MATH513 Big Data and Social Network Visualization Coursework

Luciana Dalla Valle

Academic Year 2020-21

1 Coursework Information

Please read the following points before attempting the coursework:

- The deadline for this assignment is **10 am** on **Thursday, January 14th, 2021**. You should submit your work through the MATH513 Big Data and Social Network Visualization DLE site. Your submission will be marked anonymously.
- **This is a group coursework. Please work in self-assigned groups of up to three people. The groups must be the same as those of the Practical task.**
- You should keep notes of all your meetings. Each member of the group will receive the same mark, unless any member chooses to make use of the Peer Assessment option. If you wish to make use of the Peer Assessment option, you will need to contact the Module Leader **Dr Luciana Dalla Valle** by **Wednesday, 13th January, 2021** to make an appointment.
- This assignment counts for 60% of your final mark on this module. Marks will be assigned according to the marking grid on page 16.
- Marked scripts will be returned within **20 working days** of the submission date. In particular, you will get full feedback on your work by February 11th, 2021.

- You are reminded of the **University's Academic Regulations**:

  Academic offences occur when activity is undertaken which could confer an unfair advantage to any candidate(s) in assessment. The University recognises the following (including any attempt to carry out the actions described) as academic offences, regardless of intent:

  a. Plagiarism, which is copying or paraphrasing of other people's work or ideas into a submitted assessment without full acknowledgement. More information on plagiarism is available here:
  https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/plagiarism

  b. Collusion, which is unauthorised collaboration of students (or others) in producing a submitted assessment. The offence of collusion occurs if a student copies any part of another student's work, or allows their own work to be copied. Collusion also occurs if other people contribute significantly to work that a student submits as their own.

  The complete list of regulations can be found here:
  https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations

  By submitting this coursework, all group members confirm that they have understood the University's policy on plagiarism and collusion.

We now state the relevant MATH513 Big Data and Social Network Visualization Assessed Learning Outcomes (ALOs) for this assignment.

At the end of the module the learner will be expected to be able to:

ALO1 Critically select and use a broad range of techniques to perform Big Data manipulation and visualization;

ALO2 Perform exploratory analyses to extract information, insight and innovation from data;

ALO3 Collaborate with others to produce and document **R** code and to present its professional use for Big Data or Social Network Visualization.

You should keep these ALOs in mind when doing this coursework.

2 Donald Trump’s Rallies Datasets

This coursework comprises the following related parts, based on the Donald Trump’s Rallies datasets.

The txt files uploaded on the DLE contain full speeches that Donald Trump gave at 10 of his rallies in September 2020.

The title of each file indicates the rally location and date.

You will compute and present some numerical and graphical summaries of these data sets.

2.1 Data Preparation

To begin, you need to create a single dataset based on the ten txt files provided.

The newly created dataset should include two variables indicating the location and the date of the rallies.

You might find the functions read_delim from the package readr, and as_date from the package lubridate useful.

```
## # A tibble: 10 x 3
##    speech                                                 location    date
##    <chr>                                                  <chr>       <date>
##  1 "So thank you Pennsylvania, very much. I'm thrilled t~ Latrobe     2020-09-03
##  2 "Well, thank you very much. Thank you. Thank you very~ Winston-Sa~ 2020-09-08
##  3 "We brought you a lot of car plants, Michigan. We bro~ Freeland    2020-09-10
##  4 "Well, I thank you very much. So I want to start by s~ Minden      2020-09-12
##  5 "Thank you, thank you. Wow. Wow, and I'm thrilled to ~ Henderson   2020-09-13
##  6 "Thank you, thank you very much. Thank you very much.~ Mosinee     2020-09-17
##  7 "There's a lot of people. That's great. Thank you ver~ Bemidji     2020-09-18
##  8 "What a crowd, what a crowd. Get those people over he~ Fayettevil~ 2020-09-19
##  9 "Wow, that's a big crowd. This is a big crowd. Thank ~ Ohio        2020-09-21
## 10 "Doesn't have the power. Doesn't have the staying pow~ Pittsburgh  2020-09-22
```

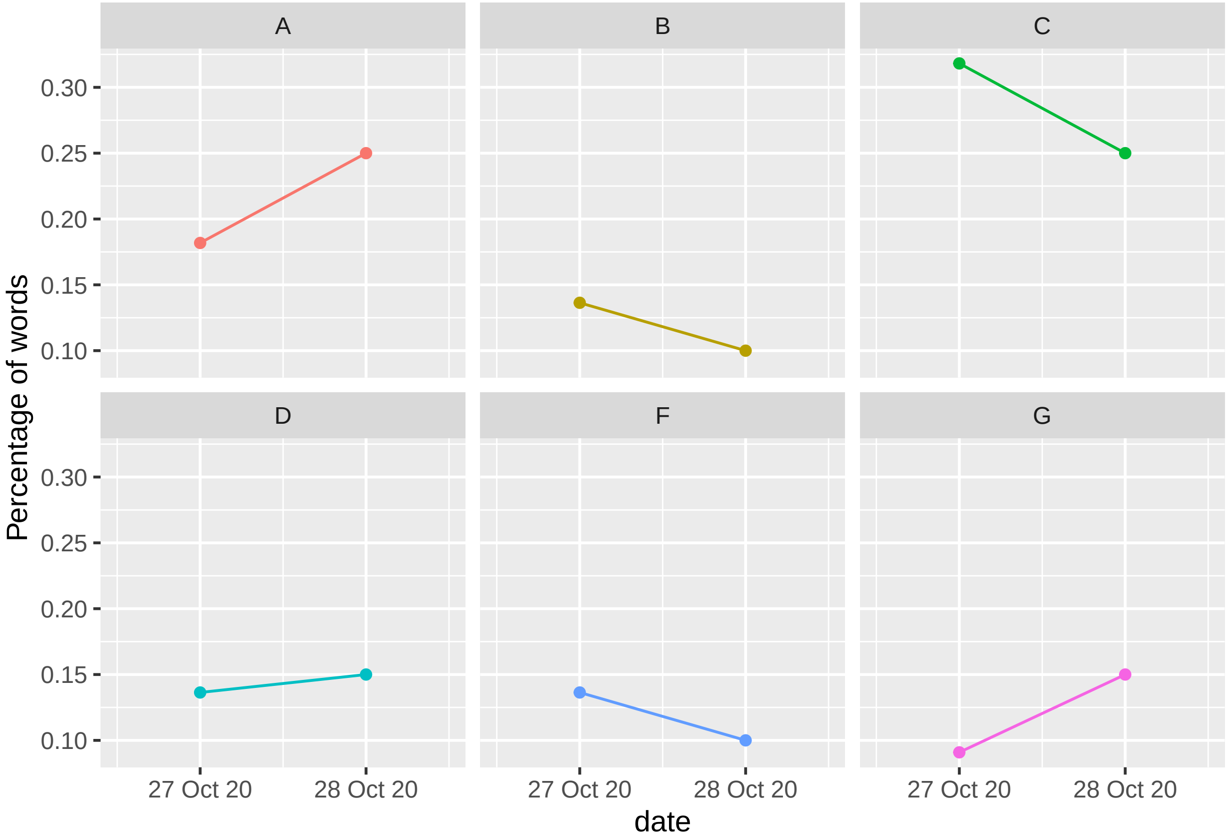
2.2 Writing an R Function Showing the Change of Word Frequency Over Time

Here is some **R** code that summarizes and visualizes an invented data set in the file `invented_data_for_illustration.txt`. You should study these data, code and output carefully.

```r
#
# Read in the data using read_delim from the readr package
#
b_d <- read_delim("invented_data_for_illustration.txt", delim = "\t")
#
# Work out percentages for each date using the dplyr package
#
b_p <- b_d %>%
    count(date, word) %>%
    group_by(date) %>%
    mutate(p = n / sum(n))
#
b_p
```

```
## # A tibble: 12 x 4
## # Groups:   date [2]
##    date       word      n      p
##    <date>     <chr> <int>  <dbl>
##  1 2020-10-27 A          4 0.182
##  2 2020-10-27 B          3 0.136
##  3 2020-10-27 C          7 0.318
##  4 2020-10-27 D          3 0.136
##  5 2020-10-27 F          3 0.136
##  6 2020-10-27 G          2 0.0909
##  7 2020-10-28 A          5 0.25
##  8 2020-10-28 B          2 0.1
##  9 2020-10-28 C          5 0.25
## 10 2020-10-28 D          3 0.15
## 11 2020-10-28 F          2 0.1
## 12 2020-10-28 G          3 0.15
```

```r
#
# Plot the contents of b_p using ggplot2
#
ggplot(b_p,
       aes(x = date,  y = p,  colour = word)) +
  geom_point() +
  geom_line() +
  labs(y = "Percentage of words") +
  facet_wrap(~ word) +
  scale_x_date(labels = date_format("%d %b %y"),
               limits = c(as_date("2020-10-26")+0.5,
                          as_date("2020-10-29")-0.5)) +
  theme(legend.position = "none")
```



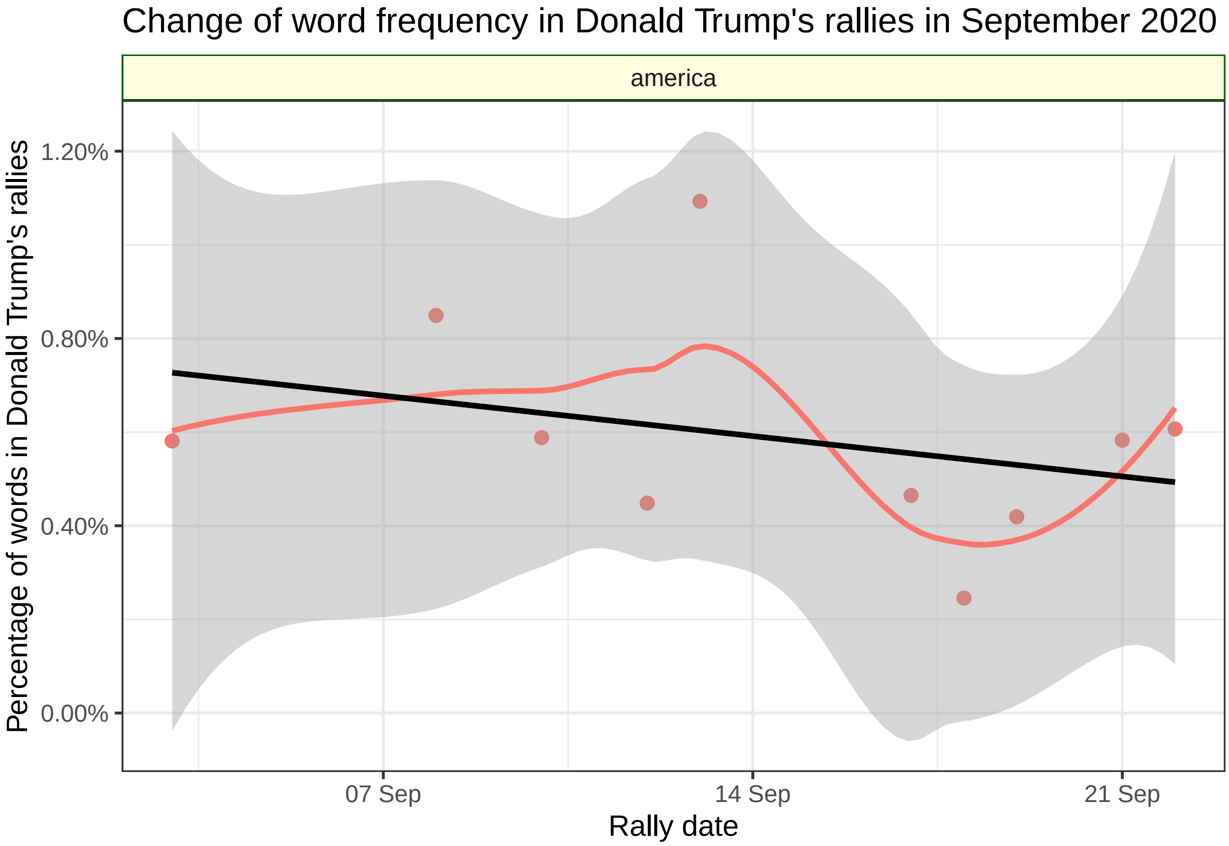
You should write an R function that produces a plot showing how the proportion of given words in Donald Trump's rallies changes over time, as shown below. The function should take in as an argument the required word. In fact, it is not hard to make your function deal with several words; you may need `%in%` to do this.

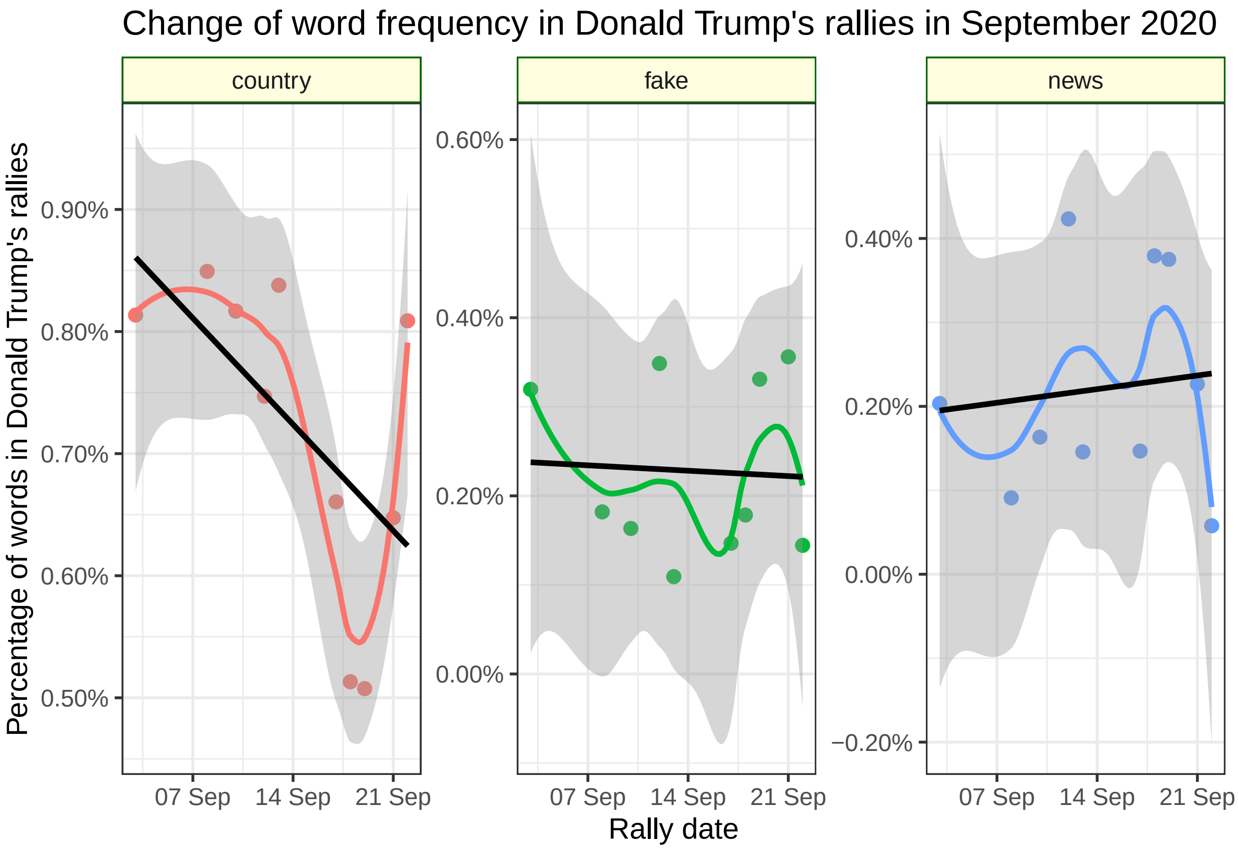
First, your function needs to tokenize the dialogue, splitting each sentence in separate words, and remove stopwords. You may find the functions of the `tidytext` package that we met in the Twitter practical session useful for this.

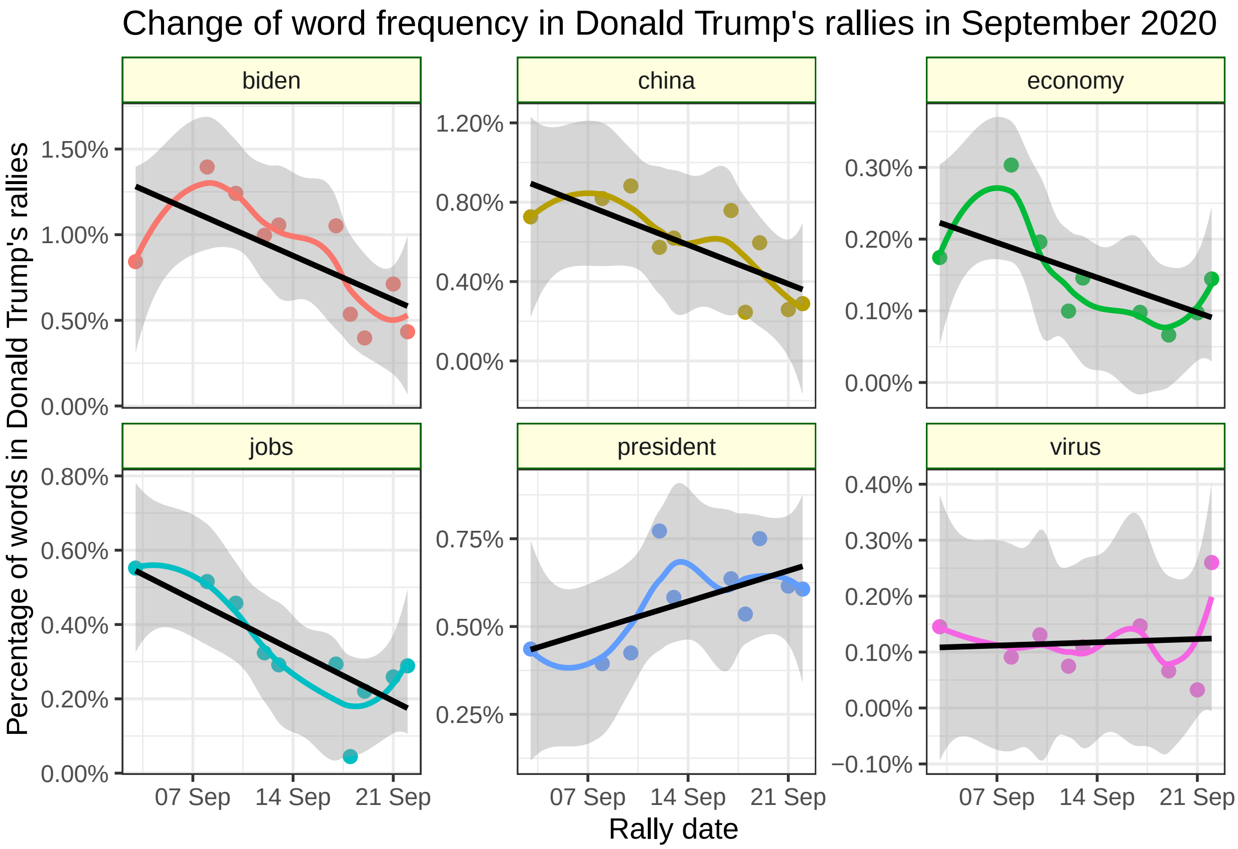
Then, you should use code similar to the above to work out the proportion of words within each rally, and then filter the results by word.

You are encouraged to improve on the plots presented here.

Discuss Donald Trump's rallies data and describe the graph that shows the change in the frequency of given words over time. What may be concluded from your graph? In particular, how do the use of specific words (such as, for example, "virus", "biden", "china", "president", "economy", "jobs") change over the ten different speeches given by Donald Trump in September 2020?




Change of word frequency in Donald Trump's rallies in September 2020
country
fake
news
Percentage of words in Donald Trump's rallies
0.90%
0.80%
0.70%
0.60%
0.50%
0.60%
0.40%
0.20%
0.00%
0.40%
0.20%
0.00%
−0.20%
07 Sep
14 Sep
21 Sep
Rally date


Change of word frequency in Donald Trump's rallies in September 2020
biden
china
economy
jobs
president
virus
Percentage of words in Donald Trump's rallies
Rally date
0.00%
0.50%
1.00%
1.50%
0.00%
0.40%
0.80%
1.20%
0.00%
0.10%
0.20%
0.30%
0.00%
0.20%
0.40%
0.60%
0.80%
0.25%
0.50%
0.75%
−0.10%
0.00%
0.10%
0.20%
0.30%
0.40%
07 Sep
14 Sep
21 Sep

2.3 Plotting the Words with Highest `tf-idf` Value

The **tf-idf** index computes the frequency of a term adjusted for how rarely it is used.

The **tf-idf** index is calculated as the product of the **term frequency** (**tf**) and the **inverse document frequency** (**idf**):

$$tf\text{-}idf = tf \times idf.$$

The **term frequency** (**tf**) identifies how frequently a word occurs in a document. However, many common words such as "the", "is", "for", etc. typically achieve the highest **tf** values.

The **inverse document frequency** (**idf**) decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. The **idf** is defined as

$$idf = \log\left(\frac{N}{n_t}\right)$$

where N is the total number of documents being assessed and n_t is the number of documents where the term t appears.

The **tf**, **idf** and **tf-idf** indexes can be easily calculated using the `bind_tf_idf` function form the `tidytext` package. The `bind_tf_idf` function is illustrated by the following code that calculates the **tf**, **idf** and **tf-idf** indexes from the `invented_data_for_illustration.txt` dataset.

```r
#
# Work out tf, idf and tf_idf indexes
#
b_t <- b_d %>%
    count(location, word) %>%
    bind_tf_idf(word, location, n) %>%
    arrange(desc(tf_idf))
b_t
```

```
## # A tibble: 10 x 6
##    location word      n     tf   idf tf_idf
##    <chr>    <chr> <int>  <dbl> <dbl>  <dbl>
##  1 Plymouth A         9 0.346  0.693  0.240
##  2 Plymouth B         5 0.192  0.693  0.133
##  3 Penzance C         7 0.438  0      0
##  4 Penzance D         3 0.188  0      0
##  5 Penzance F         2 0.125  0      0
##  6 Penzance G         4 0.25   0      0
##  7 Plymouth C         5 0.192  0      0
##  8 Plymouth D         3 0.115  0      0
##  9 Plymouth F         3 0.115  0      0
## 10 Plymouth G         1 0.0385 0      0
```

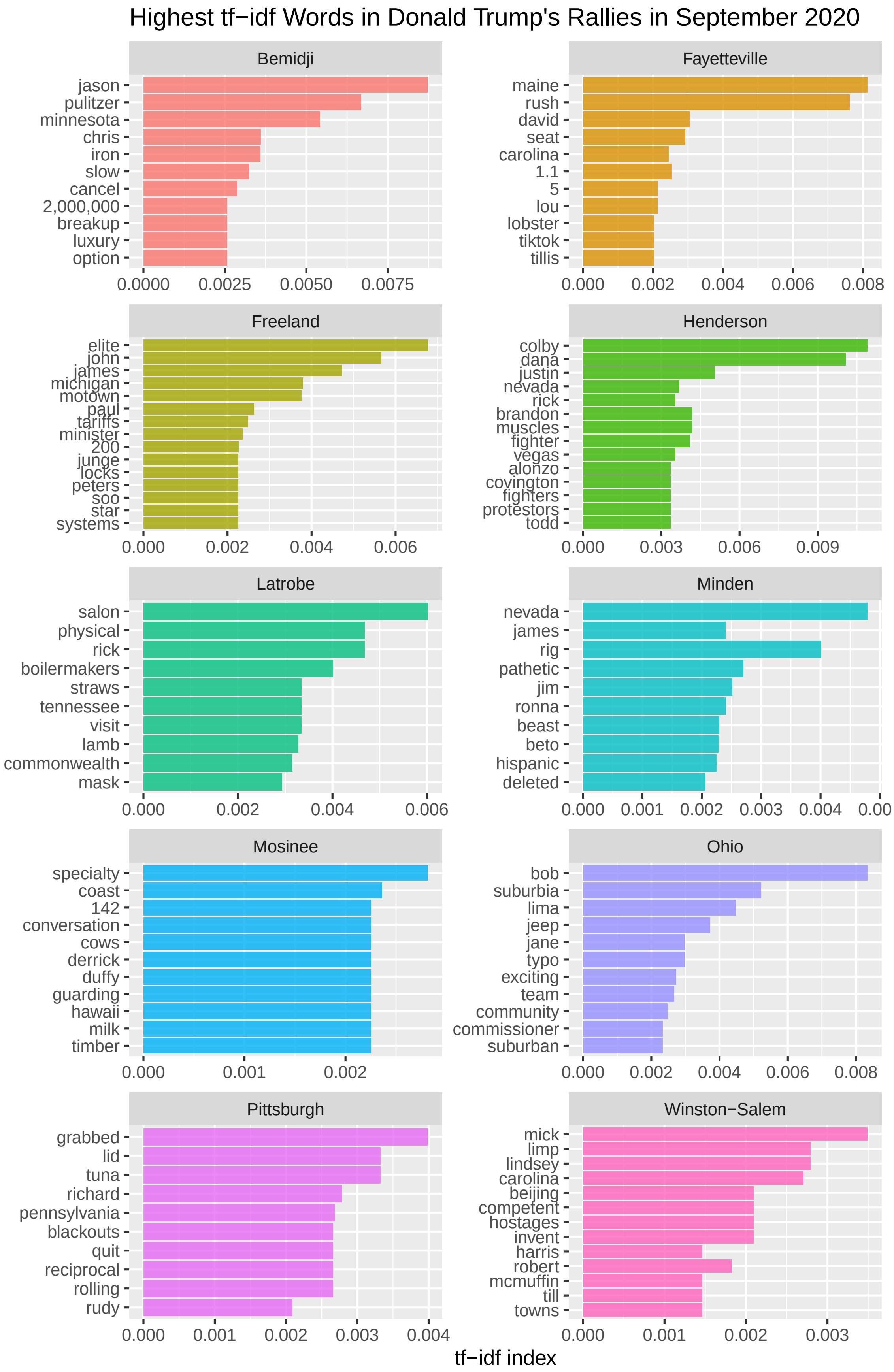
You should write **R** code that calculates the **tf**, **idf** and **tf-idf** indexes for the Donald Trump's rallies dataset, and extracts and plots the top 10 **tf-idf** words for each location.

Note that, in presence of ties, the number of extracted words could be greater than 10 words for each location.

Please, remember to tokenize the dialogue and remove stopwords first.

As always, you are encouraged to improve on the plot presented here.

Discuss what may be concluded from your graph.



Highest tf−idf Words in Donald Trump's Rallies in September 2020

2.4 Zipf’s Law

Zipf’s law is named after the American linguist George Kingsley Zipf (1902–1950) and originated in linguistic studies.

Zipf’s law states that within a group or corpus of documents, the frequency of any word is inversely proportional to its rank. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.

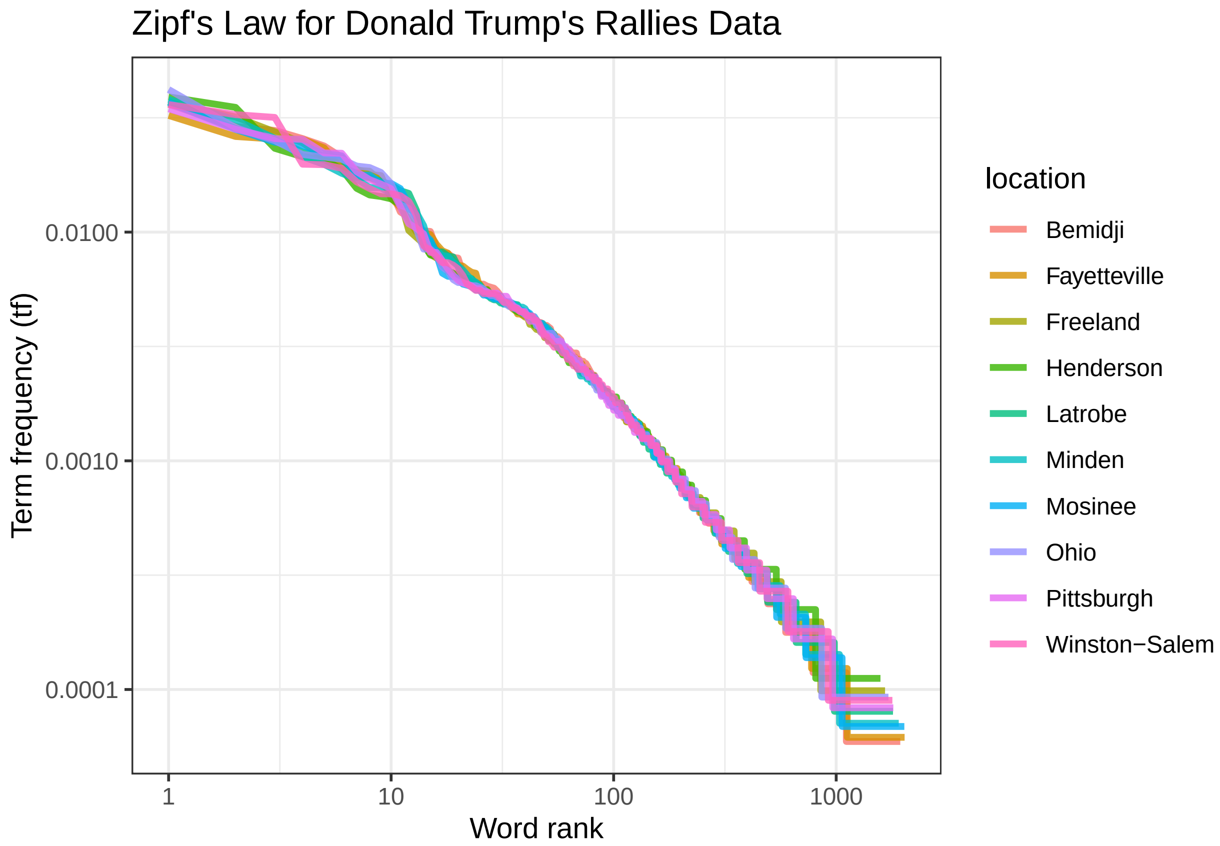
You should write **R** code that illustrates Zipf’s law by plotting Donald Trump’s rallies data on a log-log graph, with `log(term frequency)` on the vertical axis and `log(rank order)` on the horizontal axis.

First, you need to tokenize the dialogue, but, this time, you **should not** remove stopwords, since common words are important to illustrate Zipf’s law.

Then, the **term frequency** (or **tf**) can be easily calculated using the `bind_tf_idf` function, as explained above.

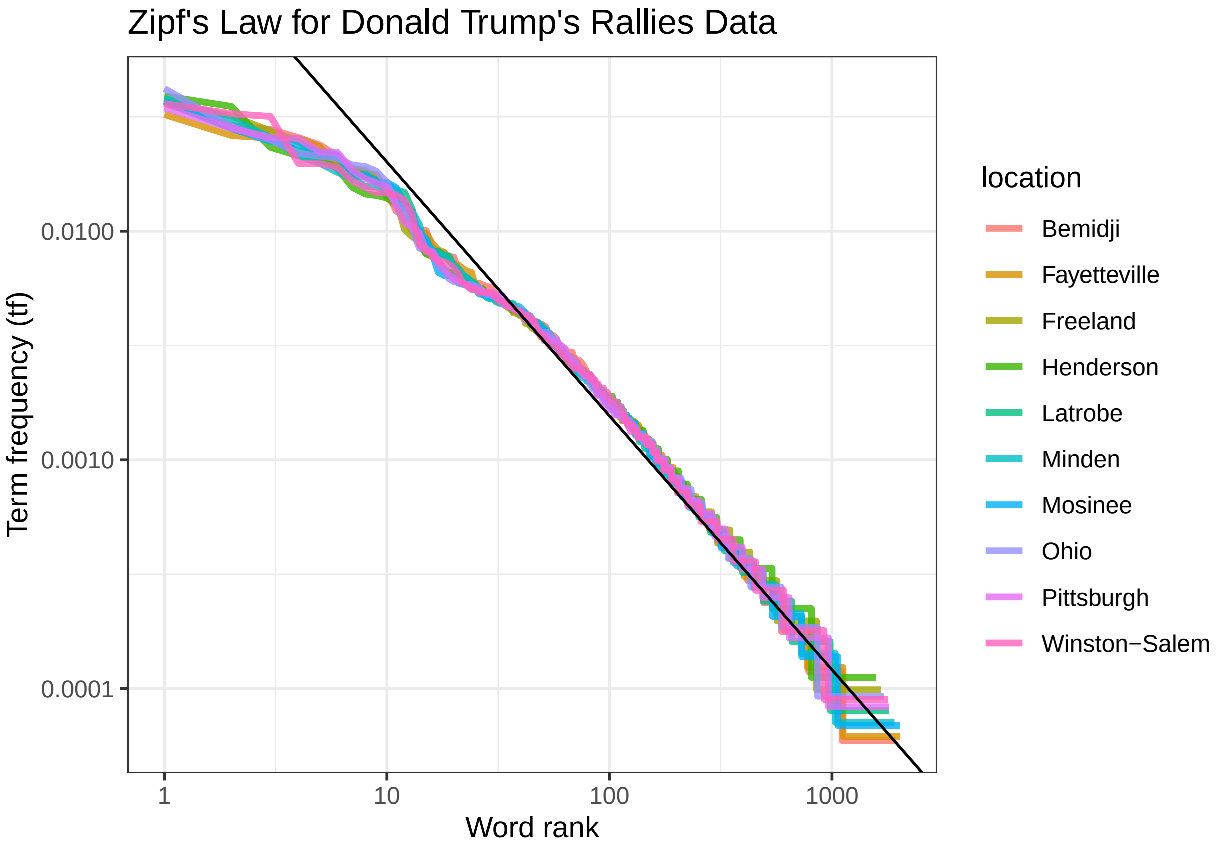
The **rank** can be obtained as the row number, after sorting the data in decreasing order of **tf** and grouping the data per location.

You should use a line plot with different colours for each location.



Now, apply the **linear regression model** to describe the effect of the `log(rank order)` to the `log(term frequency)` and provide an accurate interpretation of your results.

Add the regression line to your plot illustrating Zipf's law and comment on your results.



2.5 Discussing your Results in an R Markdown Report

You should produce an R Markdown report that:

- Discusses Donald Trump's September 2020 rallies data.
- Describes the graph that shows the change in the frequency of given words over time. You should also discuss what may be concluded from your graph. In particular, how do the use of specific words (such as, for example, "virus", "biden", "china", "president", "economy", "jobs") change over the ten different speeches given by Donald Trump in September 2020?
- Provides and illustrates **R** code that plots the words with the highest **tf-idf** index for each location. You should also discuss what may be concluded from your graph.
- Discusses the linear regression model and its application to describe Zipf's law in the context of Donald Trump's rallies data. You should provide an accurate interpretation of your regression results.
- Provides and illustrates **R** code that visualizes Zipf's law for each location of Donald Trump's rallies. You should also discuss what may be concluded from your graph.

You are required to submit your **R Markdown report** as a `pdf` file. **Your R Markdown report should not be more than 15 pages in length**, but may have an appendix that does not count towards the page limit, if you wish. Remember please that 15 pages is the limit and not the target.

2.6 Producing a Simple, Fully-Documented R Package (optional, to achieve high marks)

You should also produce a fully-documented **R** package that includes, as a minimum, the following:

- The Donald Trump's rallies joint dataset obtained as illustrated in the *Data Preparation* Section at page 3 with a suitable help page;
- An **R** function that produces a plot showing how the proportion of given words in Donald Trump's rallies changes over time with a suitable help page.

You are required to produce and submit the package **manual** as a `pdf` file.

If you submit the **R** package, the **R Markdown report**, as described above, **must be submitted as the package vignette**.

Please, note that to use a specific dataset in your vignette, the file containing the data needs to be placed in the vignettes subfolder of the package folder.

3 What You Need to Submit

One member of your group needs to submit the following **three files** electronically using the DLE.

- Your report/vignette in a file called
  `Vignette_First_Second_Third_Student_ID.pdf`
  where you substitute in the Student Identification Numbers of all the group members.
  For example, `Vignette_11034023_12045043.pdf` for a group of two people.
- The manual for your R package in a file called
  `Manual_First_Second_Third_Student_ID.pdf`.
  For example, `Manual_11034023_12045043.pdf` for the same group of two people.
- A binary version of your package built for Windows (or Mac) and called
  `Package_First_Second_Third_Student_ID.zip`.
  (or `Package_First_Second_Third_Student_ID.tgz` for Mac)
  For example, `Package_11034023_12045043.zip` for the same group of two people.
  It should be possible for the markers of this assessment to install, load and work with your package supplied in this way.

Please submit only **one** coursework per group.

If anything is unclear, you should ask **without delay**.

4 Marking Grid

MATH513 Big Data and Social Network Visualization: Coursework Marking Grid

Assessment Area	Maximum Mark
Discussion of Donald Trump's rallies data and data preparation	5
Correctness of **R** function showing the change of word frequency over time and graph discussion	15
Correctness of **R** code plotting the words with highest tf-idf and graph discussion	15
Illustration of the linear regression model and its application to describe Zipf's law, interpretation of results, correctness of **R** code and graph discussion	15
Correctness, quality and clarity of the report	10
Correctness and quality of the **R** package	20
Correctness and quality of the manual	10
Correct inclusion of the vignette in the **R** package	10
Total	100