POL290G Module 2 HW

Aaron Guerra

Discussion Question

1. Why is understanding the data generation process (DGP) crucial when working with text data? How can overlooking aspects of DGP lead to biased or misleading analysis outcomes? (You'll probably need a few paragraphs for this response.)

In any situation where you gather data, there are numerous assumptions that you make. In survey data, the sample you take and the structure of the survey have significant impacts on how internally and externally valid your results are. When working with text data, however, many of these assumptions may be obfuscated and are different from the assumptions you make in other forms of data collection. For example, a representative sample of the population in traditional statistics is well documented the repercussions of violating these assumptions are clear. However, in text data collection the availability of data and the forms in which it comes mean that traditional approaches to random sampling may not work in the same way. For example, collection of an entire corpus is often possible and in many cases, has already been undertaken by other researchers. There are still several avenues of selection bias here, particularly when collection was done by others, and these can be easy to overlook.

Another assumption to be concerned about with the data generation process for text data is the measurement of variables of interest. In contrast to experimental data or survey data, the variable of interest from a text dataset is unlikely to be directly observable, and often must be subjectively interpreted. For example, measuring sentiment in a text is significantly more subjective (and reflective of the researchers biases) than measuring sentiment in a survey (which may be more reflective of the subject's biases). Specifically, the error produced when understanding if a corpus of tweets are more liberal or more conservative is a product of the researchers and how they interpret the text, whereas the error produced when surveying a population on if they are more liberal or conservative may be significantly more a product of the respondents and how they interpret the question. In this case, overlooking researcher bias in the text sentiment analysis may bias the data produced in the direction of the researcher's expectations.

Finally, there are considerations from linguistics that suggest that language is not deterministic nor is it entirely reducable to a set of ordered rules which are always followed. As such, when analyzing text as data there is an element of error or noise which is not present in other forms of data and may require significant considerations. For example, slang or colloquial terms require context, jargon may be used, and abbreviations can cause uncertainty in analysis, and in each of these situations, there is not necessarily a definitively correct answer. As such, models used should be able to account for this additional noise, as failing to do so would result in attributing noise to phenomena when that is not the true cause. In my own research, abbreviation disambiguation (for example, the Department of Water Resources vs. DWR) requires manual coding, which can be prohibitive when looking at 120 groundwater management plans from across the state which have a very large and sometimes overlapping set of acronyms. However, failing to account for this underweight the importance of 'Department of Water Resources' as there are references to 'DWR' which are not captured under certain methods of text analysis.

Project Questions

1. Find an API (other than NYT) that has data your are interested in.

2. Write R code to access the API, using one search term (e.g., "immigration").

3. Do some basic data exploration. Depending on your data, this could be things like word frequencies, temporal trends, etc.

```r
library(tidyverse)
library(lubridate)
library(tidytext)
library(jsonlite)
```

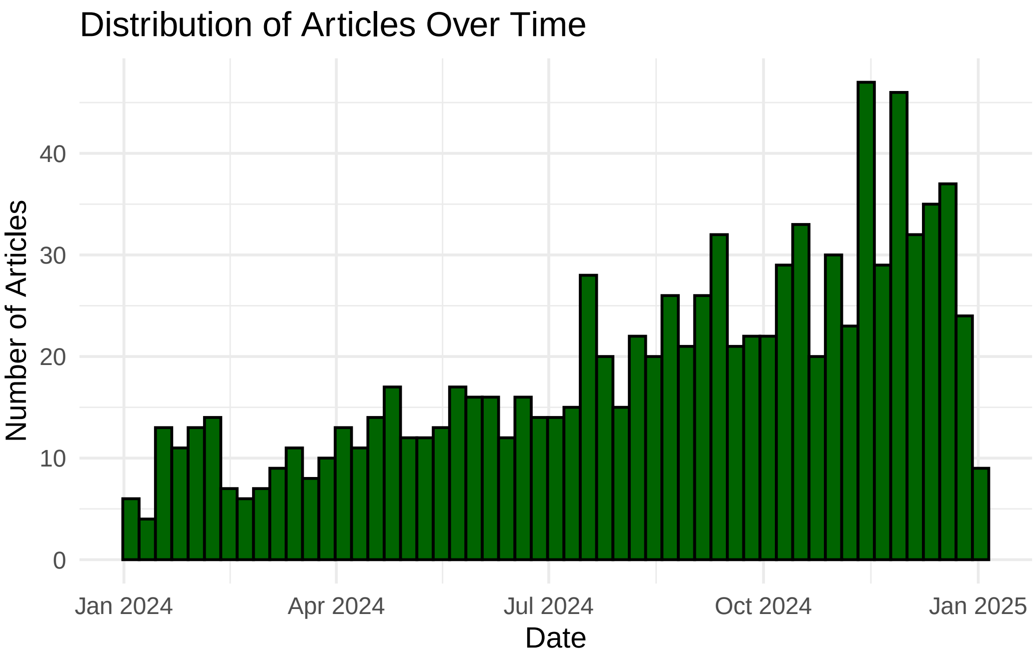
```r
apikey <- 'api-key=4c48d425-21db-402c-b2ee-9fd1fe5ae1bb'

baseurl <- 'https://content.guardianapis.com/'

search <- 'search?q=environmental&'

dates <- 'from-date=2024-01-01&to-date=2024-12-31&'

pages <- 1:100

queries <- list()

for (page in pages) {
  pagex <- paste0('page=', page, '&')
  url <- paste0(baseurl, search, dates, pagex, apikey)
  query <- fromJSON(url, flatten = TRUE)
  queries[[page]] <- query
}

save(queries, file = "queries.RData")
```

```r
load("queries.RData")

df <- tibble()

for (q in 1:length(queries)){
  dfx <- queries[[q]]$response$results %>%
    select(id, webTitle, webPublicationDate)

  df <- rbind(df, dfx)
}

env_df <- df %>%
  mutate(date = as_date(ymd_hms(webPublicationDate, tz= "UTC"))) %>%
  filter(date > "2024-01-01" & date < "2025-01-01") %>%
  select(webTitle, date)

head(env_df)
```

```
                                                                     webTitle
1                  Lloyds advert banned for making false environmental claims
2    ‘Considerable environmental damage’ from Victoria blaze – as it happened
3                  Environmental grants promised to farmers in England frozen
4 US environmental agency fast-tracking new PFAS approvals for semiconductors
5        Russia warns of severe environmental damage from Black Sea oil spill
6  Environmental groups demand EPA to start monitoring microplastics in water
        date
1 2024-12-18
2 2024-12-26
3 2024-11-26
4 2024-12-19
5 2024-12-28
6 2024-12-01
```

In this chunk, I asked Claude 3.7 Sonnet - “how do i convert this to a date in R 2025-04-03T13:00:01Z”. Useful because there are so many lubridate functions and I never know which to use for which

```r
ggplot(env_df, aes(x = date)) +
  geom_histogram(binwidth = 7, fill = "darkgreen", color = 'black') +
  labs(title = "Distribution of Articles Over Time",
       x = "Date",
       y = "Number of Articles") +
  theme_minimal()
```



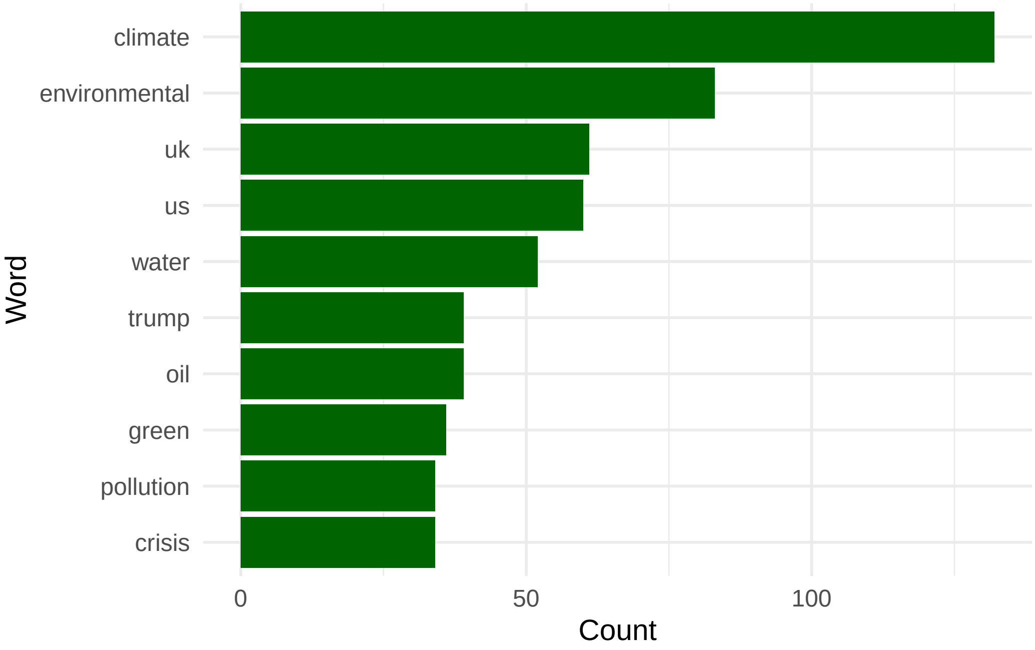
```r
env_df %>%
  group_by(date) %>%
  summarize(n=n()) %>%
  arrange(desc(n))
```

```
# A tibble: 317 x 2
   date           n
   <date>     <int>
 1 2024-12-11    12
 2 2024-11-14    11
 3 2024-11-29    10
 4 2024-11-15     9
 5 2024-11-06     8
 6 2024-11-12     8
 7 2024-11-21     8
 8 2024-11-26     8
 9 2024-11-27     8
10 2024-12-01     8
# i 307 more rows
```

```r
stop_words <- stop_words %>%
  filter(word != 'us')

words <- env_df %>%
  unnest_tokens(word, webTitle) %>%
  anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```r
top_words <- words %>%
  count(word, sort = TRUE) %>%
  head(10)

ggplot(top_words) +
  geom_col(aes(x = reorder(word, n), y = n), fill = "darkgreen")+
  coord_flip() +
  labs(x = "Word",
       y = "Count") +
  theme_minimal()
```



4. Write a summary of (1) the data source you chose, (2) what you learned from your basic data exploration, and (3) any challenges you faced along the way - like finding/accessing a relevant API, getting permissions to use the API (RIP Twitter), and any issues with processing your text data.

I chose to work with the Guardian API data. This covers the news articles posted by the Guardian, which I subset to those posted within 2024. I then filtered the data to only include articles that contained the word “environmental”, which reflects my interest in environmental policy, but should also reflect general environmental news.

To do basic data exploration, I took two approaches. First, I created a histogram of the number of articles posted over time. This histogram shows an increase in articles covering the environment in the later half of the year, with a particular spike in late November. A closer analysis of these dates (November 26th, 27th, and 28th) shows this is perhaps coincidental - there is no specific theme connecting them.

Second, I created a bar chart of the most common words in the titles of these articles. Of these most common terms, ‘climate’, ‘water’, ‘oil’, ‘green’, ‘pollution’, refer to specific environmental topics, suggesting these are the most prevalent or salient issues covered. By comparison, ‘UK’, ‘Trump’, ‘crisis’, and ‘study’ refer to the political dimensions of the environment. One omission on my first pass was that ‘UK’ was not contrasted by ‘US’, because ‘us’ is considered a stop word (see below). Upon revision, you see that ‘UK’ is ever so slightly more common than ‘US, assuming that ’us’ is not used in any article titles (which is not a flawless assumption).

In terms of challenges, I found out after completing my code with accessing the API and processing the JSON that there is a package in R specifically for this API. Some of my classmates worked using this and it certainly seemed easier to use. However, I’m glad I went through the sticky work of figuring out how to work with the API and the issues that can come up there. For example, having to loop through several pages of data, since the Guardian has limited articles per page.

5. Repeat steps 1-3 but this time with additional search terms (e.g., “immigration”, “immigrant”, “migrant”).

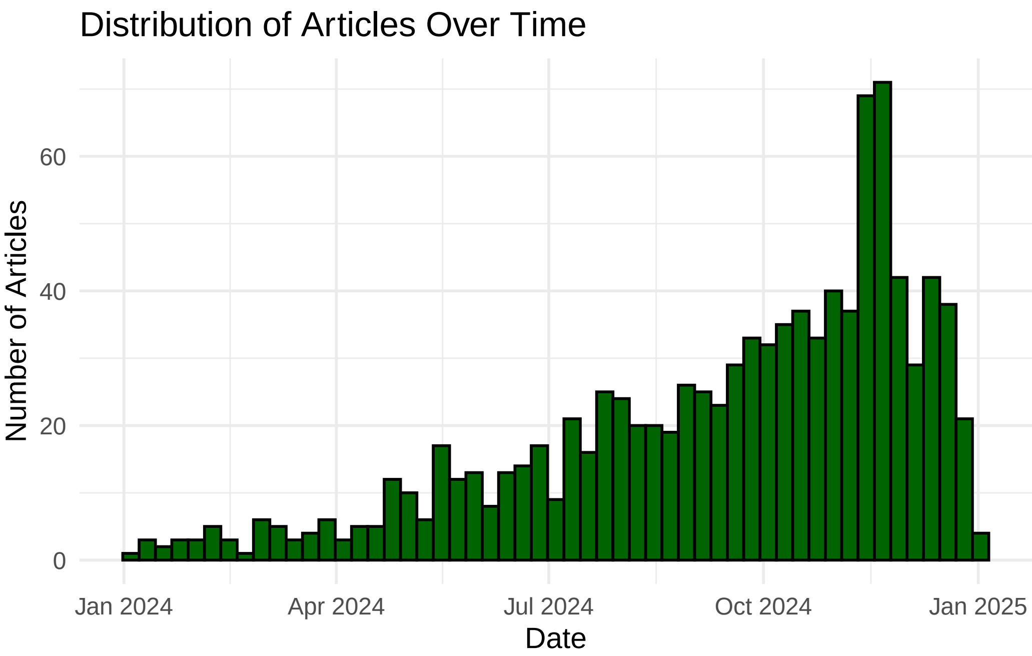
```r
apikey <- 'api-key=4c48d425-21db-402c-b2ee-9fd1fe5ae1bb'

baseurl <- 'https://content.guardianapis.com/'

search <- 'search?q=climate%20OR%20water%20OR%20pollution%20OR%20oil%20OR%20green&'

dates <- 'from-date=2024-01-01&to-date=2024-12-31&'

pages <- 1:100
```

```
queries2 <- list()

for (page in pages) {
  pagex <- paste0('page=', page, '&')
  url <- paste0(baseurl, search, dates, pagex, apikey)
  query <- fromJSON(url, flatten = TRUE)
  queries2[[page]] <- query
}

save(queries2, file = "queries2.RData")
```

```
load("queries2.RData")

df2 <- tibble()

for (q in 1:length(queries2)){
  dfx <- queries2[[q]]$response$results %>%
    select(id, webTitle, webPublicationDate)

  df2 <- rbind(df2, dfx)
}

env_df2 <- df2 %>%
  mutate(date = as_date(ymd_hms(webPublicationDate, tz= "UTC"))) %>%
  filter(date > "2024-01-01" & date < "2025-01-01") %>%
  select(webTitle, date)

head(env_df2)
```

```
1                                How climate policies reduce air pollution saving lives
2                       ‘Climate-induced poisoning’: 350 elephants probably killed by to
3           Revealed: water firms in England ‘passed’ pollution tests that were never ca
4                     Oil pollution in UK waters far worse than reported, says conservat
5 ‘It won’t wash’: Environment secretary’s bid to cosy up to water pollution protesters
6                                      BP ‘abandoning plan to cut oil output’ angers gre
        date
1 2024-11-29
2 2024-11-29
3 2024-10-26
4 2024-09-26
5 2024-10-31
6 2024-10-07
```

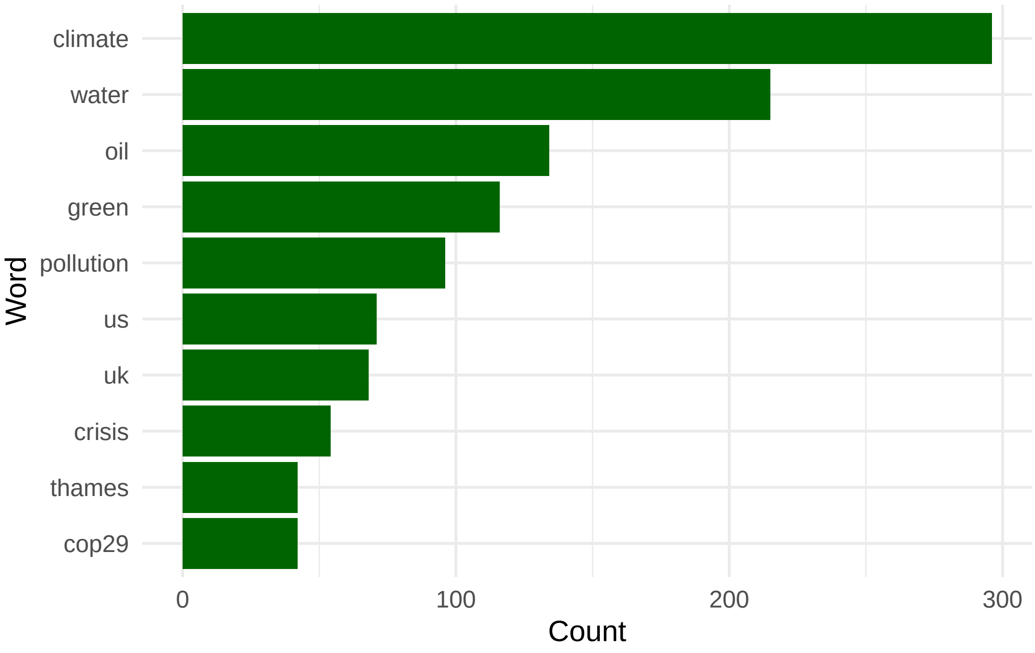
```r
ggplot(env_df2, aes(x = date)) +
  geom_histogram(binwidth = 7, fill = "darkgreen", color = 'black') +
  labs(title = "Distribution of Articles Over Time",
       x = "Date",
       y = "Number of Articles") +
  theme_minimal()
```



```r
words2 <- env_df2 %>%
  unnest_tokens(word, webTitle) %>%
  anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```r
top_words2 <- words2 %>%
  count(word, sort = TRUE) %>%
  head(10)

ggplot(top_words2) +
  geom_col(aes(x = reorder(word, n), y = n), fill = "darkgreen")+
  coord_flip() +
  labs(x = "Word",
       y = "Count") +
  theme_minimal()
```



6. Finally, discuss: what worked and what didn't, and why? Explain the similarities and differences between your two search term approaches and what you learned.

In my second pass, I decided to instead focus on the specific environmental words that were highlighted in the first pass. I used the same code as before but instead searched for climate, water, pollution, oil, and green. I actually initially did this write up based on erroneous code - after submitting and working on my questions for lecture, I realized that I had mistakenly used 'AND' to connect my search terms instead of 'OR', which accounted for a much smaller dataset. This just highlights the importance of being thorough and double checking your code :).

In this second-pass dataset, some of the same trends are apparent . The spike in interest in late November is still present, and several of the same words are present in the top 10 words. In this dataset, we still see roughly equal representation of 'US' and 'UK', however, 'US' now has slightly more representation. The five search terms all make up the five most common words, which makes sense given that these words correspond to how the search is conducted. Two new terms appear in the top 10 - 'thames' and 'cop29', while 'crisis' continues to appear. As was suggested in the Barbera et al paper, the methods of search (in this case, one keyword versus a more specific set of keywords) have a large impact on the results.