NYT Analysis

Keshav Khullar

5/3/2020

Summary and Interest:

I have chosen to evaluate subject matter through the New York Times (NYT) API, specifically, the prevelance of Coronavirus and Donald Trump in the NYT headlines across three months. I chose January/February/March because in that time rhetoric in the news changed from being about Donald Trump's impeachment to coverage of the Coronavirus.

The objective of this report is to explore which had greater influence of new coverage in this tumultuous period - Donald Trump or this coronavirus? In addition to this, exploring the data to show why coronavirus was covered so extensively.

In the following analyses, you will note that I used the API twice, once for an article search of the 'coronavirus' and another for an article search of 'Trump', which ensured that only headlines with those keywords would be shown. I also provided the function used to create the table and a link to the csv of the data extracted itself. As the function worked inconsistently, I provided a link to my github repository for each instance of using the article search. This should ensure that anyone reading this report should be able to run the code that follows, seamlessly. Alternatively, you can run the function. However it, on occasion, lost connection during the for-loop owing to the breadth of articles found on each topic.

Following this, I scraped data from worldometer to collate global statistics on the Coronavirus outbreak and have contructed visualizations to depict the information. This should demonstrate why the coronvirus is being viewed as a pandemic.

```
##api.key.nytimes<-"9qcnAZEv4mHWtMGBNHQxorQWvGrlAbEP"


##The following function converts information
##found through the API and linked URL into a dataframe. jsonlite
##allows us to do this.

##Below we evaluate the keyword "coronavirus"

# function(term){
# term <- "Coronavirus"
# begin_date <- "20200101"
# end_date <- "20200303"
#
#
# Corona.url = paste("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=",ter
m,             "&begin_date=",begin_date,"&end_date=",end_date, "&facet_filter=true&ap
i-key=",api.key.nytimes , sep="")
#
#
#
# first_search=fromJSON(Corona.url,flatten = T)
#
#
# ##To use the for loop below, we require the number of pages to be looped through.
# Total_Pages = round((first_search$response$meta$hits / 10)-1)
#
#
# dataframe <- data.frame(ID=as.numeric(), Time=character(), Snip=character(), Head.l=
character())
#
#
# for(i in 0:Total_Pages){
#     #get the search results of each page
#     Search_nyt = fromJSON(paste0(Corona.url, "&page=", i), flatten = T)
#     temp = data.frame(ID=1:nrow(Search_nyt$response$docs),
#                       Time = Search_nyt$response$docs$pub_date,
#                       Snip = Search_nyt$response$docs$snippet,
#                       Head.l = Search_nyt$response$docs$headline.main)
#     dataframe=rbind(dataframe,temp)
#     Sys.sleep(7) #sleep for 5 second
# }
#
# return(dataframe)
# }
# dataframe
#
# #As requested, the dataframe was written to a CSV.
```

```
#
# write.csv(dataframe, "NYT Coronavirus 3.csv")
# read.csv("NYT Coronavirus 3.csv",header=T,stringsAsFactors = F)

##ALTERNATE TO READ IN  CSV----------------------------------------------BELOW-------


##For Jan
df_Corona <-read.csv("https://raw.githubusercontent.com/kkhullar1/DataWrangling/master/NYT%20Coronavirus%202.csv", encoding = "UTF-8")


##For Jan+Feb
dataframe <-read.csv("https://raw.githubusercontent.com/kkhullar1/DataWrangling/master/NYT%20Coronavirus%203.csv", encoding = "UTF-8")



##-----------------------------------------------------------
```

As mentioned, the above funtion serves to access the archives of NYT articles through the article search API. We use the jsonlite package to interact directly with the API.

The search results are placed into an object using fromJSON. Loops allow us to collect a greater number of results as each page only contains 10 results only.

Explained:
We specified the phrase or ‘term’, the publish date we’re beginning our search from and end of our publish date range. The URL is searched and find a collection of variables put into a flattened list. An empty dataframe is created and these variables (e.g. response.docs.snippet) are selected, searched through and, binded and inserted into into the empty dataframe through the for loop as shown. The Sys.sleep(5) was inserted to indicate to your computer to pause between queries as the API hits a break if too many requests are made. As requested, the dataframe was then written to a CSV file. We carry out this process again for the article search of Trump.
Setting the page number: As mentioned the search only returns 10 results at any given time. However, we query the API to find out how many hits there are. We run this operation to find out the number of pages.
Data Cleaning:

I noticed that the Time column values in the dataframe should be truncated to either Year/Month or Year/Month/Day. I have done so in my code. I conducted both operations for the purpose of two visualisations that utilize each version of the dates. I did not use mutate() to augment my tables instead I added and deleted columns through multiple functions as the logic I used was easier for a reader to follow and comment on, consdering I intended on reproducing this process for the nex API search.

```
##Data Cleaning


##Format Date into Year, Month and Day for Date specific Visualisation
dates<-dataframe$Time
x<-as.POSIXct(dates)
betterDates2 <- format(as.Date(dates),
  "%Y-%m-%d")


##Format Date into Year and Month for One Visualisation
dates<-dataframe$Time
x<-as.POSIXct(dates)
betterDates <- format(as.Date(dates),
  "%Y-%m")


##Formating for Jan Data
dates_Corona<-df_Corona$Time
x<-as.POSIXct(dates_Corona)
betterDates_Corona <- format(as.Date(dates_Corona),
  "%Y-%m")




## Add Date column, remove previous Time to simplify table to only the month and year.
These are each placed in a new dataframe for the later visualisation.

dataframe$Date <-  betterDates
dataframe2 <-select (dataframe,-c(Time))




dataframe$Date <-  betterDates2
dataframe3 <-select (dataframe,-c(Time))




df_Corona$Date <-  betterDates_Corona
dataframe2_Corona <-select (df_Corona,-c(Time))



#Converting the headline column from a factor into a character for for visualisation
```

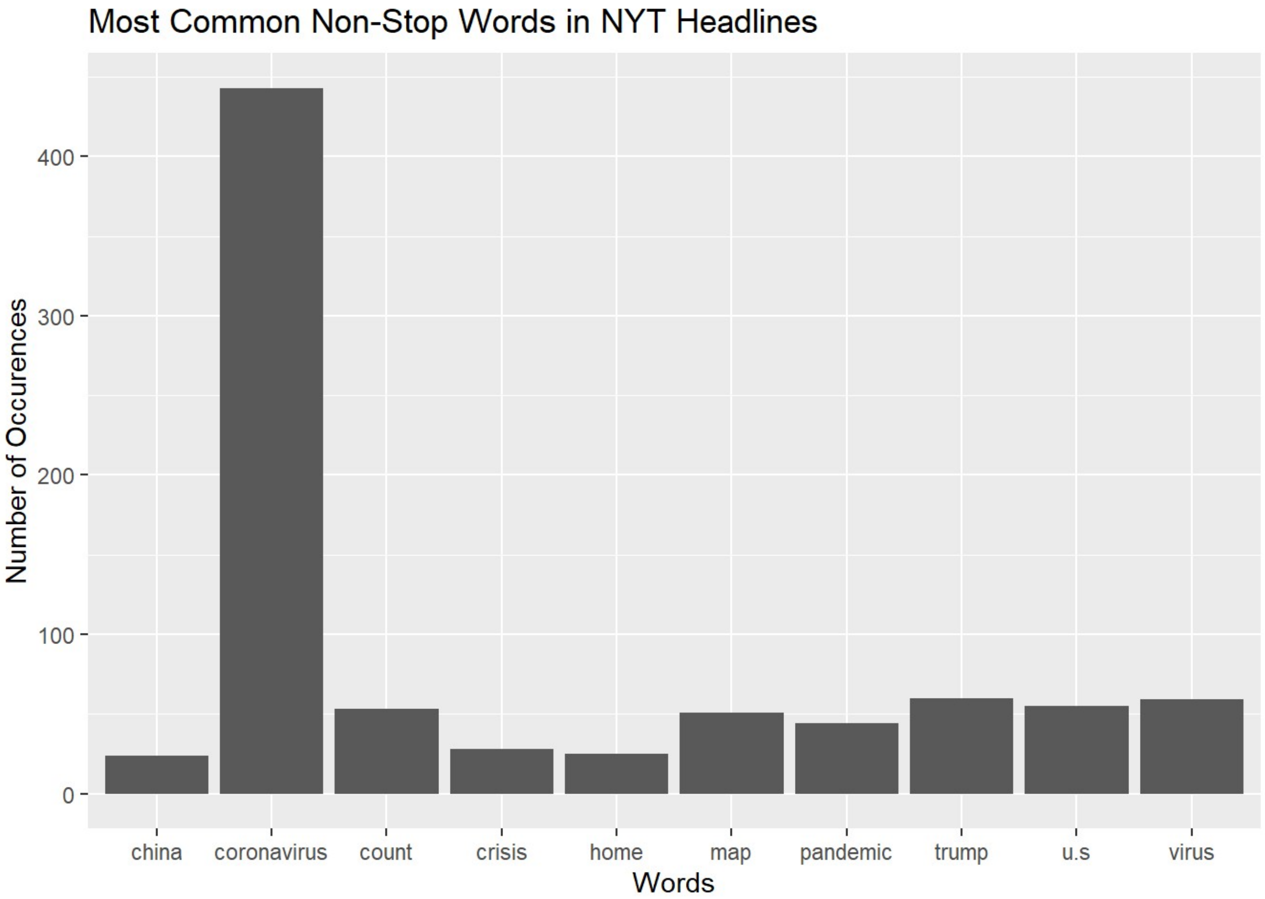
```
dataframe2$Head.l <- as.character(dataframe2$Head.l)

dataframe2_Corona$Head.l <- as.character(dataframe2_Corona$Head.l)
```

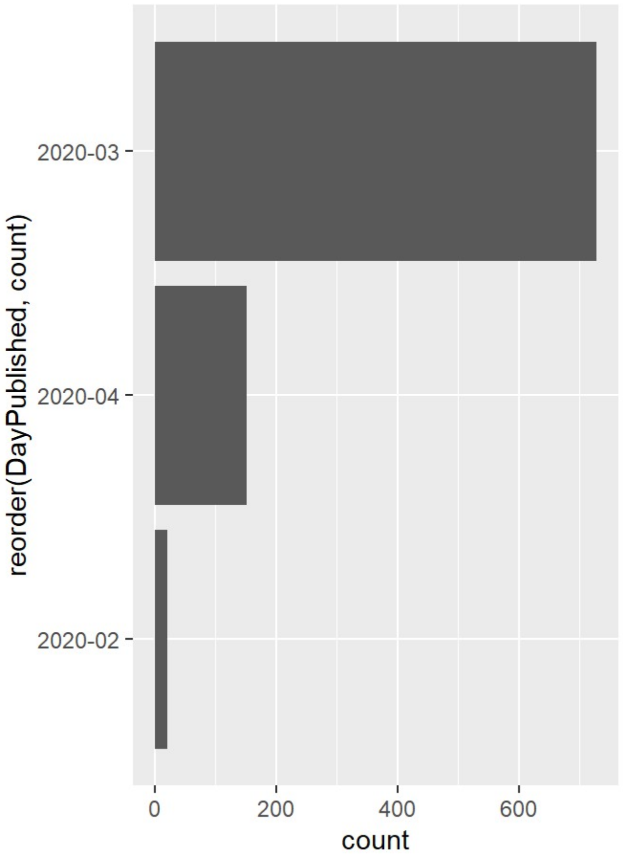
Next, I created visualizations from this cleaned data.

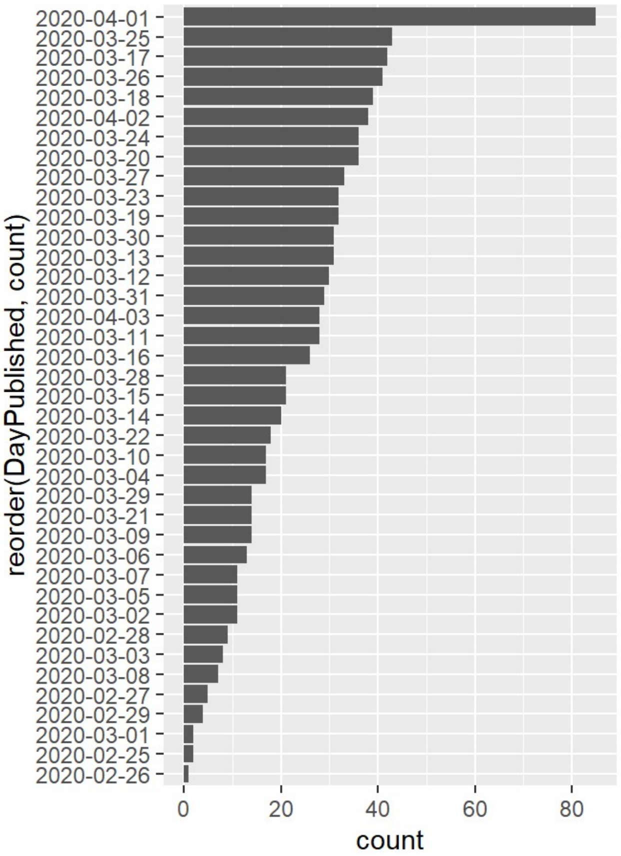
NYT Coronavirus Visualizations

Looking at the most Common Non-Stop Words in NYT Headlines in January/February/March, we see coronavirus was overwhelming, as expected given our search key. However, ignoring the presence of ‘coronavirus’ as a bar, we see impeachment and wuhan getting similar coverage, which suggests that the importance of wuhan was comporable to the impeachment taking place.

Further analysis shows that coverage of impeachment and the electoral process would be eclipsed by coverage of the coronavirus. This is somewhat demonstrated by the horizontal bar charts which shows an increase in articles mentioned about the Coronavirus as time went on. However, this is reflected more clearly by the wordclouds to follow. In addition, we see by the second bigram, based on data from January only, that both impeachment and coronavirus were in the top 10 words. The first bigram reflects the overwhelming discourse about the coronavirus that was taking place in the run up to April


Most Common Non-Stop Words in NYT Headlines
Number of Occurences
0
100
200
300
400
china
coronavirus
count
crisis
home
map
pandemic
trump
u.s
virus
Words



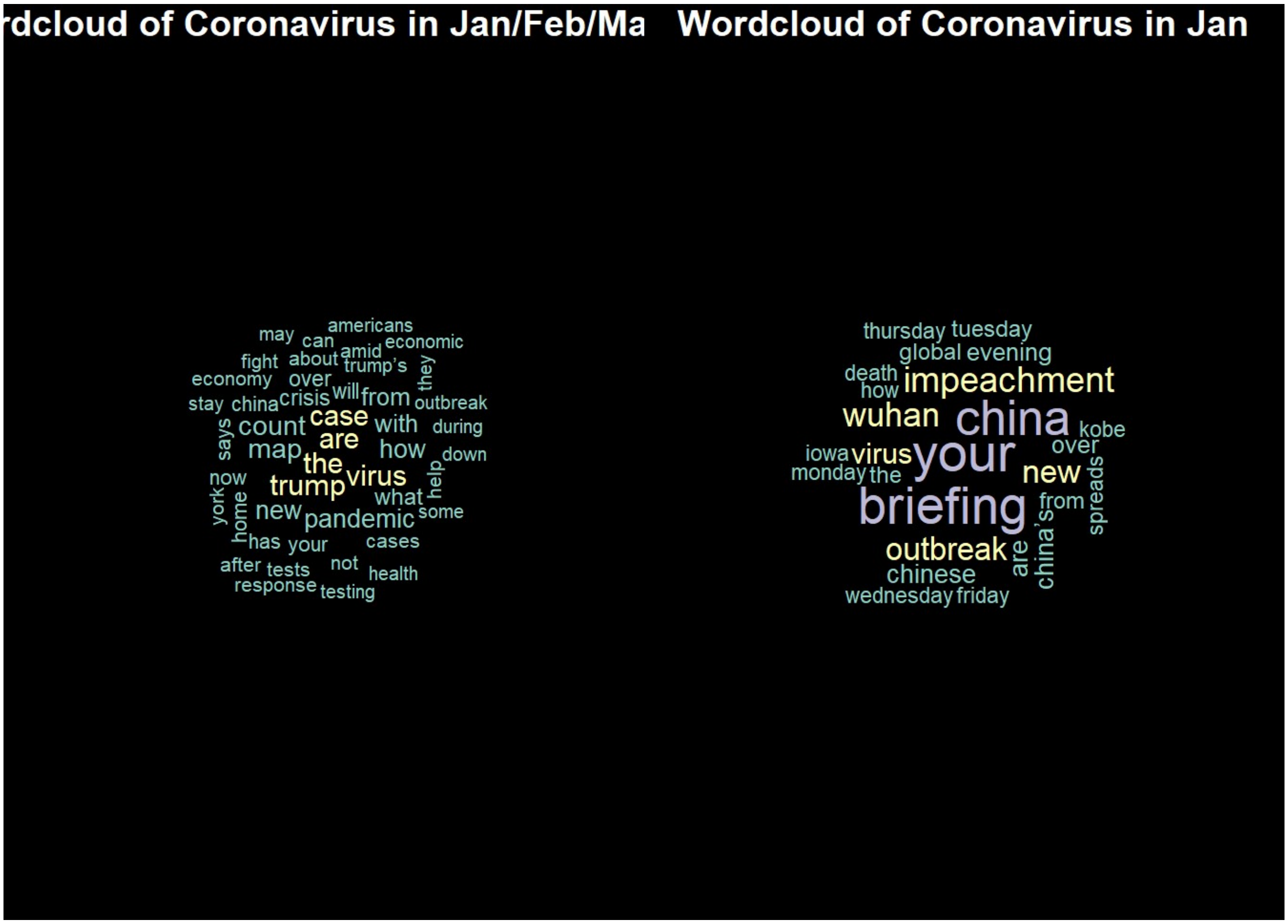


word1 <chr>	word2 <chr>	n <int>
coronavirus	map	50
coronavirus	outbreak	10
amid	coronavirus	9
coronavirus	crisis	9
coronavirus	testing	9
tests	positive	9
white	house	9
coronavirus	concerns	8
coronavirus	spreads	8
coronavirus	fight	6

1-10 of 11 rows Previous **1** 2 Next

word1 <chr>	word2 <chr>	n <int>
wuhan	coronavirus	14
evening	briefing	11
coronavirus	outbreak	10
tuesday	briefing	7
friday	briefing	6
kobe	bryant	6
thursday	briefing	6
wednesday	briefing	6
coronavirus	impeachment	5
coronavirus	spreads	5

1-10 of 11 rows Previous **1** 2 Next



Wordclouds

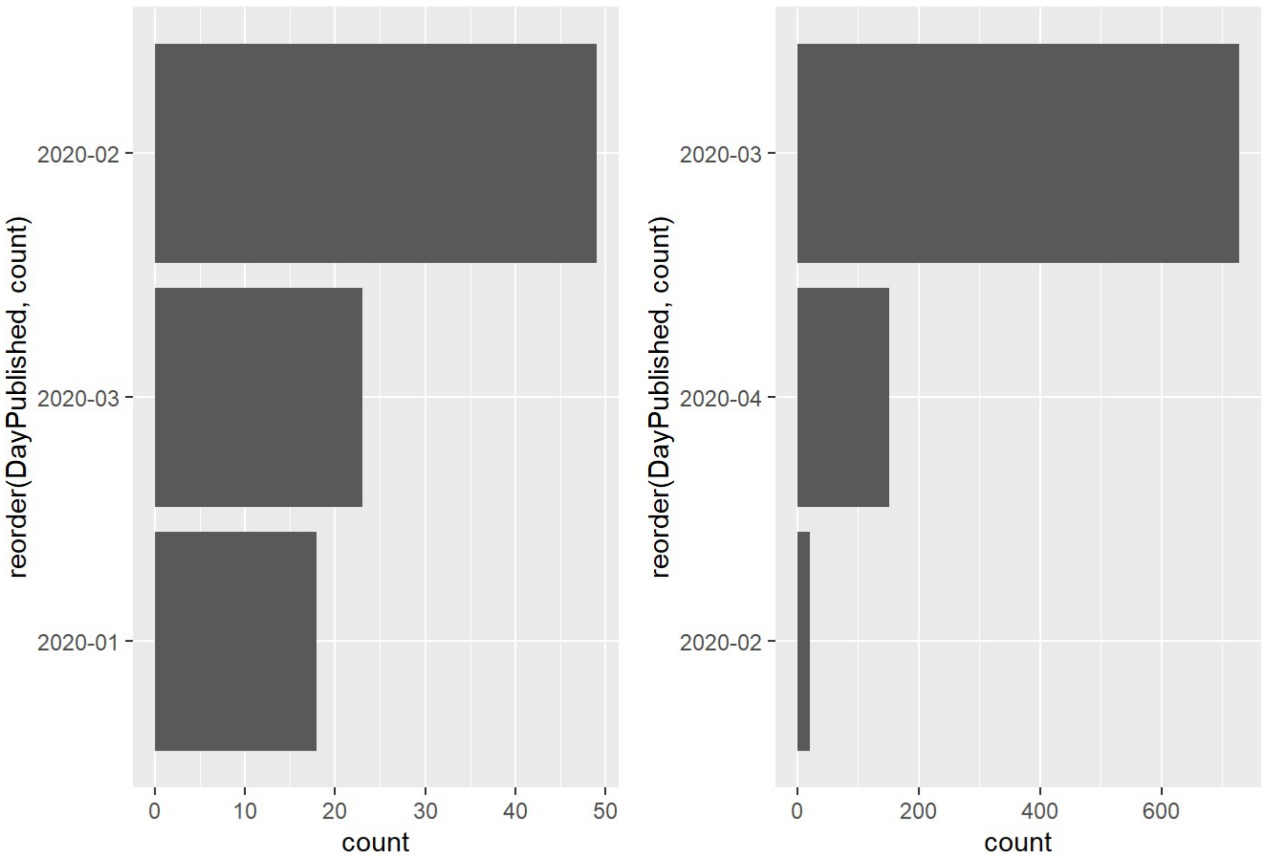
From the above word cloud from Jan, we can see that the impeachment and wuhan were almost equivalent in their being mentioned by the NYT in their headlines. However, China was more prevalent. The concept of a pandemic had not quite surfaced as it still remained as a small word. We also see other popular news at the time including the death of Kobe Bryant, and the Iowa Caucus.

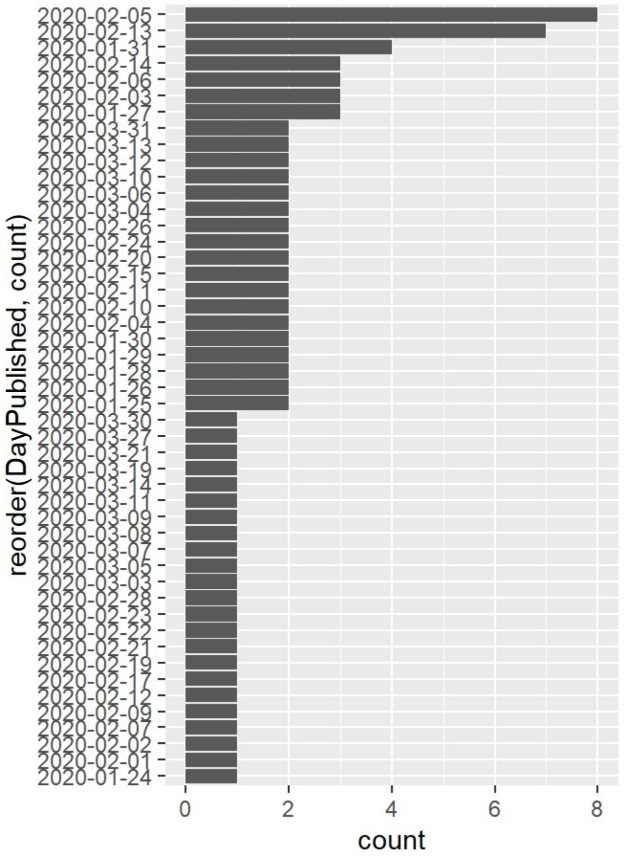
However, by March, we can see that all things non-coronavirus had largely left the NYT headlines. This is further supported by the bigrams, where the presence of ‘impeachment’ as a topic had largely reduced.

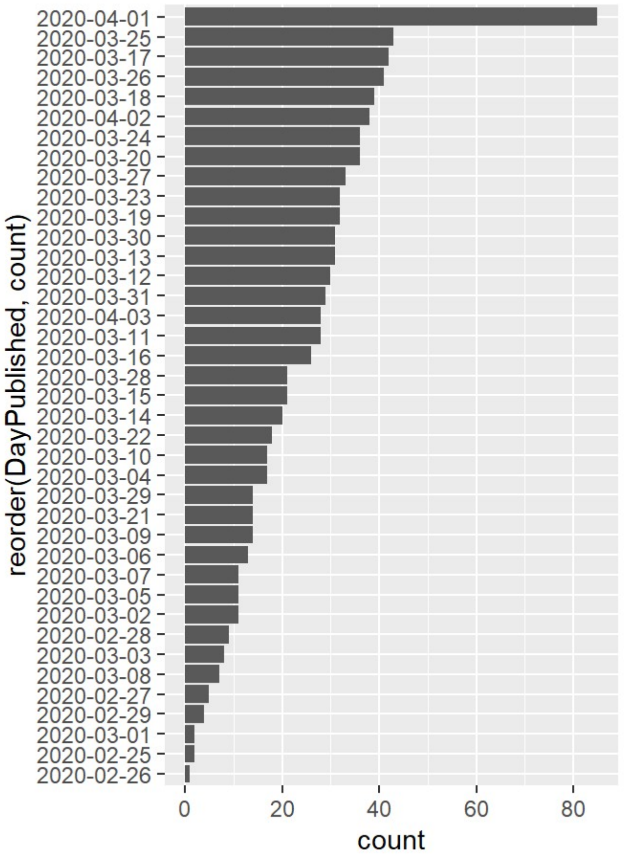
NYT Trump Data Visualizations

Bigrams

We can see from the bigrams that the impeachment trial played an overwhelming role in the Trump related news in January. The bigrams for both Jan and Jan/Feb/March show mentions of the impeachment. However, with the increasing presence of coronavirus in the news, it is understandable that ‘national security’ rose in the ranks as shown by the bigram (without stop-words), over the course of a few months. It should also be noted that in comparing horizontal graphs, side-by-side, we see that the number of articles about coronavirus over time is much greater than Trump related articles. The coronavirus articles count is on the right in the below graphs and the Trump articles count is on the left. I found it challenging to add titles for these particular graphs.


reorder(DayPublished, count)
2020-02
2020-03
2020-01
0
10
20
30
40
50
count
reorder(DayPublished, count)
2020-03
2020-04
2020-02
0
200
400
600
count



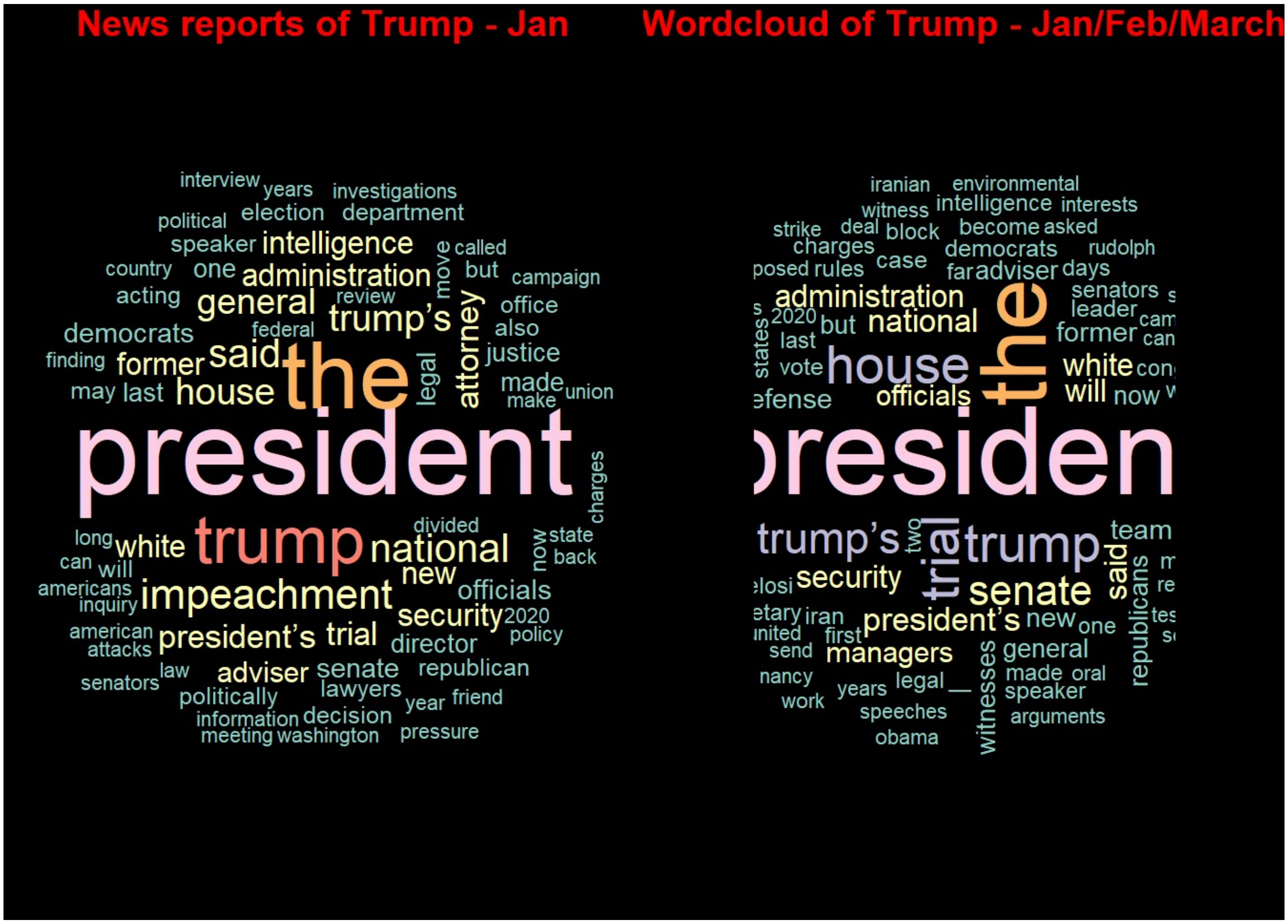


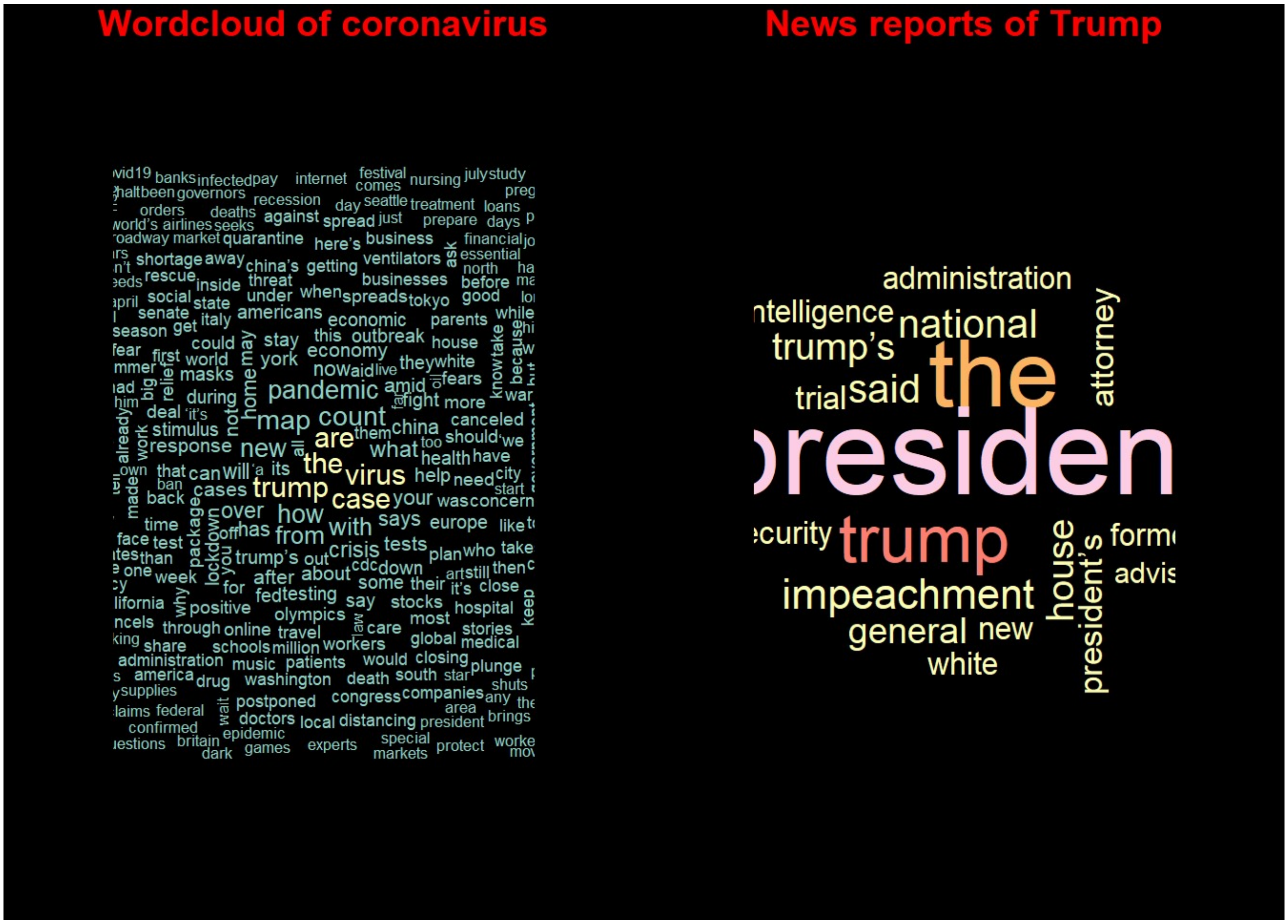
word1 <chr>	**word2** <chr>	**n** <int>
president	trump	14
president	trump’s	14
impeachment	trial	12
white	house	9
national	security	7
security	adviser	6
impeachment	managers	5
trump	administration	4
defense	team	3
house	impeachment	3

1-10 of 15 rows Previous **1** 2 Next

word1 <chr>	word2 <chr>	n <int>
president	trump	16
president	trump’s	11
white	house	8
national	security	7
impeachment	trial	6
trump	administration	6
security	adviser	5
national	intelligence	4
acting	director	3
justice	department	3

1-10 of 10 rows





Wordcloud: The Coronavirus vs. Trump

The second wordcloud, both referencing the Jan/Feb/March dataset, reflect that with regards to articles about the Coronavirus and articles about Donald Trump, while the latter showed, mostly, references to Donald Trump's impeachment process, the former showed signs of the growing concerns of Coronavirus and the influence of the impeachment process in the wordcloud (or presence in new articles) had somewhat dissipated.

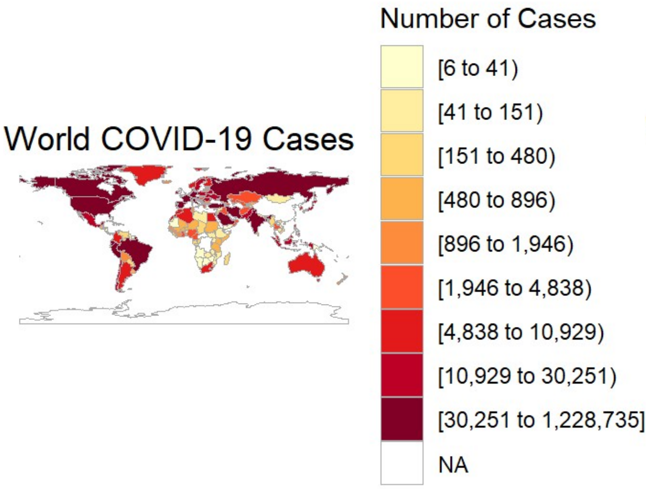
We can demonstrate that the impeachment was more overwhelming in terms of presence in the original wordcloud by the previous coronavirus related segment of the report.

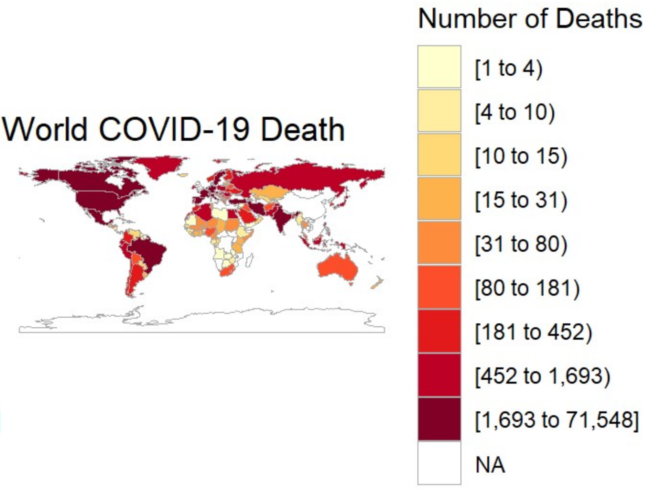
Nevertheless, we can see from the side-by-side word clouds for Trump-specifically across months that mentions of the impeachments increased, and the impeachment process seemingly unwavered.

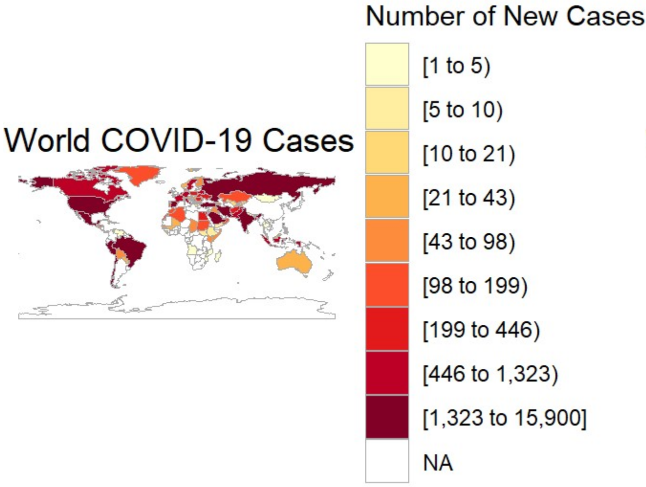
The Impact of the Coronavirus

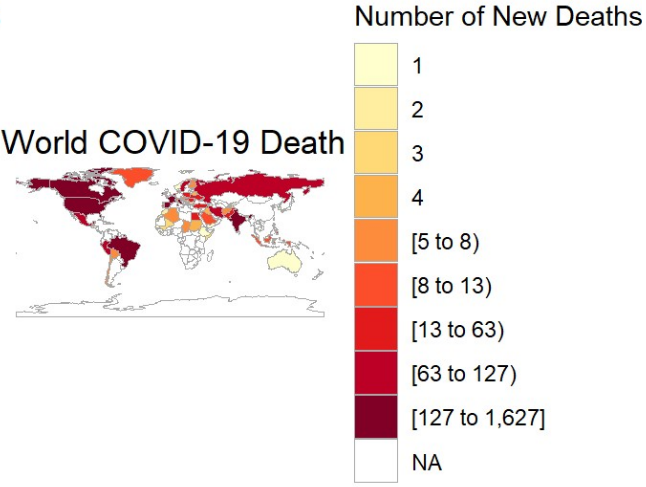
To quickly describe my process of web scraping. At the beginning, the html_node() does not parse the data, rather acts a CSS selector and there is no set range for the page being read in. This data is read into a table. Accordingly, the line that follows provides the relevant lines of the page to be read in [9:221] - and they are read into a variable. From there the data is cleaned/preprocessed so that it can be inserted into a choropleth graph with value and region being set as required.

```
Corona <- "https://www.worldometers.info/coronavirus/#countries"%>%
  read_html() %>% html_nodes("table")%>%
  html_table()%>%.[[1]]

#Taking only the relevant lines of the scraped webpage. Line 9 to 221.

country_cases <- Corona[9:221,]



#Add column to Corona dataset for state_choropleth requirement of region and value col
umns

#Meet the need for lower case and a column called 'region' for choropleth
Corona$region <- tolower(Corona$`Country,Other`)


Corona$continent <- tolower(Corona$Continent)



#Meet the need for correct formating without ',' in values for choropleth after reciev
ing error.This is to view table if interested in the origin of the values that appear
in the next segment.

Corona$value <- Corona$TotalCases
Corona$value <- as.numeric(gsub(",", "", Corona$value))


Corona$newcases <- Corona$NewCases
Corona$newcases <- as.numeric(gsub(",", "", Corona$newcases))


Corona$totaldeaths <- Corona$TotalDeaths
Corona$totaldeaths <- as.numeric(gsub(",", "", Corona$totaldeaths))

Corona$totalrecovered <- Corona$TotalRecovered
Corona$totalrecovered <- as.numeric(gsub(",", "", Corona$totalrecovered))

Corona$newdeaths <- Corona$newdeaths
Corona$totaltests <- as.numeric(gsub(",", "", Corona$TotalTests))
```


World COVID-19 Cases
Number of Cases
[6 to 41)
[41 to 151)
[151 to 480)
[480 to 896)
[896 to 1,946)
[1,946 to 4,838)
[4,838 to 10,929)
[10,929 to 30,251)
[30,251 to 1,228,735]
NA


World COVID-19 Death
Number of Deaths
[1 to 4)
[4 to 10)
[10 to 15)
[15 to 31)
[31 to 80)
[80 to 181)
[181 to 452)
[452 to 1,693)
[1,693 to 71,548]
NA





We can see from the visualizations that the number of cases is greatest in the Americas and Eastern Europe. We can also see that the number of deaths is greatest in the Americas. The first table distinctly shows us that the US is followed by Spain and Italy in these numbers, while the total number of cases is proven to be largest in the USA. Secondly, the number of new cases is greatest in the USA at 21,581 and this graphically shown as well, side-by-side with the number of new deaths, for which the USA, again, has the highest ranked position.

Conclusion

We can see that the coverage of the coronavirus in January was overwhelming in terms of NYT articles. Interestingly, though the wordclouds reflect a spillover of impeachment coverage into coverage about the coronavirus. The impeachment appeared to be consistently present in the news from January through to March. Also, given the visualizations provided, it is clear why the coronavirus garnered so much coverage.

Challenges

The biggest challenge that I faced during this project was in extracting data from the New York Times (NYT) Articles and placing them into a dataframe.This heavily relied on the API connection. I was required to put the article data into a dataframe to ensure that my data required could be fed into the code for visualizations. A great many of my initial attempts at running the NYT function, written in the first chunk, lead to the connection to the NYT timing out. Within this function, writing the for-loop to collect as many as articles as possible was another challenge as it had to be balanced with the API requesting too much data and the connection subsequently breaking. An equally important challenge was learning how to navigate the NYT code semantics through NYT development pages to ensure I could extract information that I was looking for.

Another challenge I faced was deciding upon the visualization to use to make my initial argument about the overlap in topics covered by the NYT in January. Ensuring that I could somewhat visually demonstrate this overlap was important so I created bigram tables (without stop-words) and word clouds to solidify this point; the observation that there was a visible overlap in the discourse in January, when the focus of a nation changed. Owing to time constraints, I also did not manage to involve a depiction of national sentiment, however, I think this would have been an interesting visual if there was a change insntiment around Trump following his handling of the coronavirus in the earlier days of the pandemic.

A lesser issue I faced was in formatting the document so that only the information that I wanted to show in the resulting pdf was being shown, instead of all intermediate steps being shown as well.

Lastly, after writing to csv's to circumvent the aforementioned issue with the functions, I uploaded those documents to my github repository. A subsequent challenge I had was running the data from my github repository in R as there were unicode blocks in the csv's that I had uploaded to github. Accordingly, I had to adjust my code in R to take this into account. I assigned these links to the original dataframe variables so that I would not have to change any subsequent code.