDENTIAL CANDIDATES 2012
## 6                 POLITICAL PARTIES
## 7                     US PRESIDENTS
## 8                      CONSERVATISM
## 9         US PRESIDENTIAL ELECTIONS
## 10                        ELECTIONS
## 11                         RELIGION
## 12 US PRESIDENTIAL CANDIDATES 2008
## 13             US DEMOCRATIC PARTY
## 14   HEADS OF GOVERNMENT ELECTIONS
## 15               POLITICAL DEBATES
## 16             POLITICAL CANDIDATES
## 17                  MUSLIMS & ISLAM
## 18                 TAXES & TAXATION
## 19          INTERNATIONAL RELATIONS
## 20                       LIBERALISM

#seperate out subjects and find major topics by author
meta(df_corpuspk, type="local", tag="subject") <- data_nyt$subject

pk_corpus <- corpus(df_corpuspk)


sub_pk <- gsub( " *\\(.*?\\) *", "", pk_corpus$documents$subject) #
Remove parentheses
sub_pk <- strsplit(sub_pk, ";")  # Split by ';' into a list
sub_pk <- lapply(sub_pk, FUN=trimws) # Remove whitespace
sub_pklist <- unique(unlist(sub_pk)) # Make into a list, remove
whitespace
top10sub_pk <- rownames(sort(table(unlist(sub_pk)),
decreasing=TRUE)[2:21])
sub_pk <- lapply(sub_pk, FUN=
function(A,B){top10sub_pk[match(A,top10sub_pk)]})
sub_pk <- lapply(sub_pk, function(x) x[!is.na(x)])
sub_pk1 <- tidy(top10sub_pk)
library(reshape)
Paul_Krugman_top_20_subjects <- rename(sub_pk1, c(x=" top 20 subjects
Paul Krugman"))
```

```
Paul_Krugman_top_20_subjects

## # A tibble: 20 × 1
##    ` top 20 subjects Paul Krugman`
##                               <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2               US REPUBLICAN PARTY
## 3                          POLITICS
## 4             CAMPAIGNS & ELECTIONS
## 5  US PRESIDENTIAL CANDIDATES 2012
## 6                 POLITICAL PARTIES
## 7                     US PRESIDENTS
## 8                      CONSERVATISM
## 9                          RELIGION
## 10       US PRESIDENTIAL ELECTIONS
## 11                        ELECTIONS
## 12 US PRESIDENTIAL CANDIDATES 2008
## 13              US DEMOCRATIC PARTY
## 14   HEADS OF GOVERNMENT ELECTIONS
## 15             POLITICAL CANDIDATES
## 16                POLITICAL DEBATES
## 17                  MUSLIMS & ISLAM
## 18                       LIBERALISM
## 19                  TAXES & TAXATION
## 20          INTERNATIONAL RELATIONS

#seperate out subjects and find major topics by author
meta(df_corpusnk, type="local", tag="subject") <- data_nyt$subject

nk_corpus <- corpus(df_corpusnk)


sub_nk <- gsub( " *\\(.*?\\) *", "", nk_corpus$documents$subject) #
Remove parentheses
sub_nk <- strsplit(sub_nk, ";")  # Split by ';' into a list
sub_nk <- lapply(sub_nk, FUN=trimws) # Remove whitespace
sub_nklist <- unique(unlist(sub_nk)) # Make into a list, remove
whitespace
top10sub_nk <- rownames(sort(table(unlist(sub_nk)),
decreasing=TRUE)[2:21])
sub_nk <- lapply(sub_nk, FUN=
function(A,B){top10sub_nk[match(A,top10sub_nk)]})
sub_nk <- lapply(sub_nk, function(x) x[!is.na(x)])
sub_nk1 <- tidy(top10sub_nk)
library(reshape)
Nicholas_Kristof_top_20_subjects <- rename(sub_nk1, c(x="Top 20
subjects Nicholas Kristof"))

Nicholas_Kristof_top_20_subjects
```

```
## # A tibble: 20 × 1
##    `Top 20 subjects Nicholas Kristof`
##                                 <chr>
## 1      US PRESIDENTIAL CANDIDATES 2016
## 2                  US REPUBLICAN PARTY
## 3                             POLITICS
## 4                        US PRESIDENTS
## 5      US PRESIDENTIAL CANDIDATES 2012
## 6                    POLITICAL PARTIES
## 7                CAMPAIGNS & ELECTIONS
## 8                             RELIGION
## 9                        CONSERVATISM
## 10     US PRESIDENTIAL CANDIDATES 2008
## 11                           ELECTIONS
## 12           US PRESIDENTIAL ELECTIONS
## 13                 US DEMOCRATIC PARTY
## 14       HEADS OF GOVERNMENT ELECTIONS
## 15                   POLITICAL DEBATES
## 16                     MUSLIMS & ISLAM
## 17               POLITICAL CANDIDATES
## 18                            CHILDREN
## 19                          LIBERALISM
## 20             INTERNATIONAL RELATIONS

#seperate out subjects and find major topics by author
meta(df_corpusmd, type="local", tag="subject") <- data_nyt$subject

md_corpus <- corpus(df_corpusmd)


sub_md <- gsub( " *\\(.*?\\) *", "", md_corpus$documents$subject) #
Remove parentheses
sub_md <- strsplit(sub_md, ";")  # Split by ';' into a list
sub_md <- lapply(sub_md, FUN=trimws) # Remove whitespace
sub_mdlist <- unique(unlist(sub_md)) # Make into a list, remove
whitespace
top10sub_md <- rownames(sort(table(unlist(sub_md)),
decreasing=TRUE)[2:21])
sub_md <- lapply(sub_md, FUN=
function(A,B){top10sub_md[match(A,top10sub_md)]})
sub_md <- lapply(sub_md, function(x) x[!is.na(x)])
sub_md1 <- tidy(top10sub_md)
library(reshape)
Maureen_Dowd_top_20_subjects <- rename(sub_md1, c(x="Top 20 subjects
Maureen Dowd"))

Maureen_Dowd_top_20_subjects

## # A tibble: 20 × 1
##     `Top 20 subjects Maureen Dowd`
```

```
##                                 <chr>
## 1  US PRESIDENTIAL CANDIDATES 2016
## 2              US REPUBLICAN PARTY
## 3                POLITICAL PARTIES
## 4                         POLITICS
## 5            CAMPAIGNS & ELECTIONS
## 6                     CONSERVATISM
## 7  US PRESIDENTIAL CANDIDATES 2012
## 8                    US PRESIDENTS
## 9                         RELIGION
## 10                TAXES & TAXATION
## 11                   ECONOMIC NEWS
## 12                       ELECTIONS
## 13         INTERNATIONAL RELATIONS
## 14             US DEMOCRATIC PARTY
## 15                  WEALTHY PEOPLE
## 16                  FOREIGN POLICY
## 17                         TAX LAW
## 18                     IMMIGRATION
## 19                      LIBERALISM
## 20                 MUSLIMS & ISLAM
```

```
#seperate out subjects and find major topics by author
meta(df_corpustf, type="local", tag="subject") <- data_nyt$subject

tf_corpus <- corpus(df_corpustf)


sub_tf <- gsub( " *\\(.*?\\) *", "", tf_corpus$documents$subject) #
Remove parentheses
sub_tf <- strsplit(sub_tf, ";")  # Split by ';' into a list
sub_tf <- lapply(sub_tf, FUN=trimws) # Remove whitespace
sub_tflist <- unique(unlist(sub_tf)) # Make into a list, remove
whitespace
top10sub_tf <- rownames(sort(table(unlist(sub_tf)),
decreasing=TRUE)[2:21])
sub_tf <- lapply(sub_tf, FUN=
function(A,B){top10sub_tf[match(A,top10sub_tf)]})
sub_tf <- lapply(sub_tf, function(x) x[!is.na(x)])
sub_tf1 <- tidy(top10sub_tf)
library(reshape)
Thomas_Friedman_top_20_subjects <- rename(sub_tf1, c(x="Top 20 subjects
Thomas L. Friedman"))

Thomas_Friedman_top_20_subjects
```

```
## # A tibble: 20 × 1
##    `Top 20 subjects Thomas L. Friedman`
##                                   <chr>
## 1       US PRESIDENTIAL CANDIDATES 2016
```

```
## 2                      US REPUBLICAN PARTY
## 3                        POLITICAL PARTIES
## 4                                 POLITICS
## 5                     CAMPAIGNS & ELECTIONS
## 6                             CONSERVATISM
## 7           US PRESIDENTIAL CANDIDATES 2012
## 8                            US PRESIDENTS
## 9                        TAXES & TAXATION
## 10                                RELIGION
## 11                 INTERNATIONAL RELATIONS
## 12                     US DEMOCRATIC PARTY
## 13                           ECONOMIC NEWS
## 14                               ELECTIONS
## 15          US PRESIDENTIAL CANDIDATES 2008
## 16                          WEALTHY PEOPLE
## 17                          FOREIGN POLICY
## 18                                 TAX LAW
## 19                             IMMIGRATION
## 20                              LIBERALISM
```

2

For number 2 first I convert to a quanteda corpus in order to calculate the Flesch-Kincaid score.

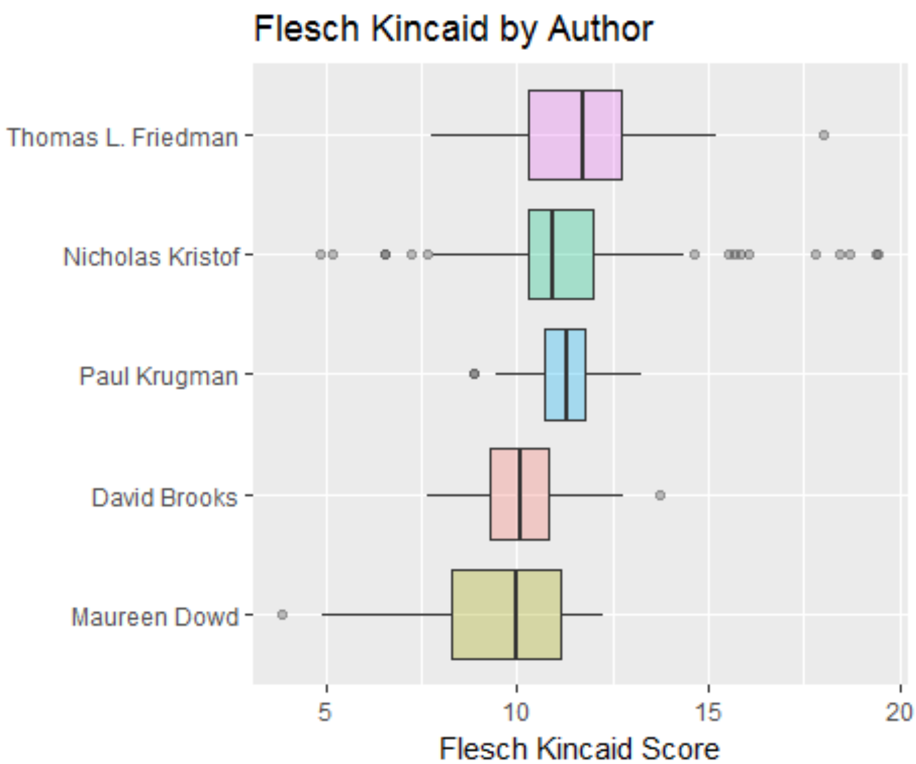
```
meta(df_corpus, type="local", tag="author") <- data_nyt$author
meta(df_corpus, type="local", tag="subject") <- data_nyt$subject
meta(df_corpus, type="local", tag="person") <- data_nyt$person
meta(df_corpus, type="local", tag="date") <- data_nyt$date

nyt_corpus <- corpus(df_corpus)  # convert to quanteda corpus
FRE_nyt <- textstat_readability(nyt_corpus,
              measure=c('Flesch.Kincaid'))
```

Next I tidy the corpus and tidy the Flesch-Kincaid grade level scores and combine hem as one dataframe for analysis.

```
fre1_nyt <- tidy(FRE_nyt)
raw_nyt <- tidy(df_corpus)

id <- rownames(fre1_nyt)
fre1_nyt <- cbind(id=id, fre1_nyt)
fre_nyt <- right_join(raw_nyt, fre1_nyt, by = "id")

## Warning in right_join_impl(x, y, by$x, by$y, suffix$x, suffix$y):
joining
## character vector and factor, coercing into character vector
```

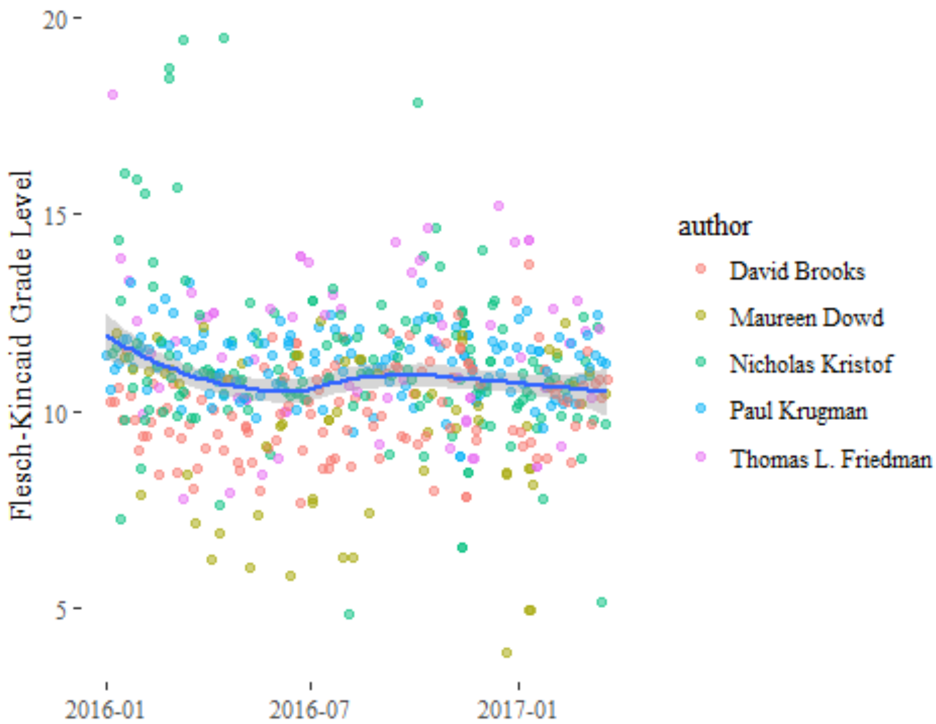
I decided to first plot the authors Flesch-Kincaid grade level as a boxplot because I feel box plots do a great job of clearly showing the reader the mean and distribution of grade levels for the articles written by each author. For this reason I did include this plot.

```
j <- ggplot(data=fre_nyt,aes(x=reorder(author, x, na.rm=TRUE), y=x))
j + geom_boxplot(aes(fill=author), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Flesch Kincaid by Author")
```



Next I plotted the Grade level by date in order to see how the grade levels might change over time. I also colored the dots by author to see how the authors might differ or change over time. I included this plot because it shows how similar all authors are in there grade levels but also shows that some authors are far from the average with some select authors in both directions.

```
fre_nyt$date <- as.Date(fre_nyt$date)

ggplot(data=fre_nyt, aes(x=date,y=x)) +
  geom_point(alpha=0.5,aes(col=author)) +
  geom_smooth() +
  guides(size=FALSE) + theme_tufte() +
  xlab("") + ylab("Flesch-Kincaid Grade Level")

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



I also created a table of the mean Flesch-Kincaid for each other in order to see each others mean.

```
frebyauthor <- aggregate(x~author, data=fre_nyt, FUN=function(x)
c(mean=mean(x), count=length(x)))

frebyauthor

##                 author     x.mean    x.count
## 1       David Brooks  10.085517 126.000000
## 2       Maureen Dowd   9.494868  63.000000
## 3   Nicholas Kristof  11.232045 157.000000
## 4       Paul Krugman  11.226747 134.000000
## 5 Thomas L. Friedman  11.560675  67.000000
```

Next I find the most popular subjects and picked out these four because they are somewhat different from one another and may show differences in the grade level by subject.

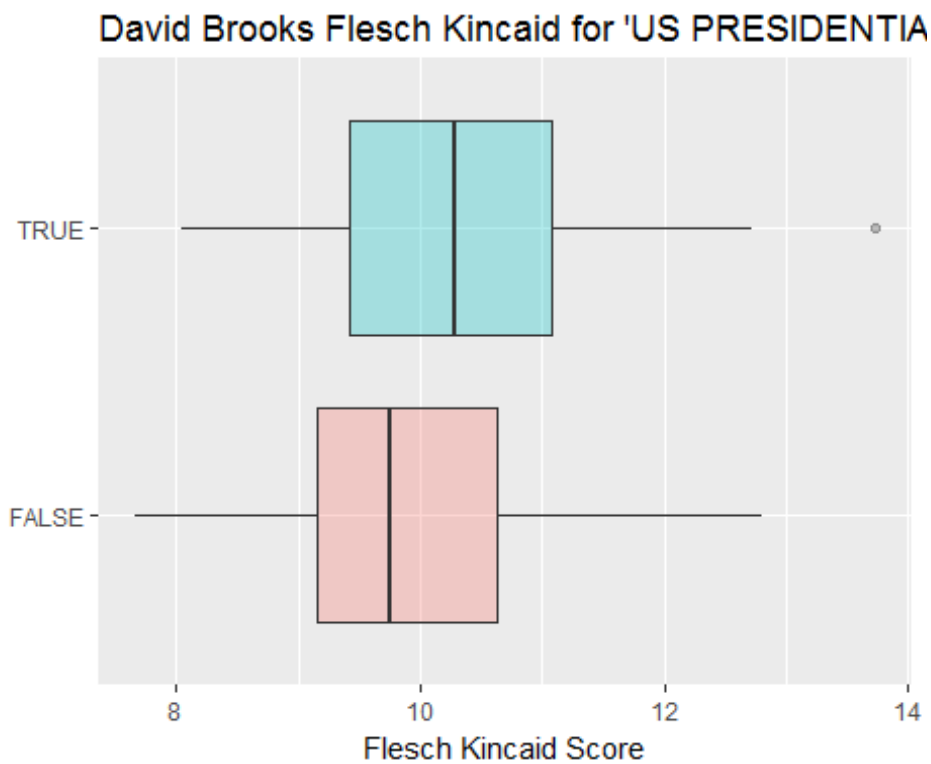
```
#seperate out subjects and find major topics
subjects <- gsub( " *\\(.*?\\) *", "", nyt_corpus$documents$subject) #
Remove parentheses
subjects <- strsplit(subjects, ";")  # Split by ';' into a list
subjects <- lapply(subjects, FUN=trimws) # Remove whitespace
subjectlist <- unique(unlist(subjects)) # Make into a list, remove
whitespace
top10subjects <- rownames(sort(table(unlist(subjects)),
```

```
decreasing=TRUE)[2:21])
subjects <- lapply(subjects, FUN=
function(A,B){top10subjects[match(A,top10subjects)]})
subjects <- lapply(subjects, function(x) x[!is.na(x)])
subjects1 <- tidy(top10subjects)
library(reshape)
subjects1 <- rename(subjects1, c(x="subject"))

#create variables for top/diferent subjects
nyt_corpus$documents$pres16_article <- grepl("US PRESIDENTIAL
CANDIDATES 2016", nyt_corpus$documents$subject, fixed=TRUE)
nyt_corpus$documents$religion_article <- grepl("RELIGION",
nyt_corpus$documents$subject, fixed=TRUE)
nyt_corpus$documents$writer_article <- grepl("WRITERS",
nyt_corpus$documents$subject, fixed=TRUE)
nyt_corpus$documents$immigration_article <- grepl("IMMIGRATION",
nyt_corpus$documents$subject, fixed=TRUE)
sub_nyt <- tidy(nyt_corpus)
sub_nyt <- right_join(fre1_nyt, sub_nyt, by = "id")

## Warning in right_join_impl(x, y, by$x, by$y, suffix$x, suffix$y):
joining
## factor and character vector, coercing into character vector

# subject variables by author

dbs <- filter(sub_nyt, author == "David Brooks")
nks <- filter(sub_nyt, author == "Nicholas Kristof")
pks <- filter(sub_nyt, author == "Paul Krugman")
mds <- filter(sub_nyt, author == "Maureen Dowd")
tfs <- filter(sub_nyt, author == "Thomas L. Friedman")
```

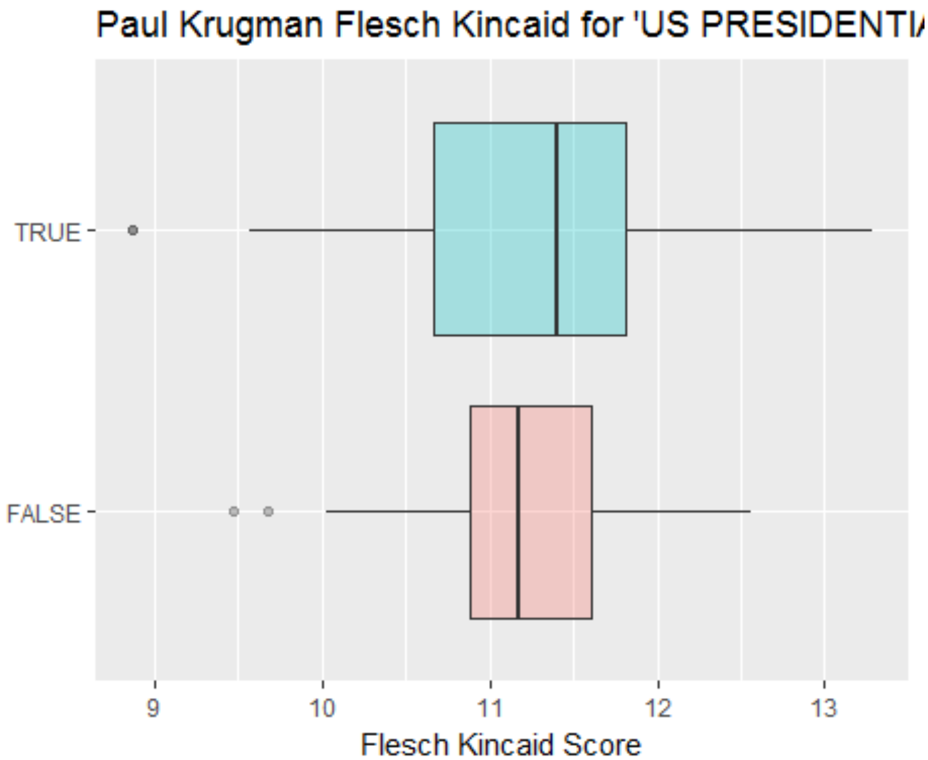
Next I create a boxplot for each author of each of the four subject vs. not that said subject to see how the authors write about these topics compared to the rest and compared to one another. Again I decided to use boxplots because they do a great job of showing the read the distribution/outlires for each author and also the mean. I feel boxplots are very easy to interpret and visually pleasing. In the final plots I only used the subjects "US PRESIDENTIAL CANDIDATES 2016" (because it is the most popular topic for all authors) and "Immigration" because it was the subject I felt showed the most differentiation between authors.

```
# fre score author by pres election 16

dbpr <- ggplot(data=dbs,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
dbpr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
```

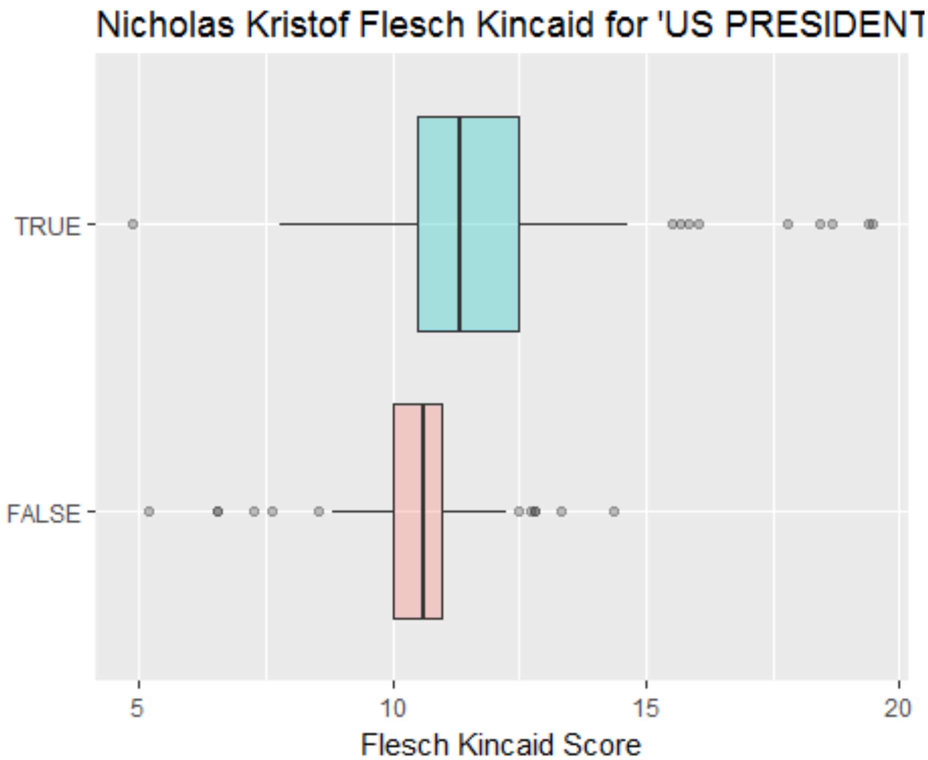
```
    ggtitle("David Brooks Flesch Kincaid for 'US PRESIDENTIAL
CANDIDATES 2016'")
```



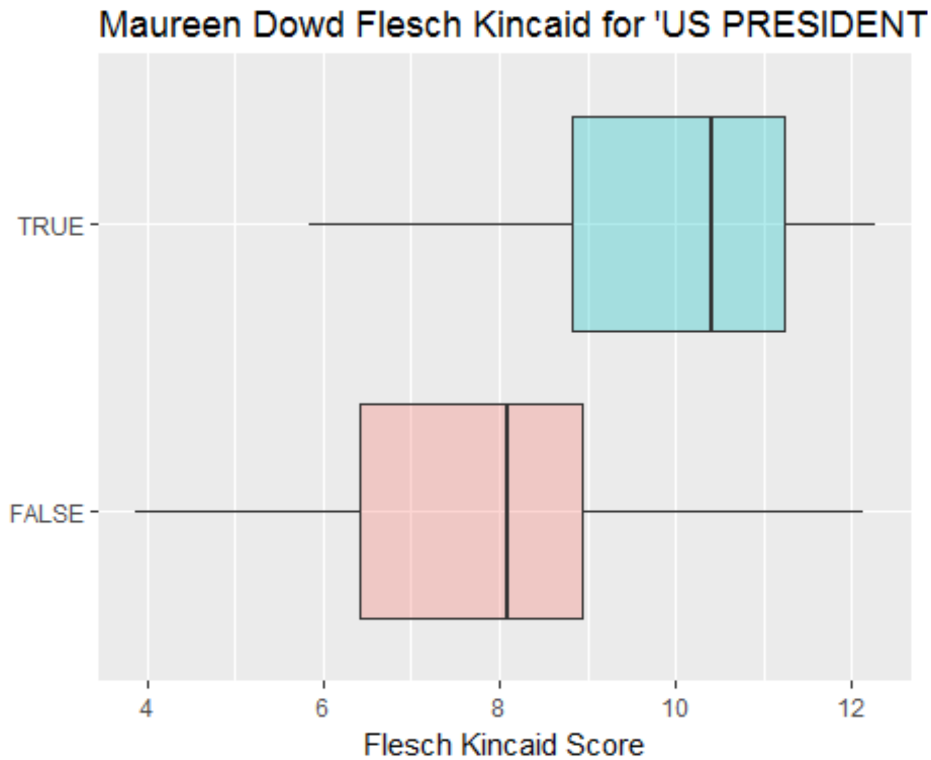

```
pkpr <- ggplot(data=pks,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
pkpr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Paul Krugman Flesch Kincaid for 'US PRESIDENTIAL
CANDIDATES 2016'")
```



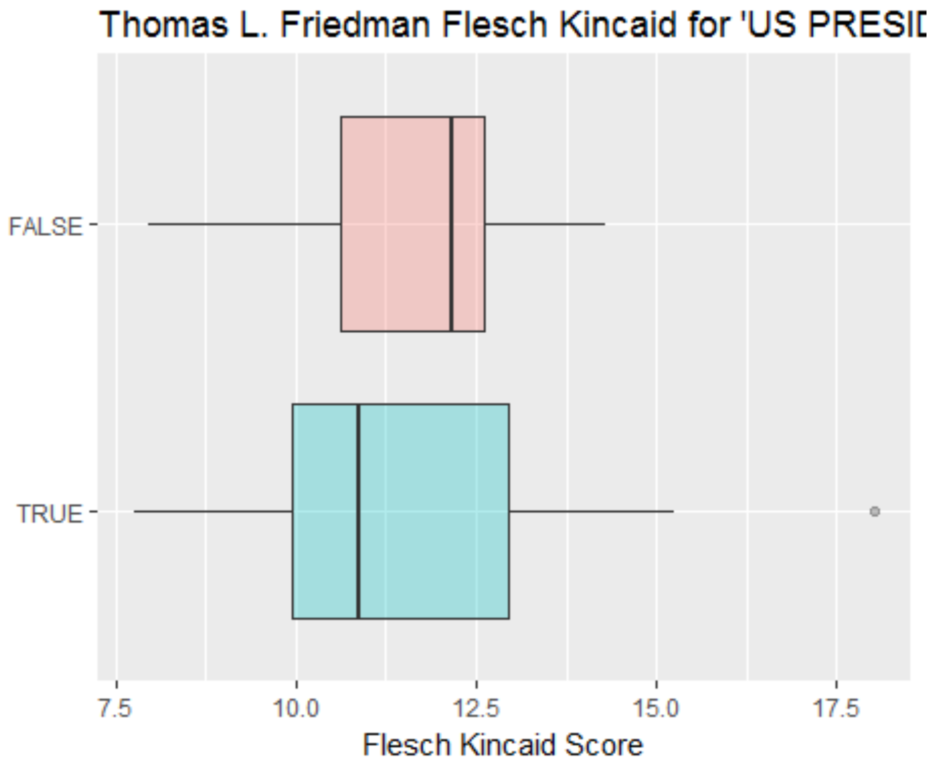

```
nkpr <- ggplot(data=nks,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
nkpr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Nicholas Kristof Flesch Kincaid for 'US PRESIDENTIAL
CANDIDATES 2016'")
```



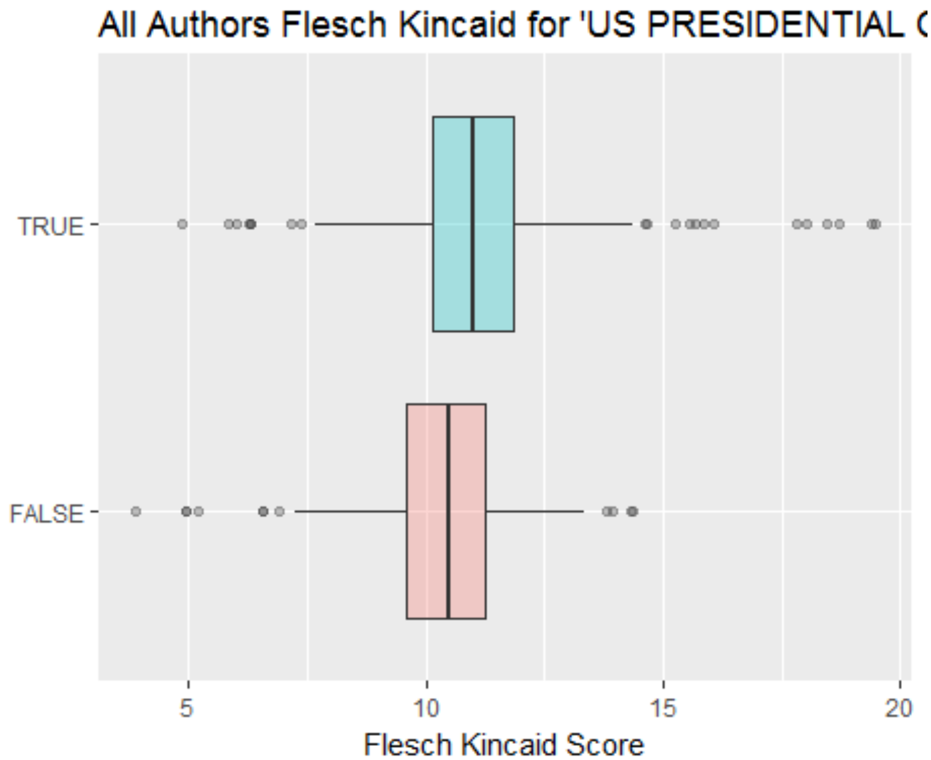

```
mdpr <- ggplot(data=mds,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
mdpr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Maureen Dowd Flesch Kincaid for 'US PRESIDENTIAL
CANDIDATES 2016'")
```



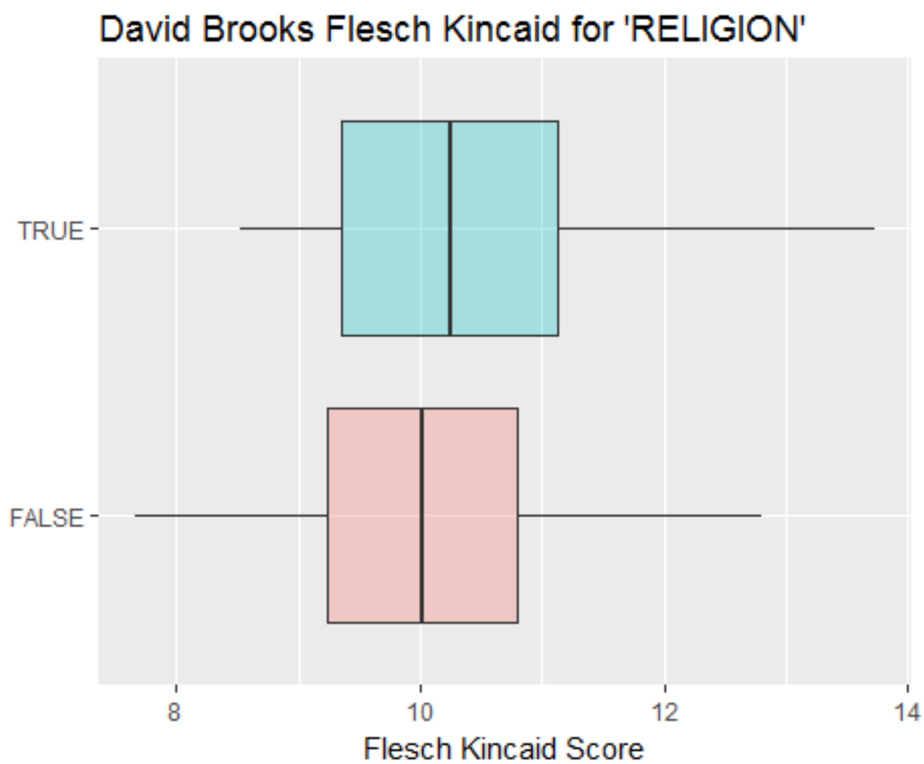

```
tfpr <- ggplot(data=tfs,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
tfpr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Thomas L. Friedman Flesch Kincaid for 'US PRESIDENTIAL
CANDIDATES 2016'")
```



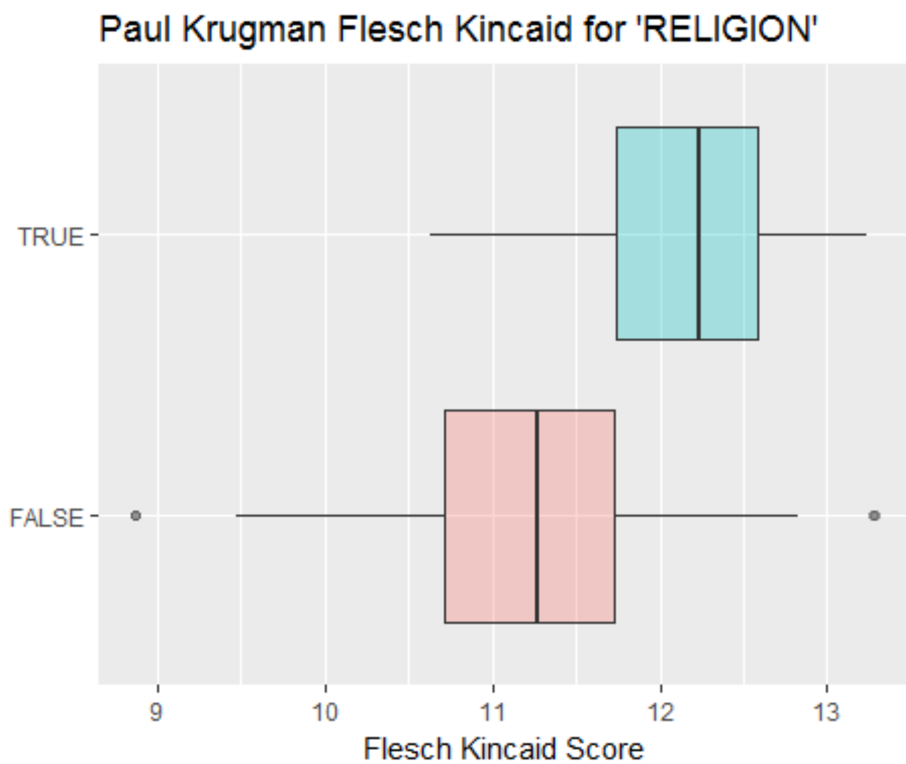

```
pr <- ggplot(data=sub_nyt,aes(x=reorder(pres16_article, x, na.rm=TRUE),
y=x))
pr + geom_boxplot(aes(fill=pres16_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("All Authors Flesch Kincaid for 'US PRESIDENTIAL CANDIDATES
2016'")
```



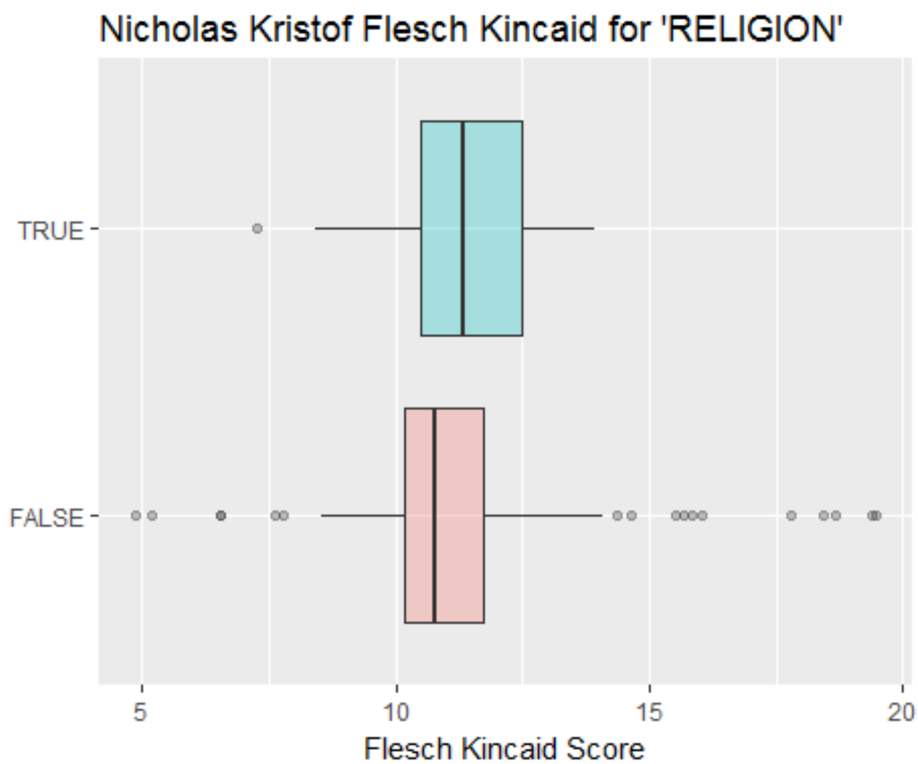

```
#by religion

dbr <- ggplot(data=dbs,aes(x=reorder(religion_article, x, na.rm=TRUE),
y=x))
dbr + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("David Brooks Flesch Kincaid for 'RELIGION'")
```



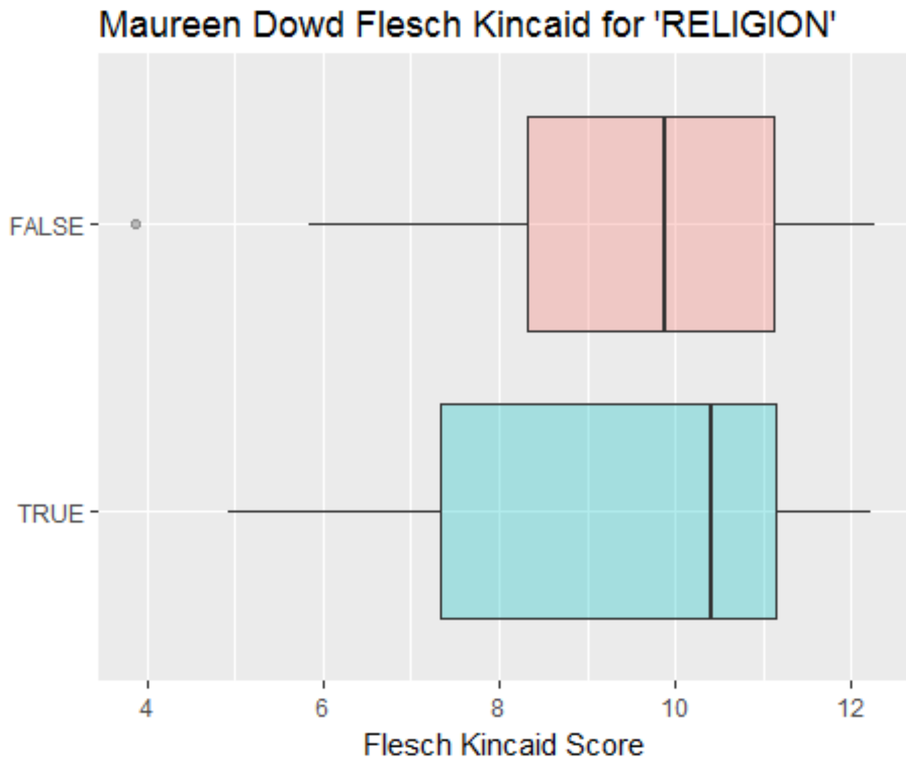

```
pkr <- ggplot(data=pks,aes(x=reorder(religion_article, x, na.rm=TRUE),
y=x))
pkr + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Paul Krugman Flesch Kincaid for 'RELIGION'")
```



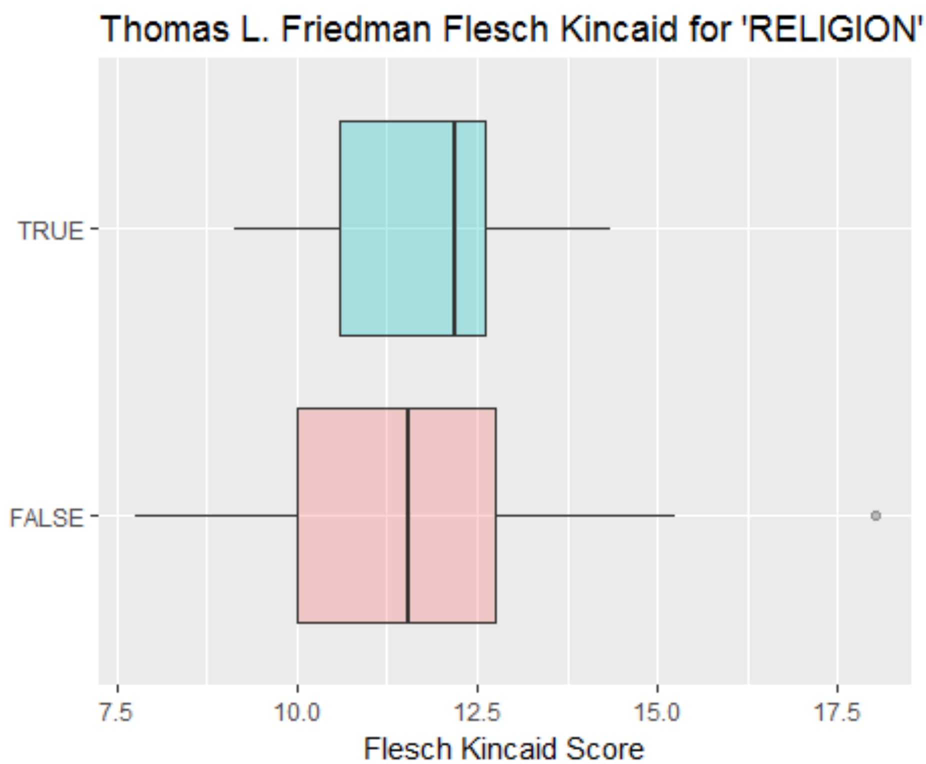

```r
nkr <- ggplot(data=nks,aes(x=reorder(religion_article, x, na.rm=TRUE),
y=x))
nkr + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Nicholas Kristof Flesch Kincaid for 'RELIGION'")
```



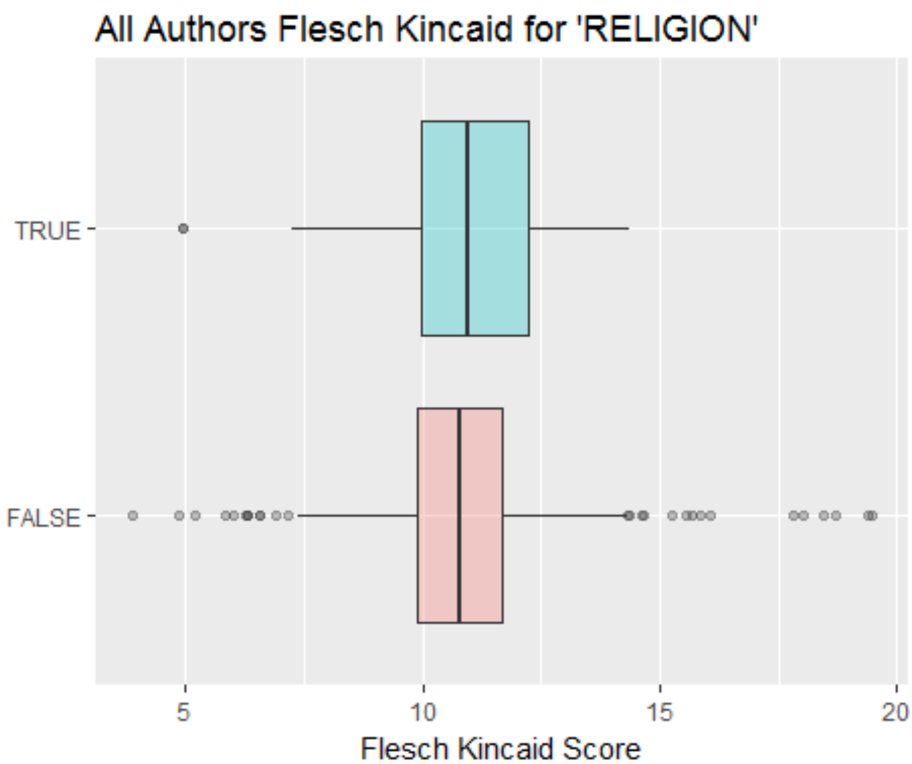

```
mdr <- ggplot(data=mds,aes(x=reorder(religion_article, x, na.rm=TRUE),
y=x))
mdr + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Maureen Dowd Flesch Kincaid for 'RELIGION'")
```



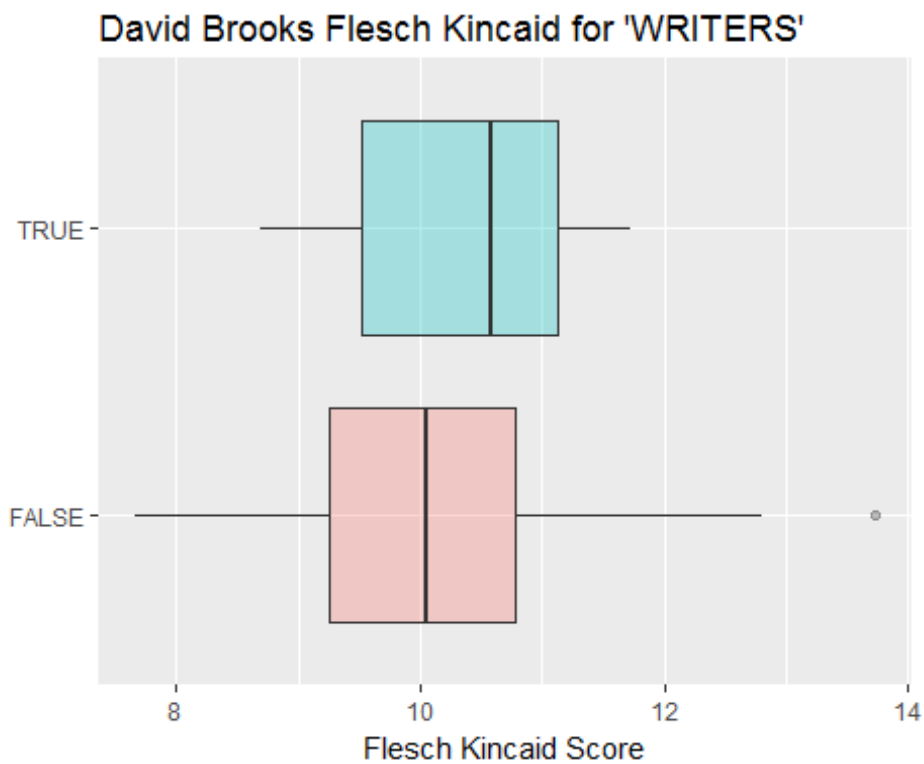

```
tfr <- ggplot(data=tfs,aes(x=reorder(religion_article, x, na.rm=TRUE),
y=x))
tfr + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Thomas L. Friedman Flesch Kincaid for 'RELIGION'")
```



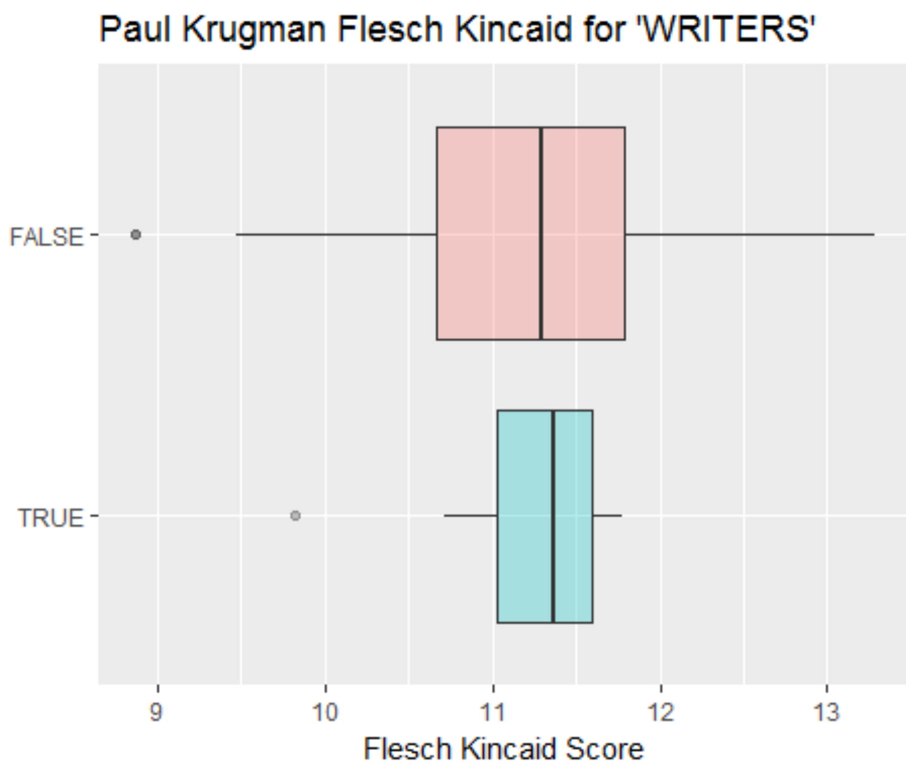

```
r <- ggplot(data=sub_nyt,aes(x=reorder(religion_article, x,
na.rm=TRUE), y=x))
r + geom_boxplot(aes(fill=religion_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("All Authors Flesch Kincaid for 'RELIGION'")
```



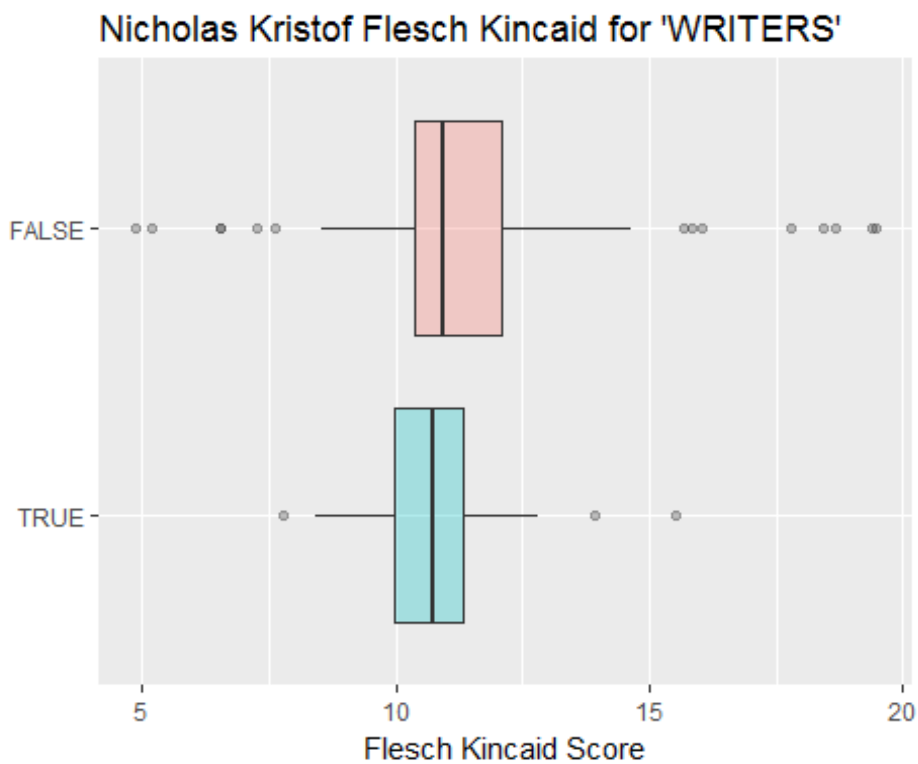

```
# by writers

dbw <- ggplot(data=dbs,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
dbw + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("David Brooks Flesch Kincaid for 'WRITERS'")
```



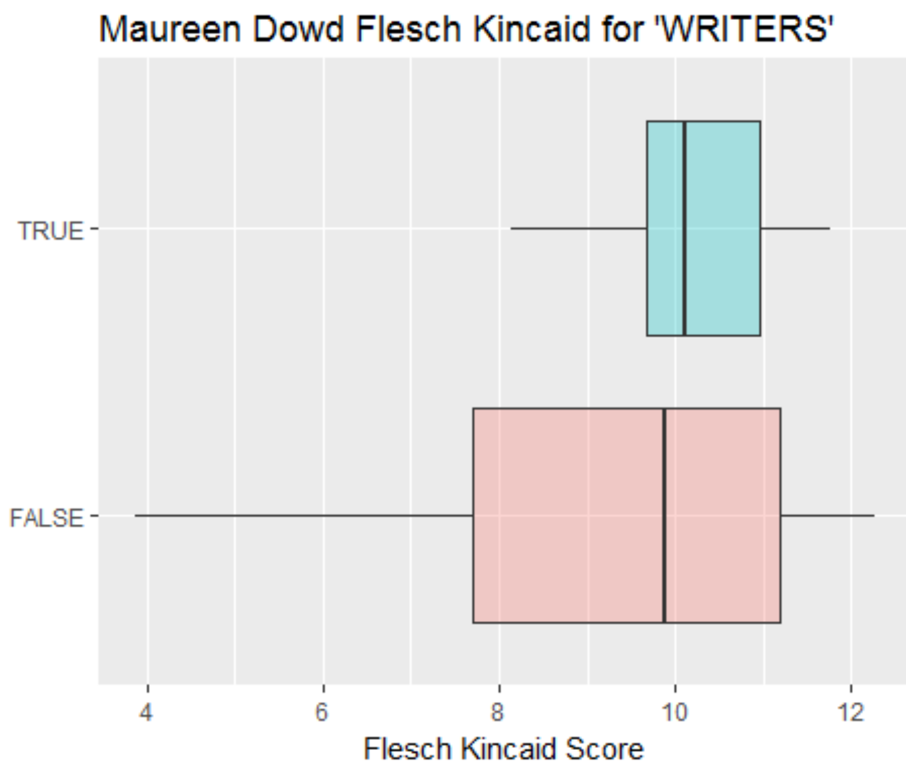

```
pkw <- ggplot(data=pks,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
pkw + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Paul Krugman Flesch Kincaid for 'WRITERS'")
```



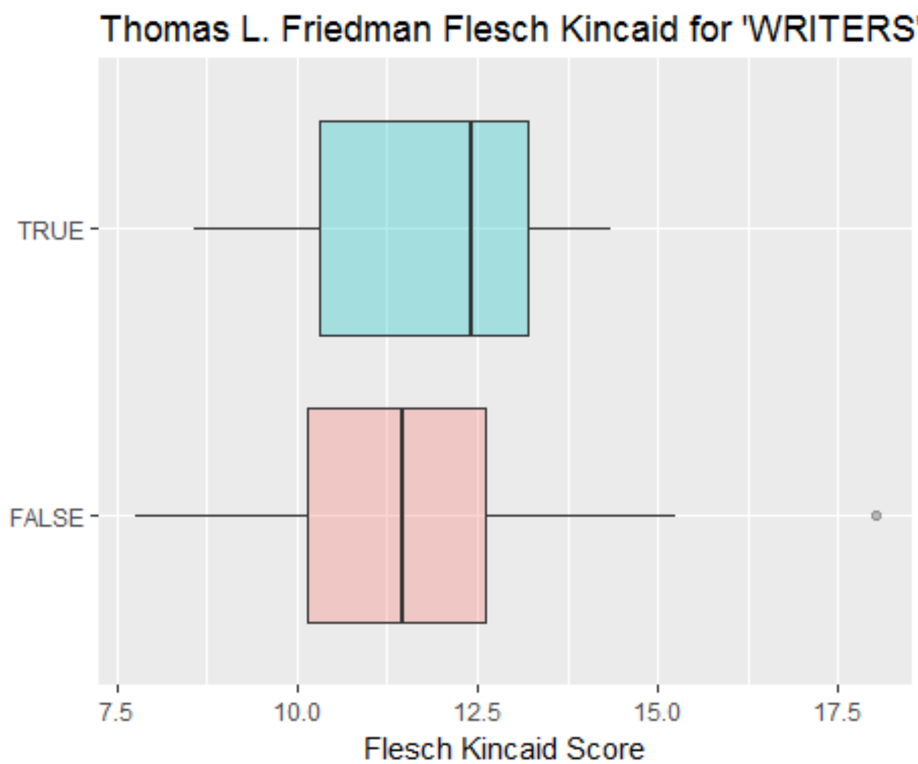

```
nkw <- ggplot(data=nks,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
nkw + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Nicholas Kristof Flesch Kincaid for 'WRITERS'")
```



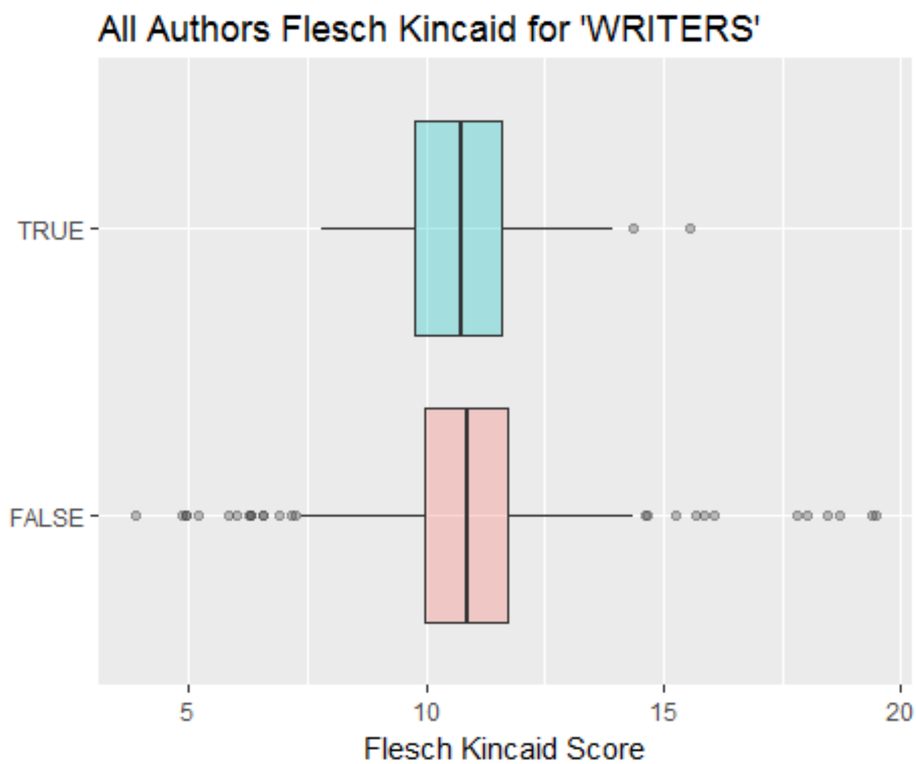

```
mdw <- ggplot(data=mds,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
mdw + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Maureen Dowd Flesch Kincaid for 'WRITERS'")
```



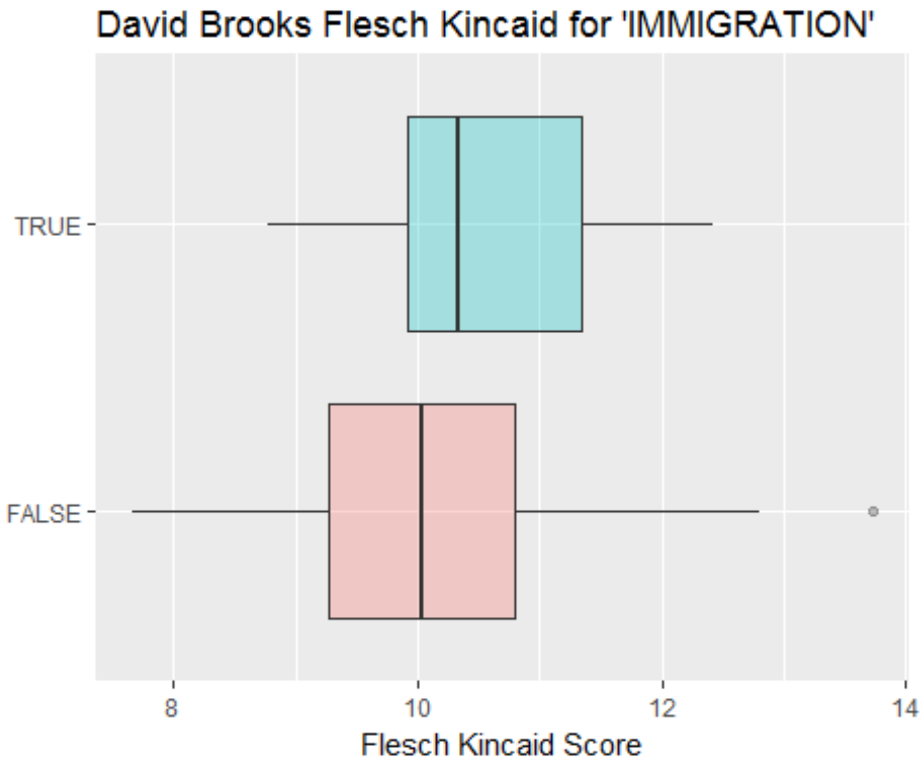

```
tfw <- ggplot(data=tfs,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
tfw + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Thomas L. Friedman Flesch Kincaid for 'WRITERS'")
```



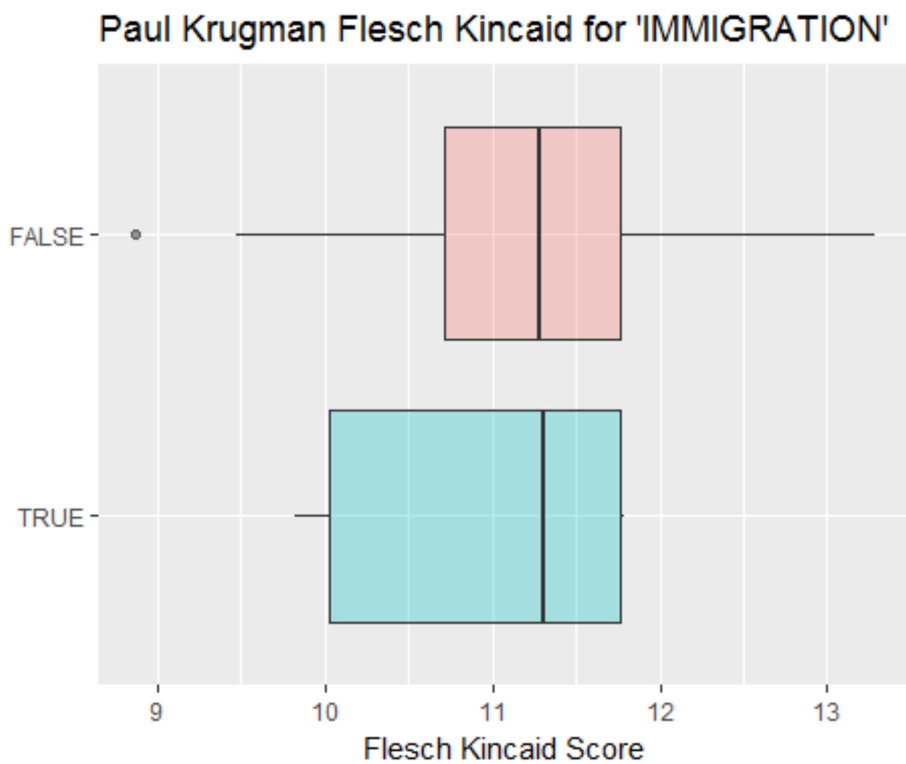

```
w <- ggplot(data=sub_nyt,aes(x=reorder(writer_article, x, na.rm=TRUE),
y=x))
w + geom_boxplot(aes(fill=writer_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("All Authors Flesch Kincaid for 'WRITERS'")
```



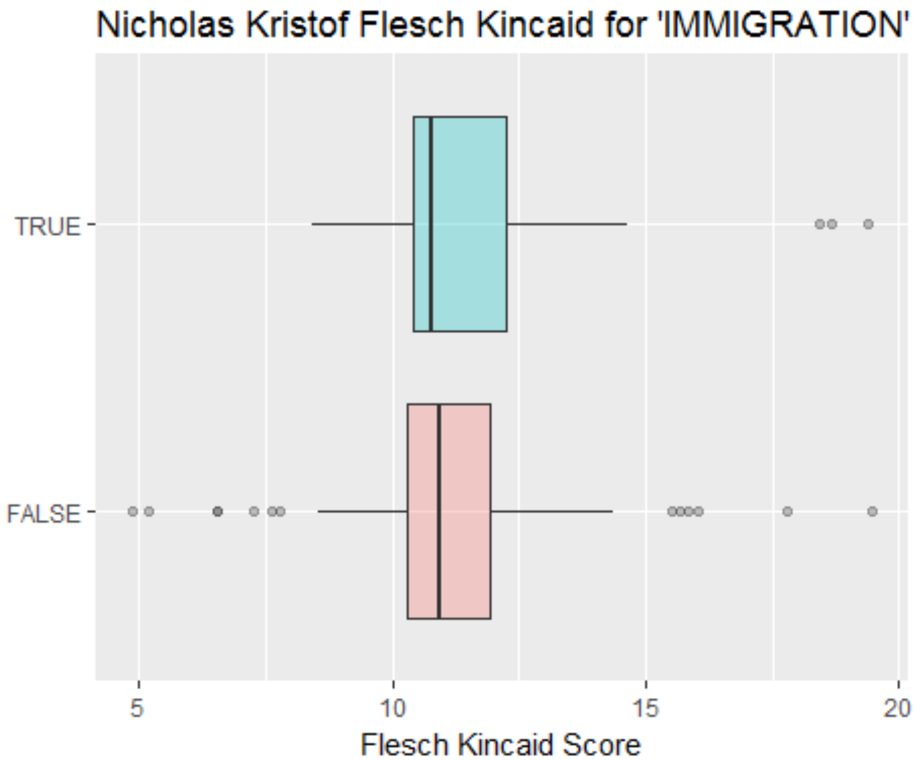

```
# by IMMIGRATION

dbi <- ggplot(data=dbs,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
dbi + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("David Brooks Flesch Kincaid for 'IMMIGRATION'")
```



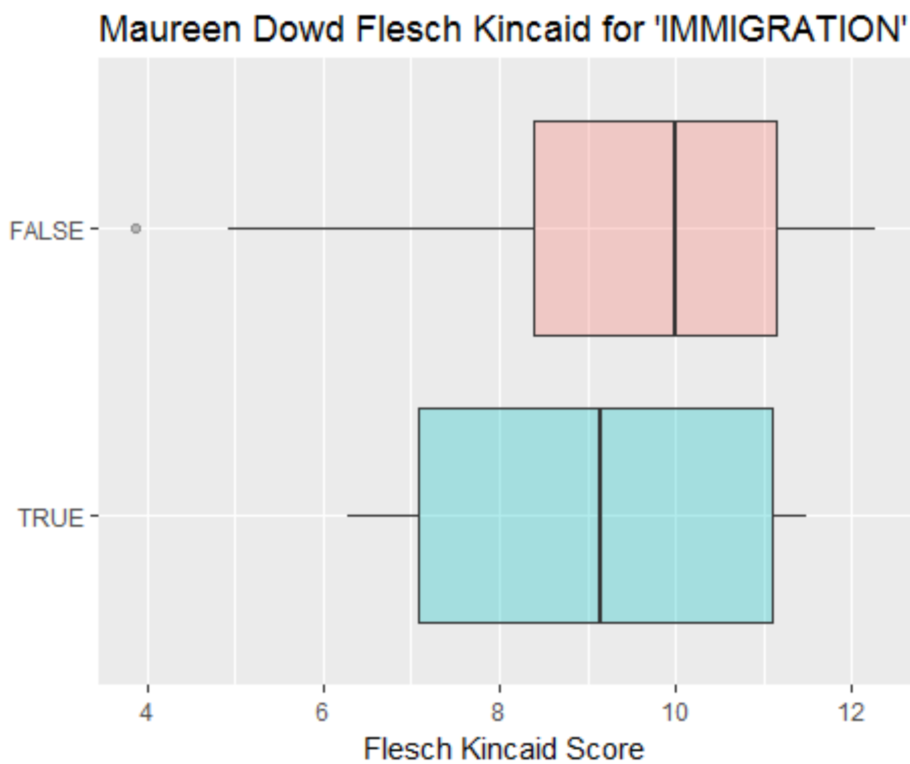

```
pki <- ggplot(data=pks,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
pki + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Paul Krugman Flesch Kincaid for 'IMMIGRATION'")
```



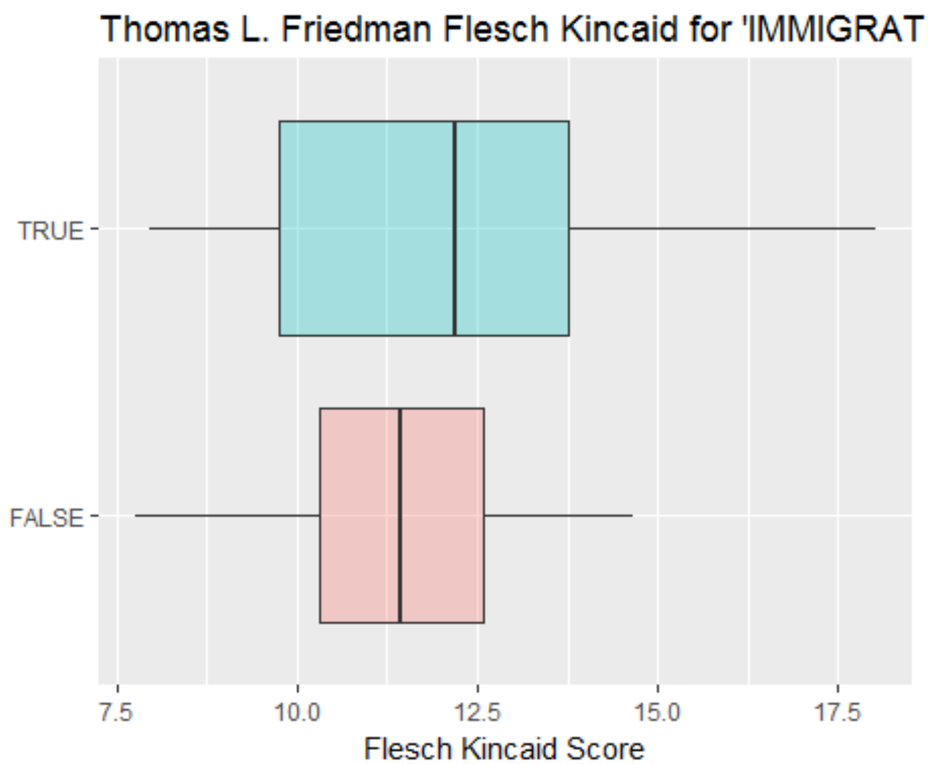

```
nki <- ggplot(data=nks,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
nki + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Nicholas Kristof Flesch Kincaid for 'IMMIGRATION'")
```



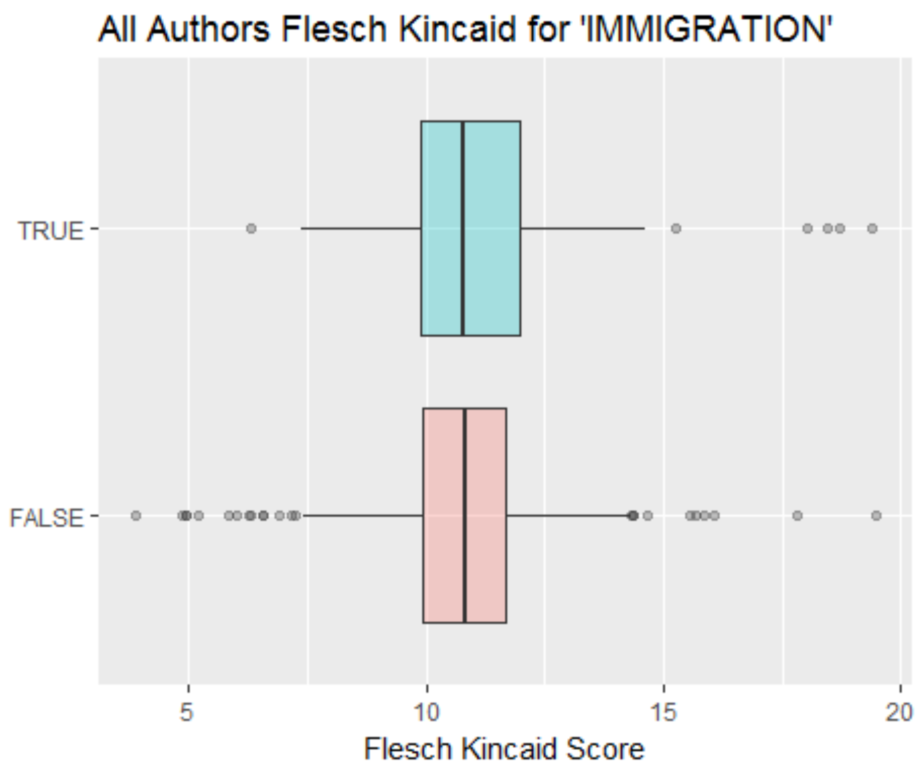

```
mdi <- ggplot(data=mds,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
mdi + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Maureen Dowd Flesch Kincaid for 'IMMIGRATION'")
```




```
tfi <- ggplot(data=tfs,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
tfi + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("Thomas L. Friedman Flesch Kincaid for 'IMMIGRATION'")
```




```
i <- ggplot(data=sub_nyt,aes(x=reorder(immigration_article, x,
na.rm=TRUE), y=x))
i + geom_boxplot(aes(fill=immigration_article), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Flesch Kincaid Score") +
    guides(fill=FALSE) +
    ggtitle("All Authors Flesch Kincaid for 'IMMIGRATION'")
```



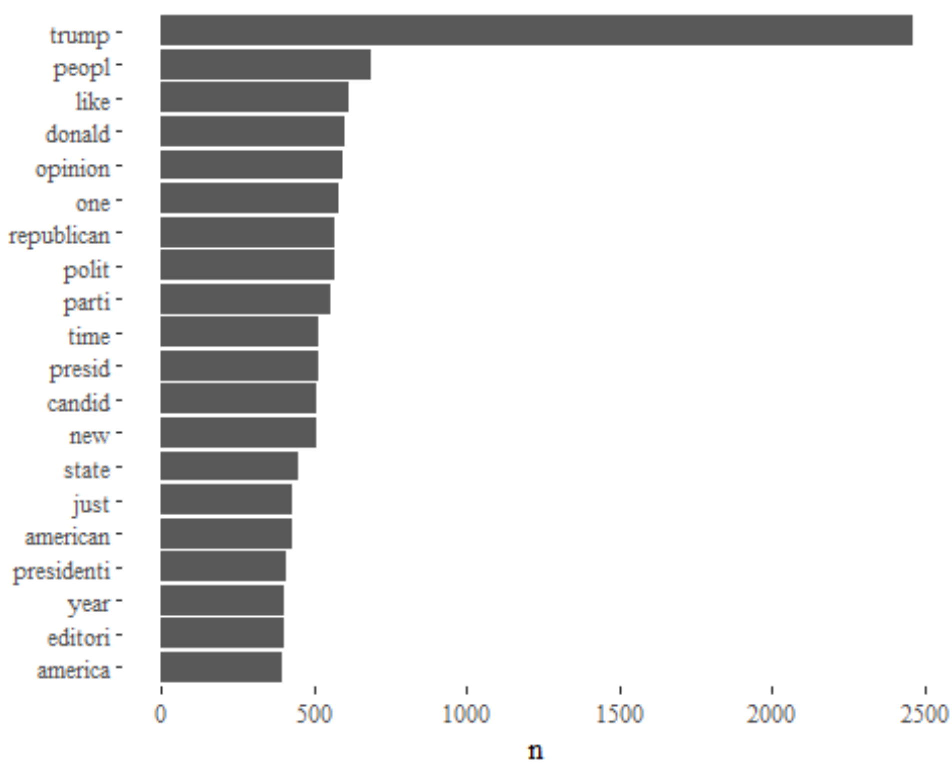
3

For number 3 I first had to seperate and create dataframes for Trump and Clinton seperatly. In order to do this I considered a Trump article to speak of Trump but not included Clinton and there are 222 articles in total. For Clinton I did the same and there were only 39 articles which Is a limitation but I felt it was ok overall.

```
corpus$documents$trump_article <- grepl("TRUMP",
corpus$documents$person, fixed=TRUE)
corpus$documents$clinton_article <- grepl("CLINTON",
corpus$documents$person, fixed=TRUE)
hvt_nty <- tidy(corpus)

trump <- filter(hvt_nty, trump_article == TRUE)
trump <- filter(trump, clinton_article == FALSE)

clinton <- filter(hvt_nty, clinton_article == TRUE)
clinton <- filter(clinton, trump_article == FALSE)

df_sourcetrump <- DataframeSource(trump)
df_corpustrump <- VCorpus(df_sourcetrump)

df_source_clin <- DataframeSource(clinton)
df_corpus_clin <- VCorpus(df_source_clin)
```

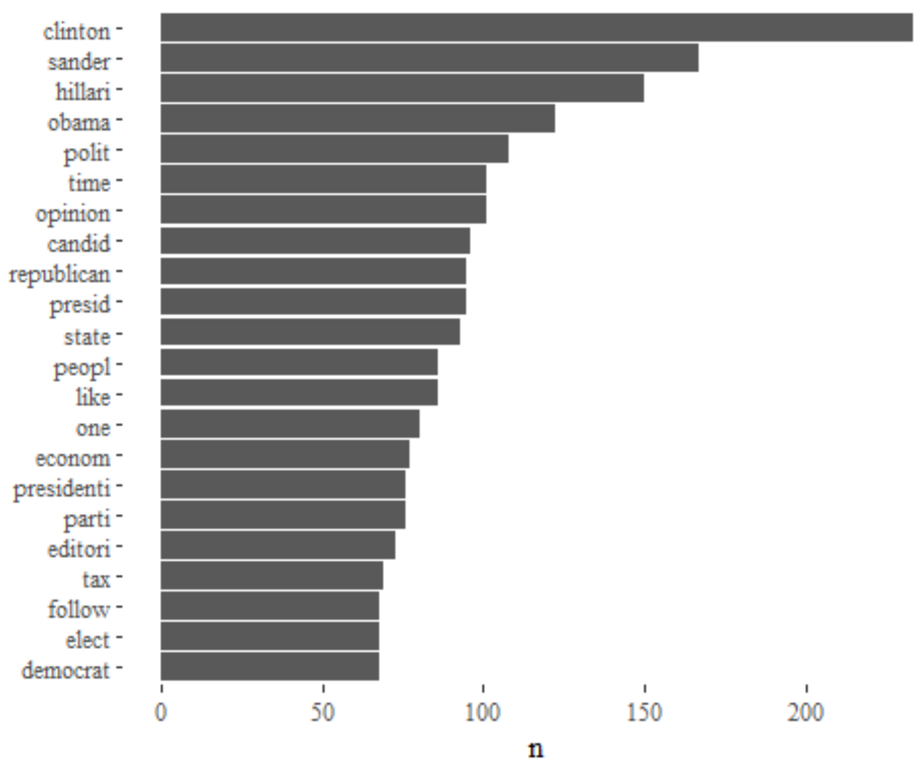
In order to see how the two differ I looked at the top 20 words used in Trump and Clinton articles. After doing this I decided to remove the first and last names of each candidate. I used these two plots in the final polished plots but I made Trump red for republican and Clinton blue for Democrat.

```
df_corpustrum <- clean_corpus(df_corpustrump)

trump_stemmed <- tm_map(df_corpustrum, stemDocument)

trump_tdm <- TermDocumentMatrix(trump_stemmed)

trump_td <- tidy(trump_tdm)

trump_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```



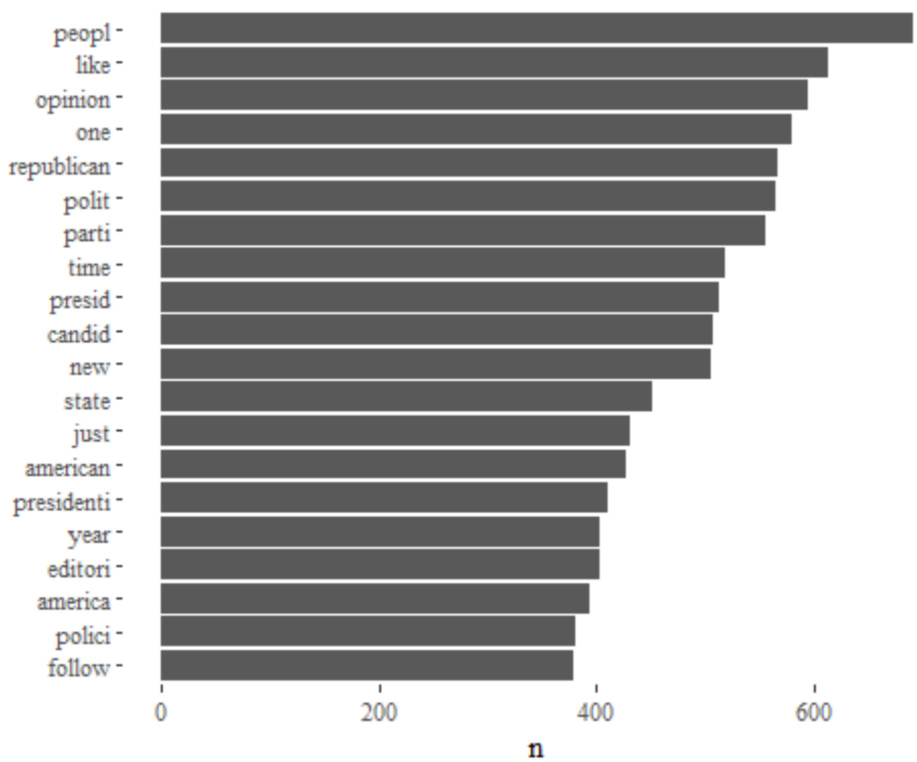

```
df_corpus_clin <- clean_corpus(df_corpus_clin)

clin_stemmed <- tm_map(df_corpus_clin, stemDocument)

clin_tdm <- TermDocumentMatrix(clin_stemmed)
```

```
clin_td <- tidy(clin_tdm)

clin_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```



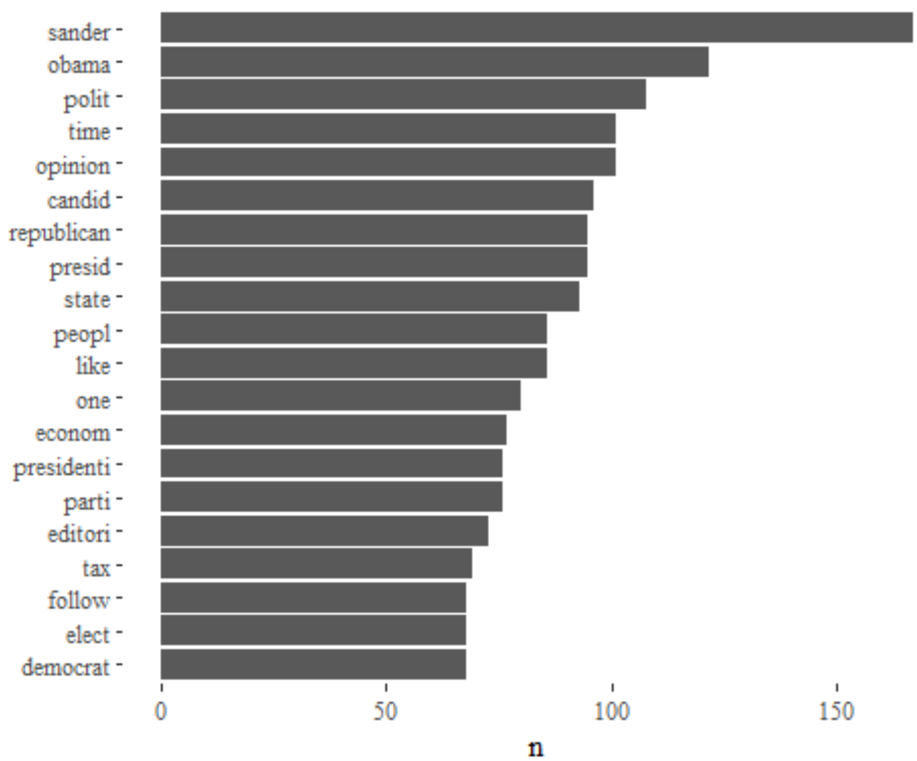

```
new_stops1 <- c("clinton", "trump", "hillari", "donald")

#clean text
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("en")))
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, removeWords, c("nicholas", "nickkristof",
"kristof", "say", "can", "will", "clinton", "trump", "hillary",
"donald"))
  return(corpus)
}
```

```
df_corpustrump <- clean_corpus(df_corpustrump)
df_corpus_clin <- clean_corpus(df_corpus_clin)

trump_stemmed <- tm_map(df_corpustrump, stemDocument)

trump_tdm <- TermDocumentMatrix(trump_stemmed)

trump_td <- tidy(trump_tdm)

trump_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```



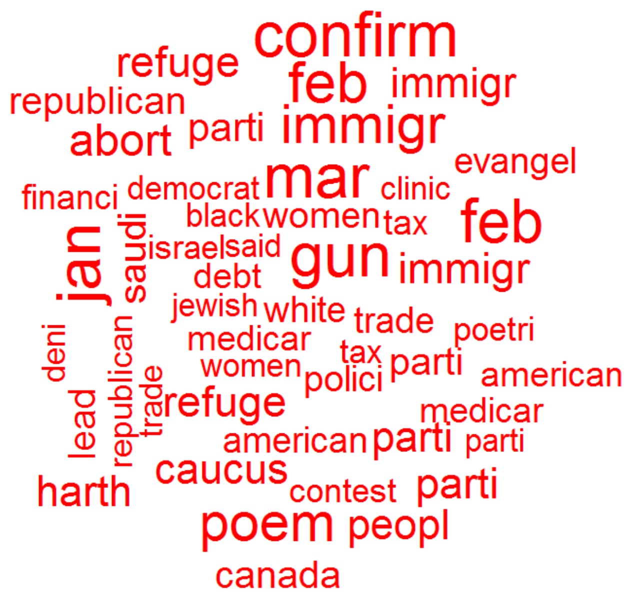

```
clin_stemmed <- tm_map(df_corpus_clin, stemDocument)

clin_tdm <- TermDocumentMatrix(clin_stemmed)

clin_td <- tidy(clin_tdm)

clin_td %>%     group_by(term) %>%
                summarise(n = sum(count)) %>%
```

```
                top_n(n = 20, wt = n)  %>%
                mutate(term = reorder(term, n)) %>%
ggplot(aes(term, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```

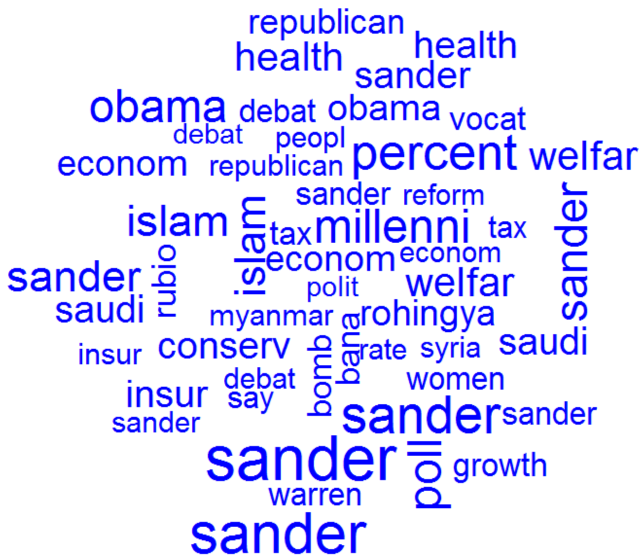


Next I again attempted word clouds but ran into the same issue as before. When attempting to solve the issue I was unsuccessful but since I had the graphs above I decided to not include the word cloud because they are basically accomplishing the same thing which is the top words used.

```
trump_tf_idf <-  trump_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))



# Set seed - to make your word cloud reproducible
set.seed(1234)

# Create a wordcloud for the values in word_freqs
wordcloud(trump_tf_idf$term, trump_tf_idf$tf,
          max.words = 50, colors = "red")
```

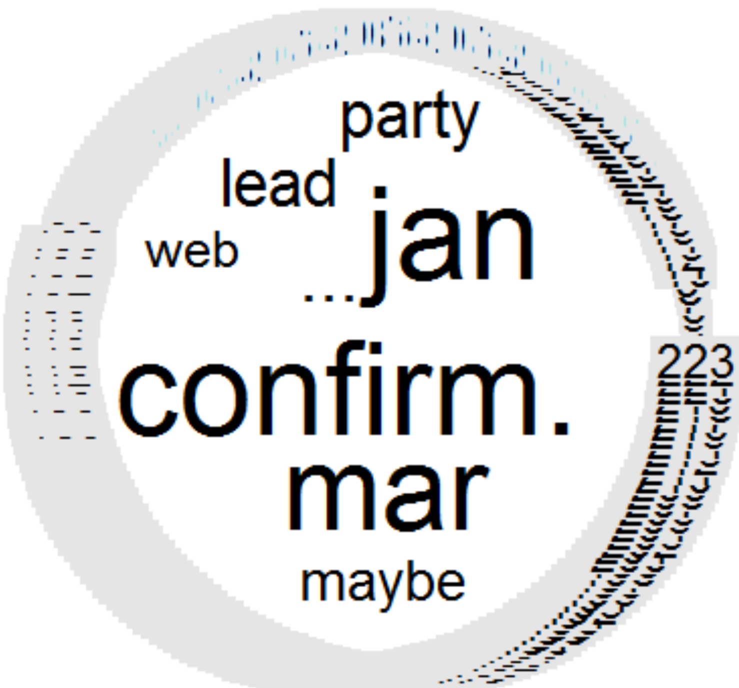



```r
clin_tf_idf <-  clin_td %>%
                bind_tf_idf(term, document, count) %>%
                arrange(desc(tf_idf))


# Set seed - to make your word cloud reproducible
set.seed(1234)

# Create a wordcloud for the values in word_freqs
wordcloud(clin_tf_idf$term, clin_tf_idf$tf,
          max.words = 50, colors = "blue")
```




```
t <- trump$text
c <- clinton$text

t_clean <- clean_byauthor(t)
c_clean <- clean_byauthor(c)

c <- paste(t_clean, collapse=" ")
c <- paste(c_clean, collapse=" ")

tc_vector <- c(t, c)

tc_vector <- removeWords(tc_vector,c(stopwords("english"), "Paul
Krugman", "David Brooks", "Thomas L. Friedman", "Maureen Dowd",
"Nicholas Kristof", "nickkristof", "say", "can", "will", "jan", "feb",
"mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec",
"clinton", "trump", "hillary", "donald"))
```

```r
# create corpus
tc_corp <- Corpus(VectorSource(tc_vector))

# create term-document matrix
tc_tdm <- TermDocumentMatrix(tc_corp)

# convert as matrix
tc_tdm <- as.matrix(tc_tdm)

# add column names
#colnames(tc_tdm) <- c("TRUMP", "CLINTON")

comparison.cloud(tc_tdm, random.order=FALSE,
colors = brewer.pal(8, "Set1"),
title.size=1.5
```



Next I create the dictionaries (positive and negative) for sentiment analysis. I also use a function to compute the sentiment score for each article and then attach the scores to the Clinton and Trump dataframes for analysis.

```r
pos <- read.table("dictionaries/positive-words.txt", as.is=T)
neg <- read.table("dictionaries/negative-words.txt", as.is=T)
neg[1:15,]

##  [1] "2-faced"     "2-faces"     "abnormal"    "abolish"
"abominable"
```

```
## [6] "abominably"  "abominate"   "abomination" "abort"
"aborted"
## [11] "aborts"      "abrade"      "abrasive"    "abrupt"
"abruptly"
```

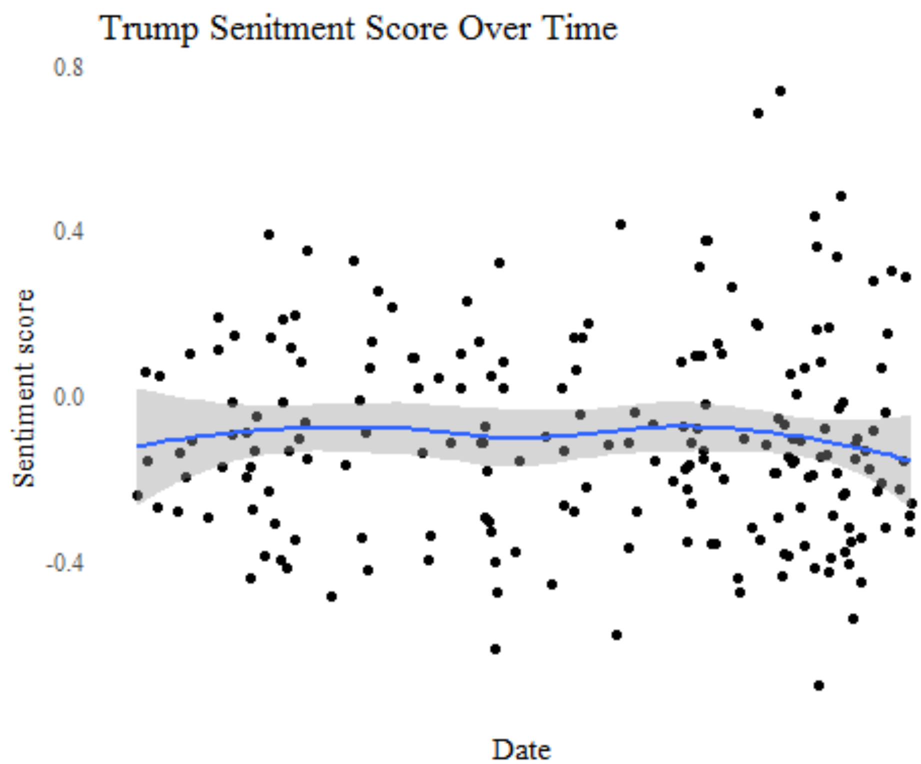
```
pos[1:15,]
```

```
##  [1] "a+"             "abound"        "abounds"       "abundance"
##  [5] "abundant"       "accessable"    "accessible"    "acclaim"
##  [9] "acclaimed"      "acclamation"   "accolade"      "accolades"
## [13] "accommodative" "accomodative"  "accomplish"
```

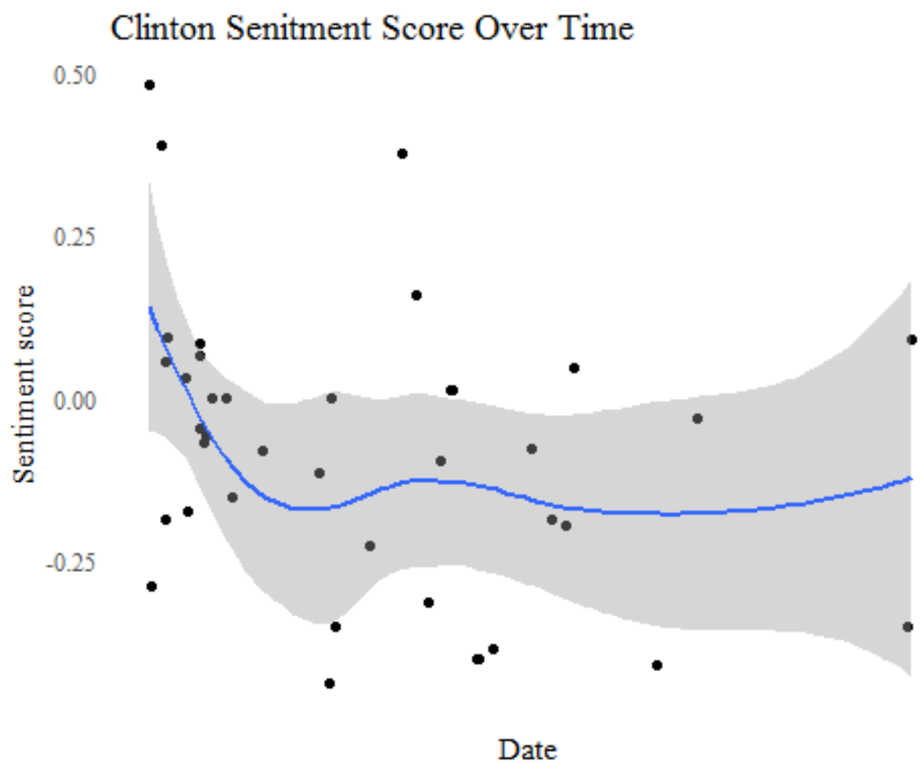
```
sentiment <- function(words){
  require(quanteda)
  tok <- quanteda::tokenize(words)
  pos.count <- sum(tok[[1]]%in%pos[,1])
  cat("\n positive words:",tok[[1]][which(tok[[1]]%in%pos[,1])],"\n")
  neg.count <- sum(tok[[1]]%in%neg[,1])
  cat("\n negative words:",tok[[1]][which(tok[[1]]%in%neg[,1])],"\n")
  out <- (pos.count - neg.count)/(pos.count+neg.count)
  return(out)
}

trump$text <- as.character(trump$text)
trump$sentiment <- NA
for (i in 1:nrow(trump)){
  trump[[i,16]] <- sentiment(trump[[i,1]])
}
```

After accomplishing that I was finally able to plot. I plotted for Trump and Clinton sentiment scores by date to see how the scores change over time. The scores do not show much change overtime except for a small hump in Clinton which I looked into and found out it was the time she won the Democratic Nomination. In the final plots I used color to show the authors in order to compare them and see how they differ. I also chose to include word count as the size of the dots because I feel it might be important/useful to know how long these specific articles are.

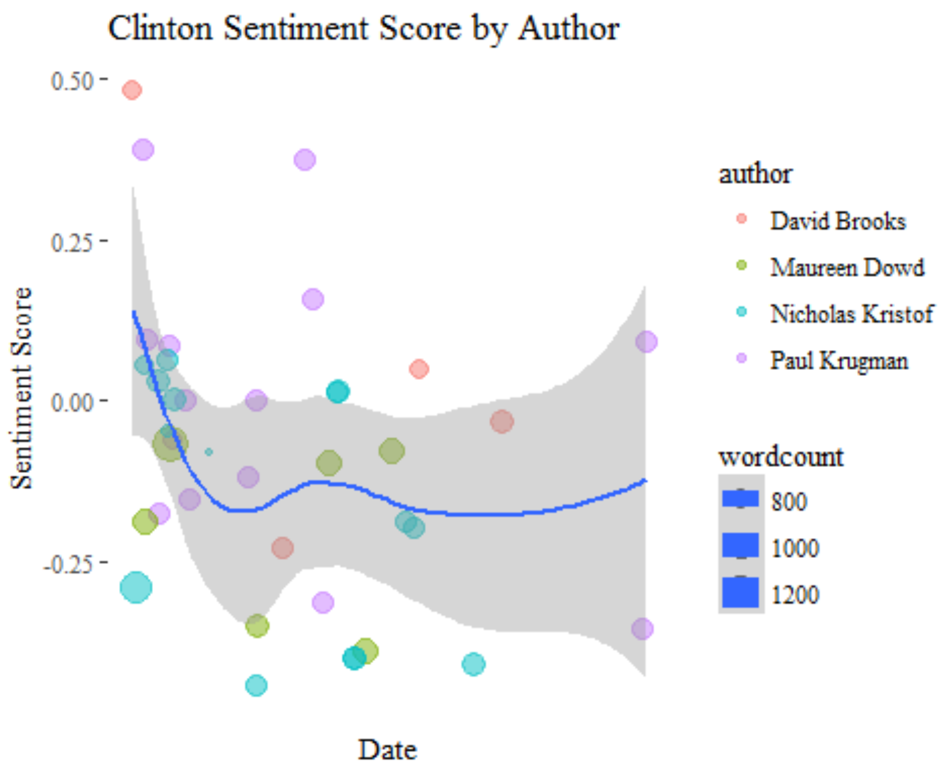
```
trump$date <- as.Date(trump$date)

ggplot(trump,
  aes(x=date, y=sentiment)) + geom_point() +
  ylab("Sentiment score") + xlab("Date") + theme_tufte() +
  geom_smooth() +
  ggtitle("Trump Senitment Score Over Time") +
  theme(axis.text.x=element_blank(),axis.ticks=element_blank())

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



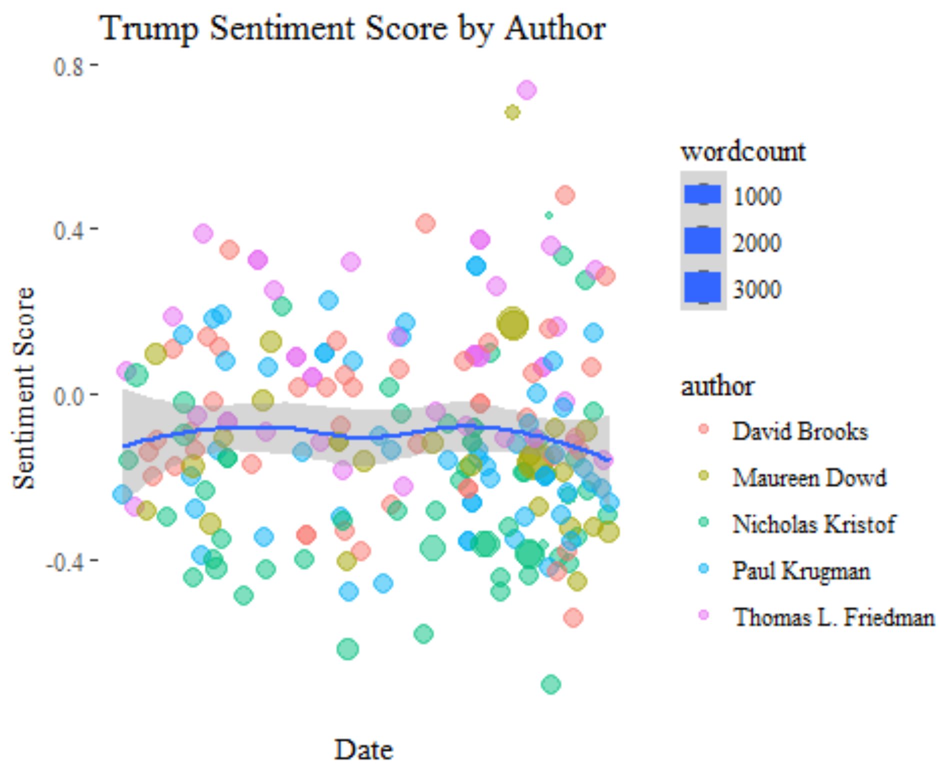

```
clinton$date <- as.Date(clinton$date)

ggplot(clinton,
  aes(x=date, y=sentiment)) + geom_point() +
  ylab("Sentiment score") + xlab("Date") + theme_tufte() +
  geom_smooth() +
  ggtitle("Clinton Senitment Score Over Time") +
  theme(axis.text.x=element_blank(),axis.ticks=element_blank())

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




```r
ggplot(data=clinton, aes(x=date,y=sentiment, size=wordcount)) +
  geom_point(alpha=0.5,aes(col=author)) +
  geom_smooth() +
  theme_tufte() +
  xlab("Date") + ylab("Sentiment Score") +
  ggtitle("Clinton Sentiment Score by Author") +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank())

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




```
ggplot(data=trump, aes(x=date,y=sentiment, size=wordcount)) +
  geom_point(alpha=0.5,aes(col=author)) +
  geom_smooth() +
  theme_tufte() +
  xlab("Date") + ylab("Sentiment Score") +
  ggtitle("Trump Sentiment Score by Author") +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank())

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



Last I calculate the mean for overall sentiment score of Trump and Clinton as well as the mean scores for Tump and Clinton by each individual author. I did this and included it in the final plots specifically because I feel the graphs are nice but it would be more easy and beneficial to also see the actual means for comparison.

```
overall_sent_clin <- mean(clinton$sentiment)
overall_sent_trump <- mean(trump$sentiment)

overall_sent_clin

## [1] -0.0809698

overall_sent_trump

## [1] -0.1003955

clinton_db <- filter(clinton, author == "David Brooks")
clinton_nk <- filter(clinton, author == "Nicholas Kristof")
clinton_pk <- filter(clinton, author == "Paul Krugman")
clinton_md <- filter(clinton, author == "Maureen Dowd")
clinton_tf <- filter(clinton, author == "Thomas L. Friedman")

trump_db <- filter(trump, author == "David Brooks")
trump_nk <- filter(trump, author == "Nicholas Kristof")
trump_pk <- filter(trump, author == "Paul Krugman")
trump_md <- filter(trump, author == "Maureen Dowd")
trump_tf <- filter(trump, author == "Thomas L. Friedman")
```

```r
db_sent_trump <- mean(trump_db$sentiment)
nk_sent_trump <- mean(trump_nk$sentiment)
pk_sent_trump <- mean(trump_pk$sentiment)
md_sent_trump <- mean(trump_md$sentiment)
tf_sent_trump <- mean(trump_tf$sentiment)

db_sent_clin <- mean(clinton_db$sentiment)
nk_sent_clin <- mean(clinton_nk$sentiment)
pk_sent_clin <- mean(clinton_pk$sentiment)
md_sent_clin <- mean(clinton_md$sentiment)
tf_sent_clin <- mean(clinton_tf$sentiment)
```

```r
db_sent_trump
```

```
## [1] -0.0598379
```

```r
nk_sent_trump
```

```
## [1] -0.2305594
```

```r
pk_sent_trump
```

```
## [1] -0.1252047
```

```r
md_sent_trump
```

```
## [1] -0.1227132
```

```r
tf_sent_trump
```

```
## [1] 0.08380603
```

```r
db_sent_clin
```

```
## [1] 0.06669739
```

```r
nk_sent_clin
```

```
## [1] -0.1517392
```

```r
pk_sent_clin
```

```
## [1] 0.001378833
```

```r
md_sent_clin
```

```
## [1] -0.1946379
```

```r
tf_sent_clin
```

```
## [1] NaN
```