Visualizations

Amanda McDermott

5/7/2019

Shiny

I was having issues uploading my Shiny to shinyapps.io for some reason but have worked out an alternative. The Shiny is on my GitHub so using the code below will compile the app and will automatically pop up once finished compiling. It takes a bit because there are a lot of parts and the tokenized dataset is huge, but it works.

```
runGitHub("Glacieus/DataSci", "Glacieus", subdir = "Shiny/")
```

Visualizations

```
# Positive and Negative Political Lexicon
# Source: https://rstudio-pubs-static.s3.amazonaws.com/338458_3478e1d95ccf49bf90b30abdb4e3bd40.html
url <- read_html("https://rstudio-pubs-static.s3.amazonaws.com/338458_3478e1d95ccf49bf90b30abdb4e3bd40.h

words <- url %>%
  html_nodes("#full-lexicon td") %>%
  html_text() %>%
  as_tibble() %>%
  .[seq(1, nrow(.), 2), ]

score <- url %>%
  html_nodes("#full-lexicon td") %>%
  html_text() %>%
  as_tibble() %>%
  .[seq(2, nrow(.), 2), ]

poli_lexicon <- cbind(words, score)
colnames(poli_lexicon) <- c("word", "sent_score")
poli_lexicon <- poli_lexicon %>%
  transform(sent_score = as.double(sent_score))
```

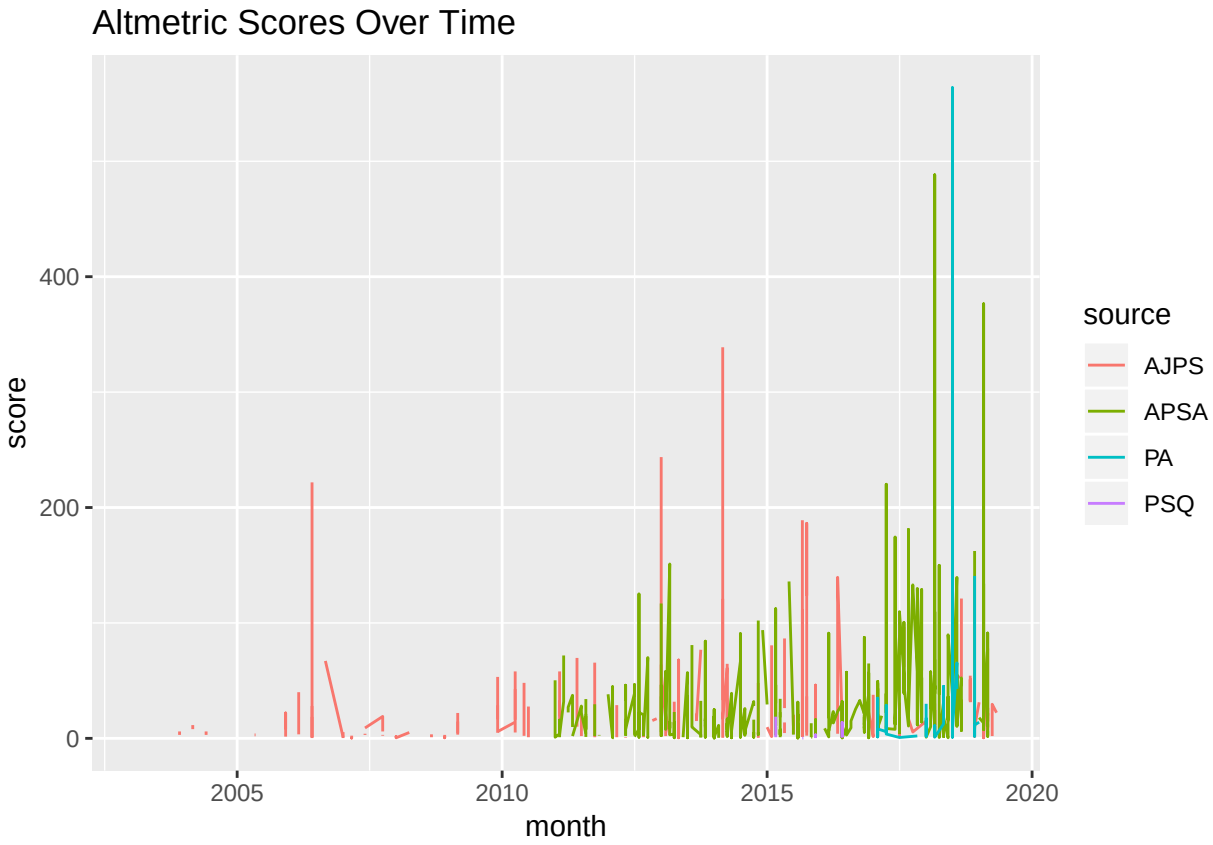
```
my_stopwords <- tibble(word = c("stix", "1", "2", "3", "4", "0", "5", "x1d6fc", "e.g", "al", "6", "x_",


tidytexts <- full_texts %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(my_stopwords) %>%
  left_join(arm_poli_lexicon, by = "word") %>%
  left_join(poli_lexicon, by = "word")
```

```
## Joining, by = "word"
## Joining, by = "word"
```

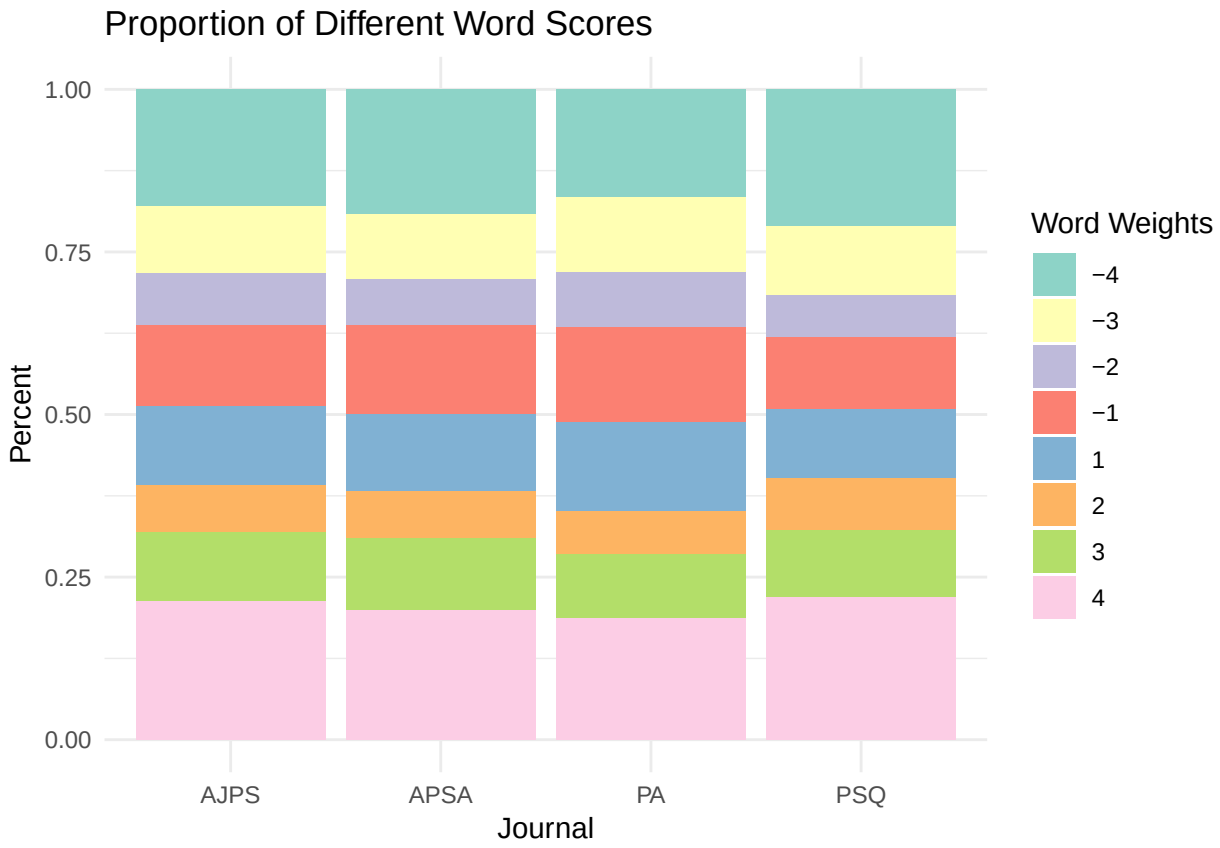
```
full_texts %>%
  group_by(month = floor_date(date, "month"), source) %>%
  ggplot(., aes(month, score)) +
  geom_line(aes(color = source))+
  ggtitle("Altmetric Scores Over Time")
```

```
## Warning: Removed 11 rows containing missing values (geom_path).
```



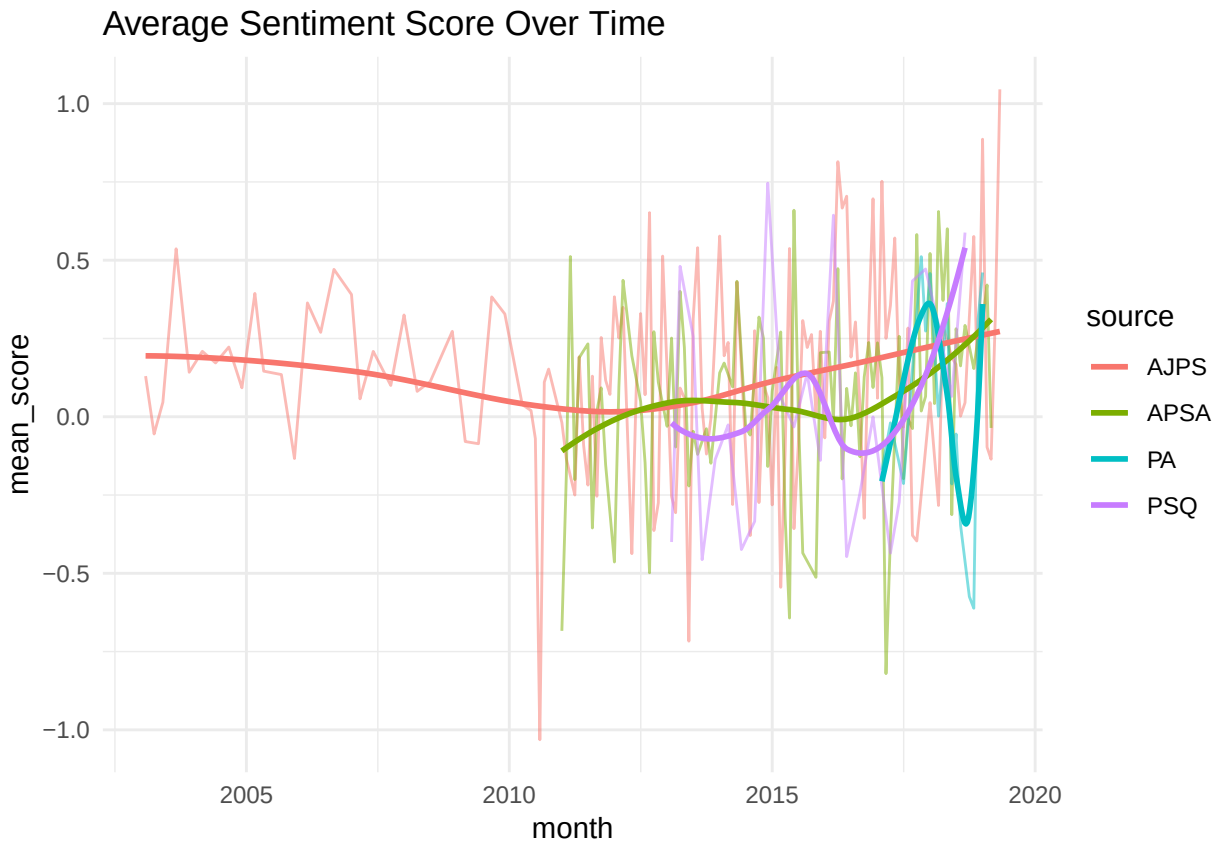

```
# prop of dfferent sentiments
tidytexts %>%
  filter(!is.na(sentiment)) %>%
  ggplot(., aes(source, fill = sentiment)) +
  geom_bar(position = "fill") +
  scale_fill_brewer(palette = "Set3") +
  xlab("Journal") +
  ylab("Percent") +
  ggtitle("Proportion of Different Sentiments") +
  theme_minimal()
```


```
# Prop of Different Word Weights
tidytexts %>%
  filter(!is.na(sent_score)) %>%
  group_by(source) %>%
 # summarize(mean_sent_score = mean(sent_score)) %>%
  ggplot(., aes(source, fill = as.factor(sent_score))) +
  geom_bar(position = "fill") +
  scale_fill_brewer(palette = "Set3") +
  xlab("Journal") +
  ylab("Percent") +
  ggtitle("Proportion of Different Word Scores") +
  theme_minimal() +
  guides(fill = guide_legend(title = "Word Weights"))
```



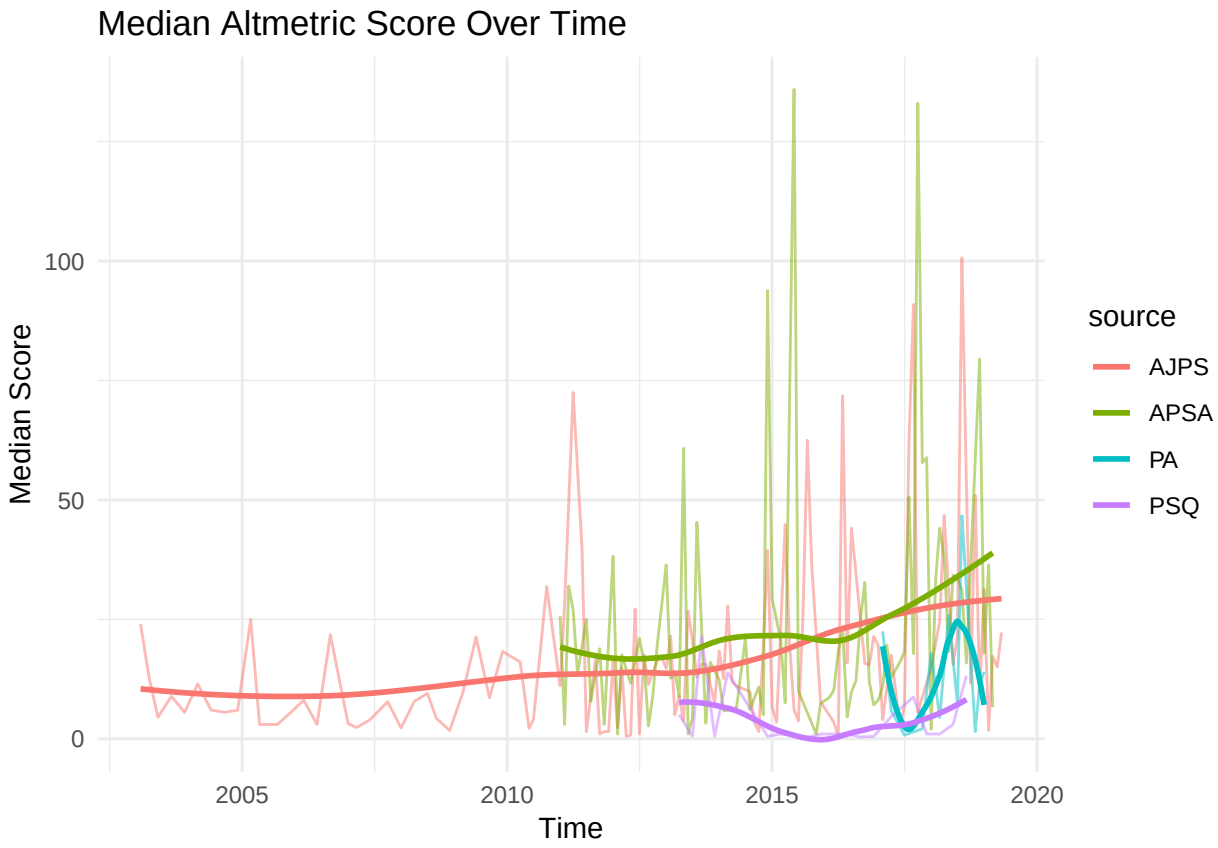

```
# avg sentitment score over time
tidytexts %>%
  filter(!is.na(sent_score)) %>%
  group_by(month = floor_date(date, "month"), source) %>%
  summarize(mean_score = mean(sent_score)) %>%
  ggplot(., aes(month, mean_score))+
  geom_line(aes(group = source, color = source), alpha = .5) +
  geom_smooth(aes(group = source, color = source, weight = 2), se = F)+
  theme_minimal() +
    ggtitle("Average Sentiment Score Over Time")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#altmetric score over time
full_texts %>%
  filter(!is.na(score)) %>%
  group_by(month = floor_date(date, "month"),  source, year = floor_date(date, "year")) %>%
  summarize(med_score = median(score),
            total_tweets = sum(cited_by_tweeters_count)) %>%
  ggplot(., aes(month, med_score)) +
  geom_line(aes(color = source), alpha = .5) + theme_minimal() +
  geom_smooth(aes(color = source), se = F) +
  ggtitle("Median Altmetric Score Over Time") +
  xlab("Time") +
  ylab("Median Score")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

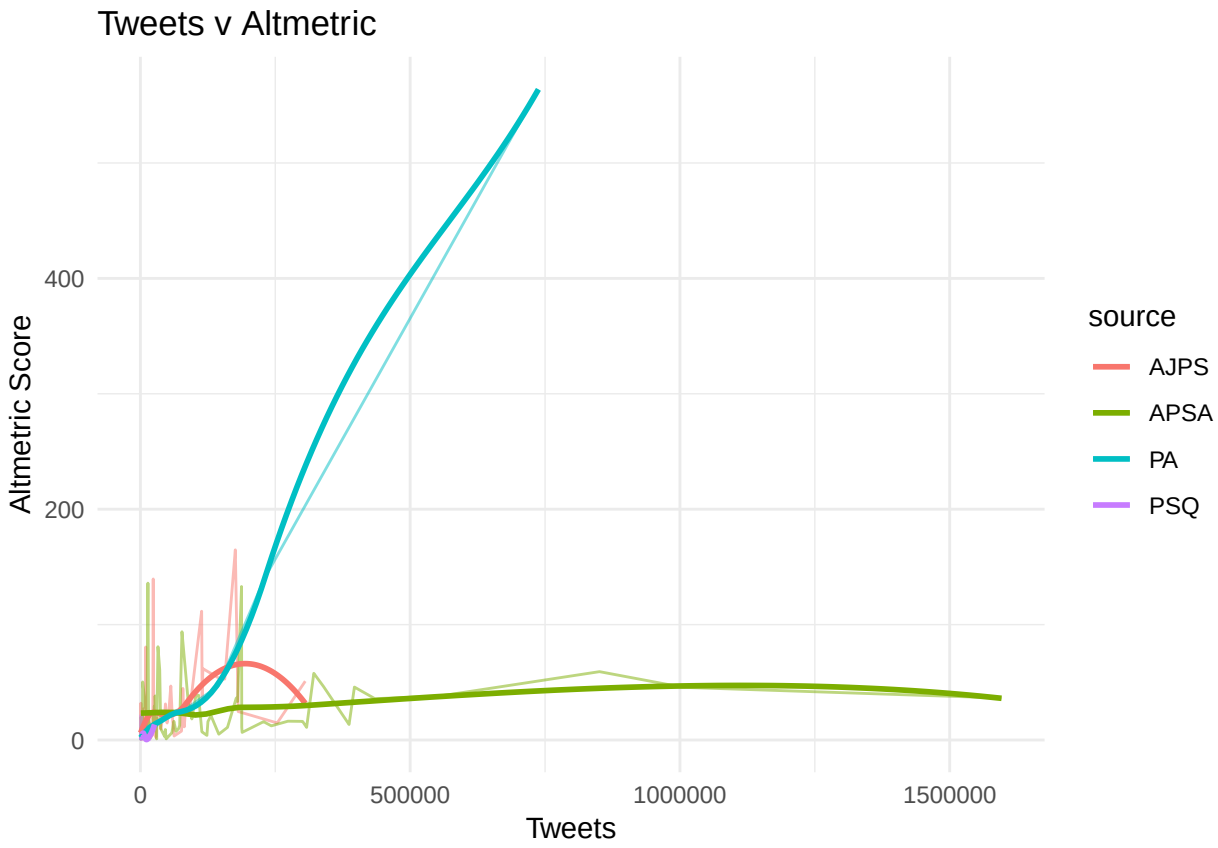



```
# sentiment score versus altmetric score
tidytexts %>%
  filter(!is.na(sent_score)) %>%
  filter(!is.na(score)) %>%
  group_by(month = floor_date(date, "month"),  source, year = floor_date(date, "year")) %>%
  summarize(med_score = median(score),
            total_tweets = sum(cited_by_tweeters_count),
            mean_sent_score = mean(sent_score)) %>%
  ggplot(., aes(total_tweets, med_score)) +
  geom_line(aes(color = source), alpha = .5) +
  geom_smooth(aes(color = source, weight = 2), se = F)+
  theme_minimal()+
  ggtitle("Tweets v Altmetric") +
  ylab("Altmetric Score")+
  xlab("Tweets")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

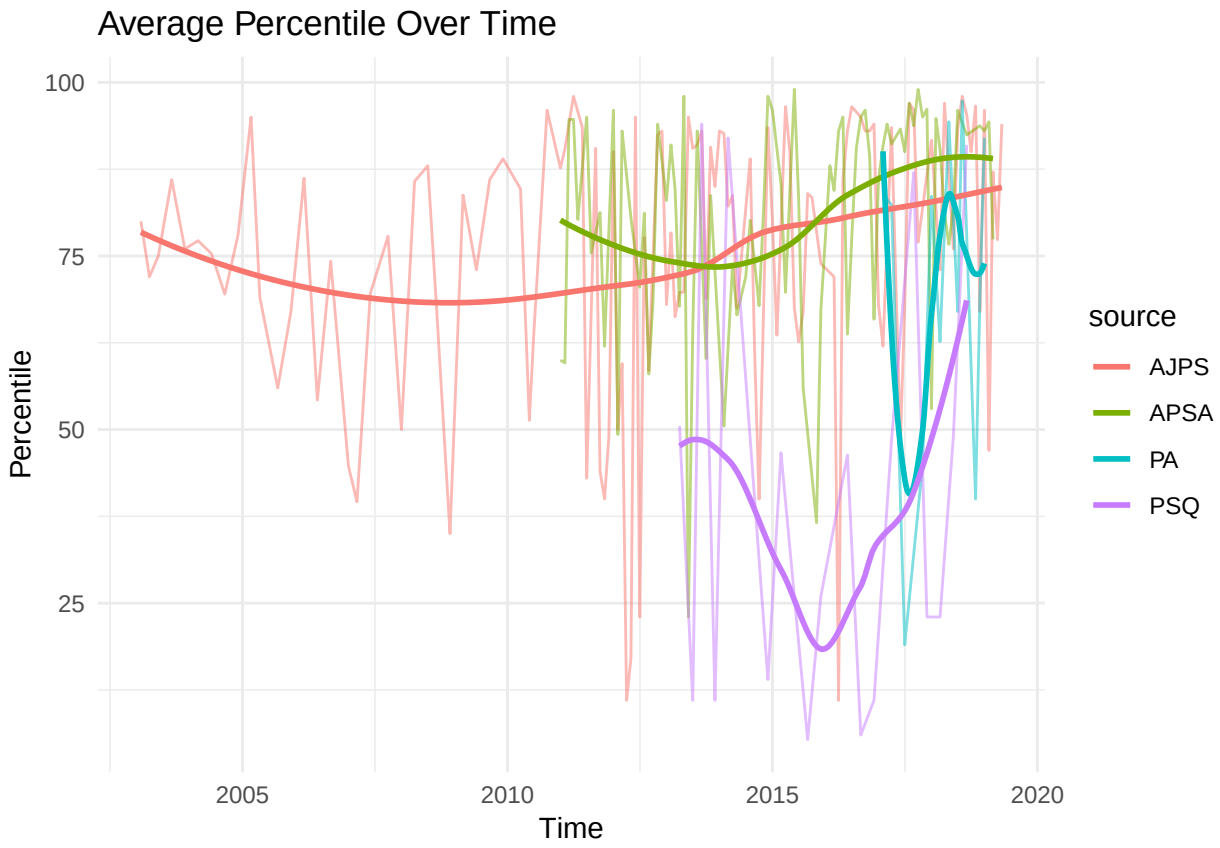
```
## Warning: Removed 46 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 46 rows containing missing values (geom_path).
```




```
# percentile all time over time
full_texts %>%
  filter(!is.na(score)) %>%
  group_by(month = floor_date(date, "month"),  source, year = floor_date(date, "year")) %>%
  summarize(med_score = median(score),
            avg_pct = mean(context.all.pct),
            ) %>%
  ggplot(., aes(month, avg_pct)) +
  geom_line(aes(color = source), alpha = .5) +
  geom_smooth(aes(color = source, weight = 2), se = F) +
  theme_minimal() +
  ggtitle("Average Percentile Over Time") +
  ylab("Percentile") +
  xlab("Time")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

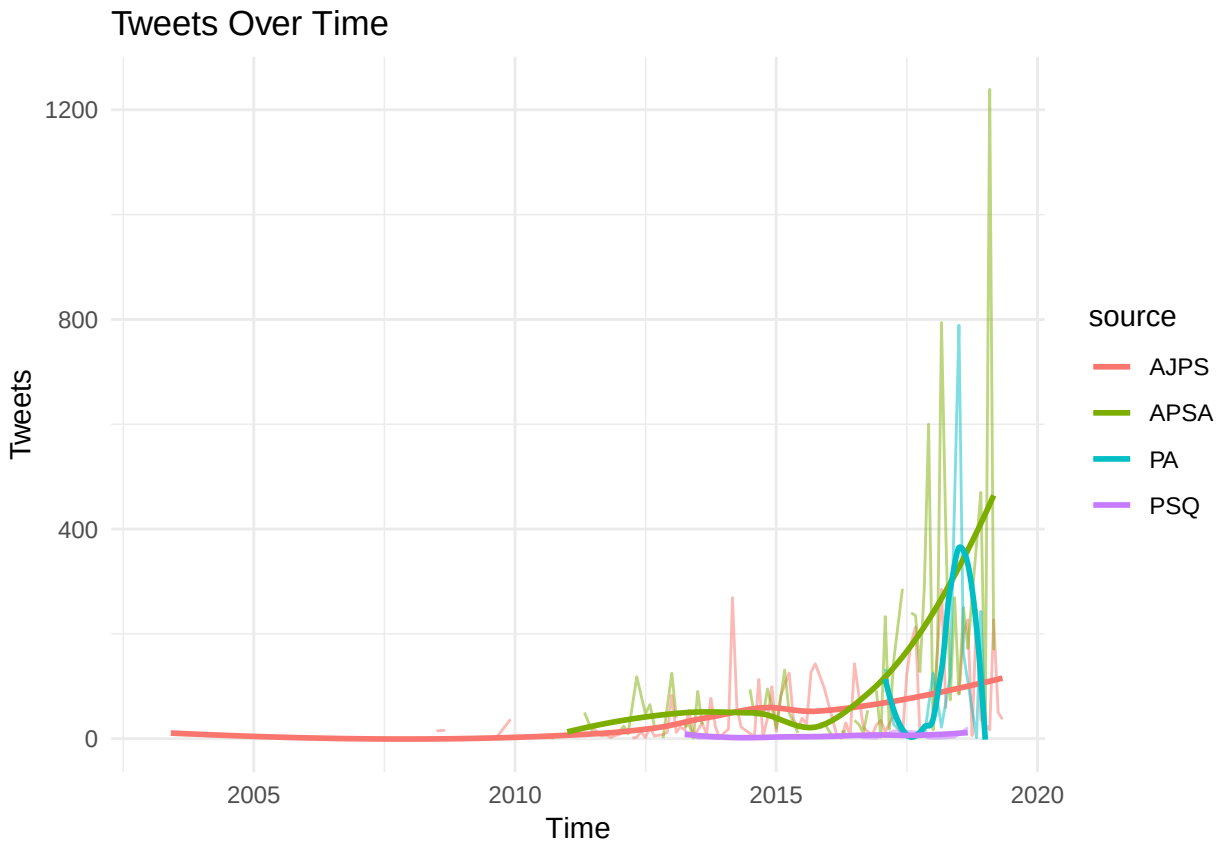



```
# Tweets over time
full_texts %>%
  filter(!is.na(score)) %>%
  group_by(month = floor_date(date, "month"),  source, year = floor_date(date, "year")) %>%
  summarize(med_score = median(score),
            sum_tweets = sum(cited_by_tweeters_count),
            ) %>%
  ggplot(., aes(month, sum_tweets)) +
  geom_line(aes(color = source), alpha = .5) +
  geom_smooth(aes(color = source, weight = 2), se = F) +
  theme_minimal() +
  ggtitle("Tweets Over Time") +
  ylab("Tweets") +
  xlab("Time")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 46 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
# Top articles
full_texts %>%
  select(-text) %>%
  select(date, title, source, score) %>%
  arrange(desc(score)) %>%
  head()
```

```
## # A tibble: 6 x 4
##   date       title                                                source score
##   <date>     <chr>                                                <chr>  <dbl>
## 1 2018-07-31 Gendered Citation Patterns across Political Scie~ PA      564.
## 2 2018-03-22 Bias in Perceptions of Public Opinion among Poli~ APSA    488.
## 3 2019-02-19 Local News and National Politics                  APSA    376.
## 4 2014-03-05 Conspiracy Theories and the Paranoid Style(s) of~ AJPS    339.
## 5 2014-09-02 Assortative Mating on Ideology Could Operate Thr~ AJPS    272.
## 6 2013-01-22 When Are Women More Effective Lawmakers Than Men? AJPS    244.
```

```
# Average altmetric score
full_texts %>%
  filter(!is.na(score)) %>%
 # count(source)
  group_by(source) %>%
  summarize(mean_score = mean(score))
```

```
## # A tibble: 4 x 2
##   source mean_score
##   <chr>       <dbl>
## 1 AJPS         23.3
```

```
## 2 APSA        28.6
## 3 PA          33.5
## 4 PSQ          4.35
```

```
# average sentiment score
tidytexts %>%
  filter(!is.na(sent_score)) %>%
  group_by(source) %>%
  summarize(mean_score = mean(sent_score))
```

```
## # A tibble: 4 x 2
##   source mean_score
##   <chr>       <dbl>
## 1 AJPS      0.128
## 2 APSA      0.0503
## 3 PA       -0.00869
## 4 PSQ       0.0567
```

Summary of Findings

The Political Analysis had the lowest sentiment score but the highest Altmetric score and had the most methodology-related terms. The Political Science Quarterly had one of the highest sentiment scores and the lowest Altmetric score, making it the journal scoring in the lowest percentile on average. It additionally had the most security related terms. All journals had relatively the same word "weight", so one journal was not significantly different in sentiment scores. The American Journal of Political Science and American Political Science Association had many popular articles overtime, possibly considered as outliers.

Unsupervised topic modeling can possibly be used in the future to analyze what journal articles are tweeted or talked about the most, as shown the tweets over time plot, the Political Analysis had articles frequently discussed. However, as shown by the American PS Association in the Tweets versus Altmetric score plot, an article that's tweeted more doesn't necessarily mean the article will have a higher score. Furthermore, articles that are discussed more does not mean they are always good articles, they could be discussed because they are very controversial or poorly researched.

Overall, I want to find a way to streamline my webscraping process because it is very time-consuming and I was not able to obtain all doi information for all articles. I also want to incorporate regular news articles like the New York Times, which I initially started doing but then realized the massive volume of articles put out by the NYT. Roughly 200,000 articles equated to about two months of articles. This massive amount of text would be computationally difficult to process and parse for contextual information. I additionally want to find a way to contextualize the content of articles further so I know what authors discuss. LDA can potentially help here. Lastly, I'm attending the "Text as Data" workshop on May 28th to hopefully gain insight on how to quantitatively analyze text.