Visualising data from Joe Biden's speeches

Creating the joint dataset

The joint dataset was obtained by using the following code:

```
fileLocation <- list.files(path=".", pattern=".txt", full.names = TRUE)
filename <- lapply(basename(fileLocation), file_path_sans_ext)

location <- lapply(filename, gsub, pattern = regex("_(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)
location <- gsub(x = location, pattern = regex(pattern = "_"), replacement = " ")
event <- lapply(filename, gsub, pattern = regex(".*(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)?|
event <- gsub(x = event, pattern = regex(pattern = "_"), replacement = " ")
date <- lapply(str_match(string = filename, regex("(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)?|

list <- lapply(fileLocation, read_delim, delim = " ")
list <- mapply(cbind, list, "location" = location, SIMPLIFY = FALSE)
list <- mapply(cbind, list, "event" = event, SIMPLIFY = FALSE)
list <- mapply(cbind, list, "date" = date, SIMPLIFY = FALSE)

dfList <- as_tibble(do.call(rbind.data.frame, list))
names(dfList)[1] <- "speech"
```

The dataset which joins together all 6 of Joe Biden's speeches contains 5 columns. Starting with the "speech" column, which contains the whole transcript of a speech given by Joe Biden. These transcripts are divided into parts for each speech by the "part" column. Then there are the "location" column which notes where the speech took place, the "event" column which notes how was the event called when the speech took place, and finally the "date" column which notes when did the speech took place.

The basic information for each speech is given in this table:

Date	**Location**	**Name of Event**
20 / 08 / 2020	Milwaukee	Democratic National Convention
20 / 09 / 2020	Philadelphia	SCOTUS
23 / 09 / 2020	Charlotte	Racial Equity Discussion
26 / 09 / 2020	Washington	US Conference of Mayors
29 / 09 / 2020	Cleveland	Whistle Stop Tour
25 / 11 / 2020	Wilmington	Thanksgiving

The R command *cbind* joins the 6 separate speeches by their respective columns, but first a function was applied to them to identify each column from the data. These two functions are:

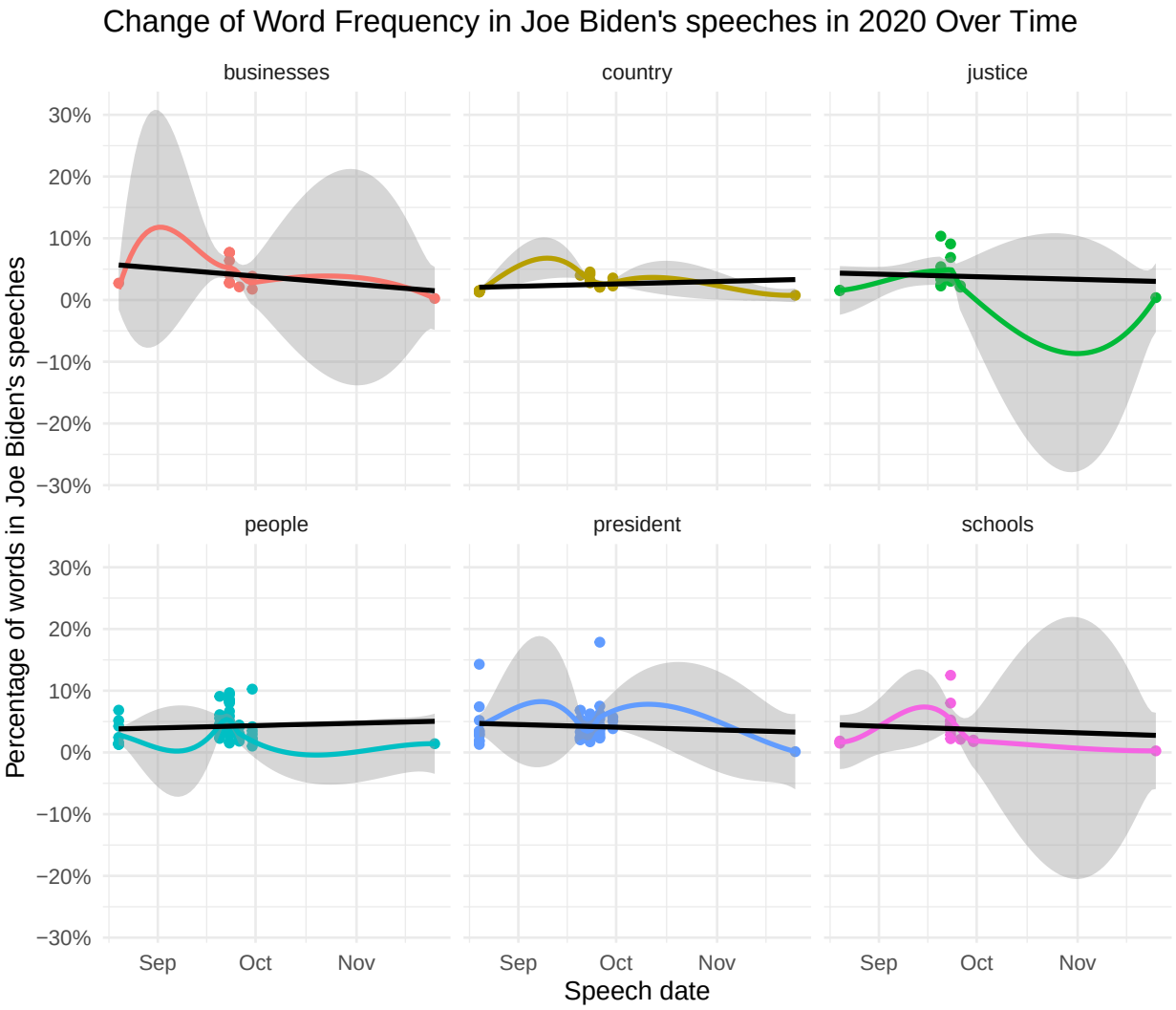
- *lapply*
- *mapply*

With these two, R is able to firstly locate the speech files saved in the sub folder. Then it extracts the relevant information before it is then added into the joint dataset named **dfList**

Frequency of words over time

The following graphs show the change in the frequency of certain words over time. In particular the words "schools", "justice", "president", "businesses", "country" and "people", generated by the following code:

```
frequency(c("schools", "justice", "president", "businesses", "country", "people"))
```



From the graphs, it can be noted that the chosen words in question don't change in frequency that often during Biden's speeches. This can suggest that Joe Biden gave each topic the same amount of coverage in each of his speeches.

There is a slight negative trend in the percentage of frequency for four of the six words. This could be helped by the final speech, which was conducted on Thanksgiving, where Joe Biden's speech would've been less about him and his election campaign and more on the values of the holiday itself. This is why words like "country" and "people" do not have a negatively sloped trendline as these are words that are more likely to be used in a speech about Thanksgiving.

"Businesses" is the word with the steepest trendline of the six chosen words. A probable cause for this is not a lack of use of the word in Biden's latter speeches, but rather the high volume of its usage in his earlier speeches. Around late September, the USA was only a few months outside their first COVID-19 lockdown, so many businesses had been struggling to keep up with costs when income had massively reduced or even completely gone over the previous 3 to 4 months.

It was therefore, of greater importance for Biden to include his ideas for businesses following the pandemic as it was one of the greater problems facing the country at the time.

Overall, there isn't much variation in the chosen words over time, other than the majority slowly decreasing in frequency over time.

Top tf-idf words

The term tf-idf **(term frequency-inverse document frequency)** measures the relevancy of a particular word within a document relative to a collection of documents. It is calculated using the following equation:

$$tf \cdot idf$$

where:

$$tf = \frac{Number\ of\ times\ the\ term\ appears\ in\ the\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

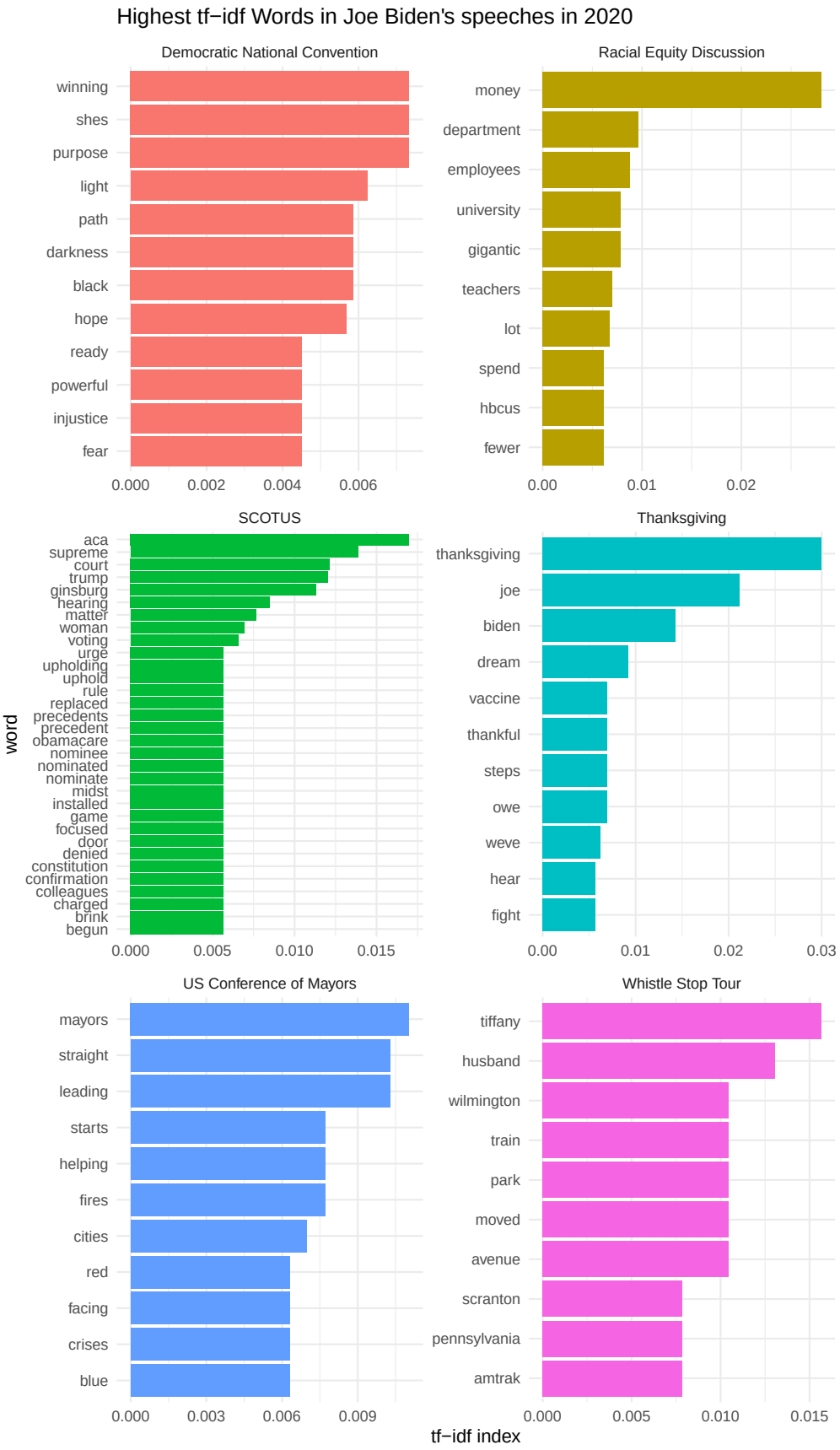
and

$$idf = \frac{Number\ of\ documents\ in\ the\ selection}{Number\ of\ documents\ in\ the\ selection\ that\ contain\ the\ term}$$

The highest 10 tf-idf words for each of Joe Biden's six speeches have been calculated and presented below:

```
df <- speechesData
names(df)[1] <- "word"
df[[1]] <- tolower(df[[1]])
df[[1]] <- str_replace_all(df[[1]], "[^[:alnum:] ]", "")
df <- separate_rows(df,1,sep = " ")
df <- df[!(is.na(df$word) | df$word==""), ]

df %>%
  anti_join(stop_words, by = c("word" = "word")) %>%
  count(event, word) %>%
  bind_tf_idf(word, event, n) %>%
  arrange(desc(tf_idf)) %>%
  group_by(event) %>%
  top_n(10) %>%
  ggplot(aes(x = reorder(word, tf_idf), y = tf_idf, fill = event)) +
  theme_minimal() +
  geom_col() +
  coord_flip() +
  facet_wrap(~ event, scales = "free", ncol = 2) +
  labs(y = "tf-idf index", x = "word", title = "Highest tf-idf Words in Joe Biden's speeches in 2020") +
  theme(legend.position = "none")
```


Highest tf−idf Words in Joe Biden's speeches in 2020
Democratic National Convention
winning
shes
purpose
light
path
darkness
black
hope
ready
powerful
injustice
fear
0.000
0.002
0.004
0.006
Racial Equity Discussion
money
department
employees
university
gigantic
teachers
lot
spend
hbcus
fewer
0.00
0.01
0.02
SCOTUS
aca
supreme
court
trump
ginsburg
hearing
matter
woman
voting
urge
upholding
uphold
rule
replaced
precedents
precedent
obamacare
nominee
nominated
nominate
midst
installed
game
focused
door
denied
constitution
confirmation
colleagues
charged
brink
begun
0.000
0.005
0.010
0.015
Thanksgiving
thanksgiving
joe
biden
dream
vaccine
thankful
steps
owe
weve
hear
fight
0.00
0.01
0.02
0.03
US Conference of Mayors
mayors
straight
leading
starts
helping
fires
cities
red
facing
crises
blue
0.000
0.003
0.006
0.009
Whistle Stop Tour
tiffany
husband
wilmington
train
park
moved
avenue
scranton
pennsylvania
amtrak
0.000
0.005
0.010
0.015
word
tf−idf index

Here, we are able to see the wide variety of words used for each of Joe Biden's speeches. Each graph shows the top 10 most frequently used words, yet in some there are more words included, most noticeably in the speech named 'SCOTUS'. This is due to some words being the joint 10th most frequent, where R has been coded to include them.

What we can conclude from the graph is that *thanksgiving* is the most relevant word used by Joe Biden, as it has the highest score out of the total number of words used (0.03). It must be noted that it does not come as a surprise, as the speech was made on Thanksgiving and it is not a word that would likely come up in an election campaign speech throughout August and September.

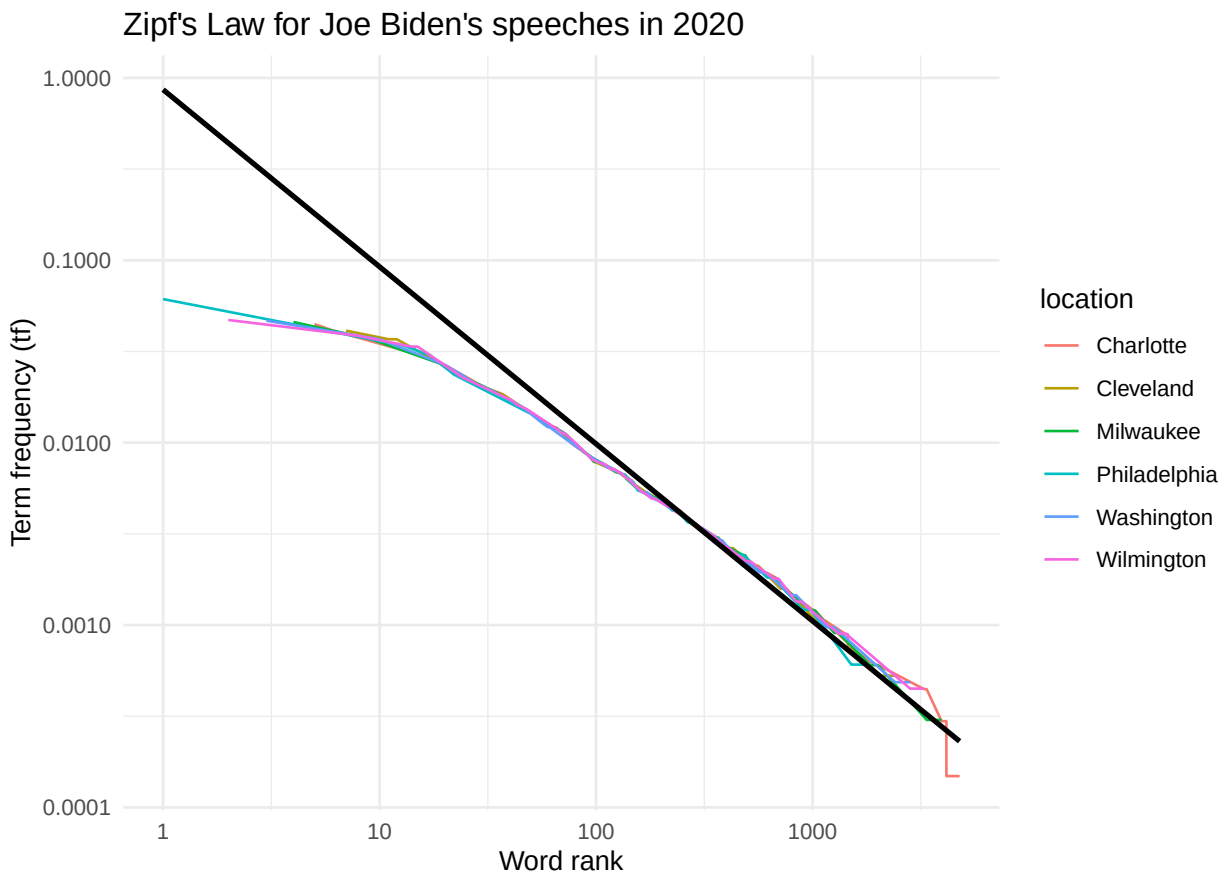
There are two clear trends that can be seen from these graphs that indicate Joe Biden's need to focus on one topic in particular. As discussed earlier, *thanksgiving* is the dominant term for that particular speech. Plus this was after Joe Biden had won the election, so there wasn't a great need for him to focus on relevant topics to increase his votes.

The 'Racial Equity Discussion' also contains just one word (money) that is deemed more relevant than others made in that speech, whereas most of the others have a number of words that take up similar amounts of relevancy as the top word. For these speeches, it can be deduced that Joe Biden did not have a particularly unique topic to talk about, but instead gave a more even distribution of time to a few extra topics.

Zipf's Law

Zipf's law states that the frequency of a word is inversely proportional to its rank. This leads to the most frequently used word being said twice as often as the second, three times as the third and so on. The results of this for Joe Biden's speeches can be seen below:

```
df %>%
  count(location, word) %>%
  bind_tf_idf(word, location, n) %>%
  arrange(desc(tf)) %>%
  mutate(rank = row_number()) %>%
  group_by(location) %>%
  ggplot(aes(x = rank, y = tf, colour = location)) +
  geom_line(linewidth = 1.4) +
  geom_smooth(method = "lm", se = FALSE, colour = "black", linewidth = 0.5) +
  theme_minimal() +
  scale_x_log10() +
  scale_y_log10(labels = label_comma()) +
  labs(y = "Term frequency (tf)", x = "Word rank", title = "Zipf's Law for Joe Biden's speeches in 2020
```



What can be seen clearly is that there is indeed a negative relationship between the term frequency the the word rank, so Zipf's law holds for all six speeches. Zipf's law is not as accurate for the words that are towards the top of the word ranking, as shown by all six speeches flowing below the regression line, but from the words ranked from 100 onward, Zipf's law holds almost exactly.