TextMining HW

Kaiyu Yan, Fionnuala McPeake, Angela Zhai, Miller Xu

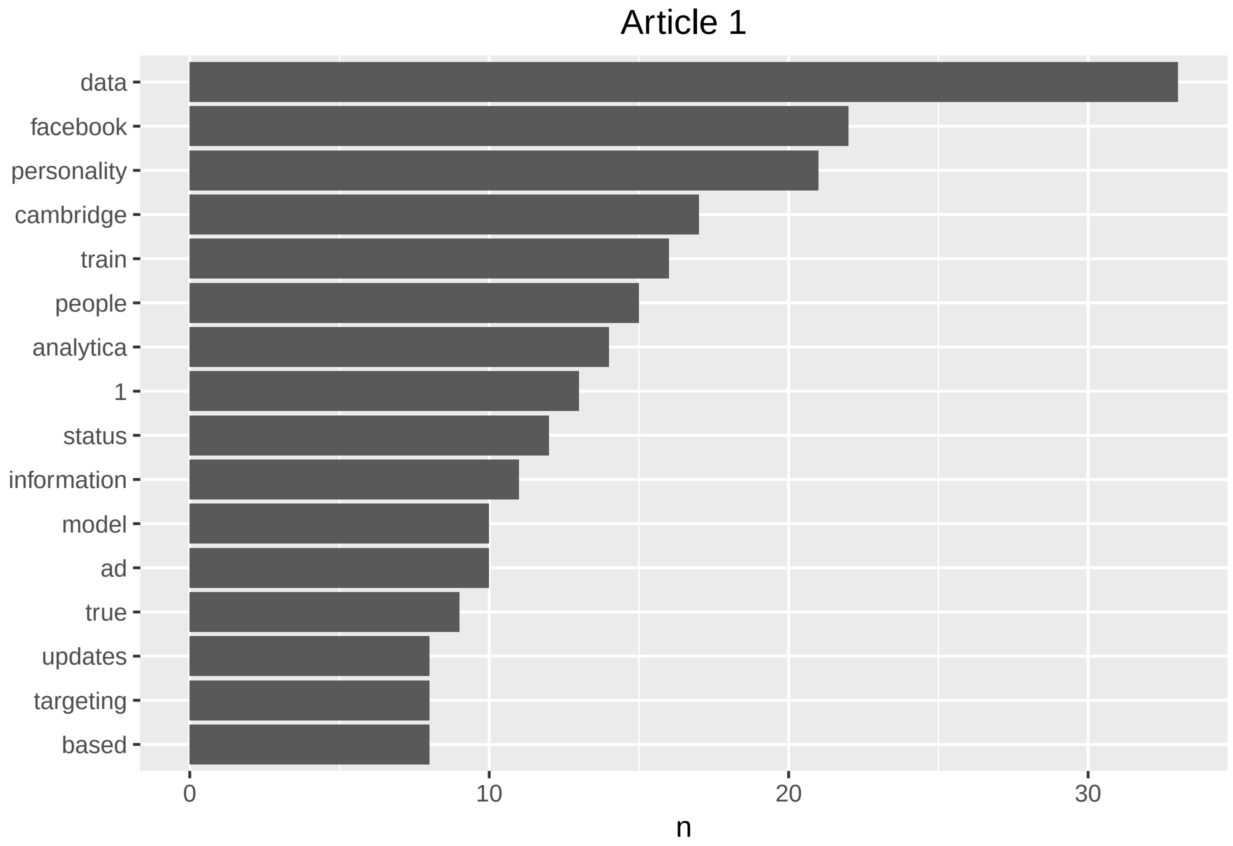
November 2, 2018

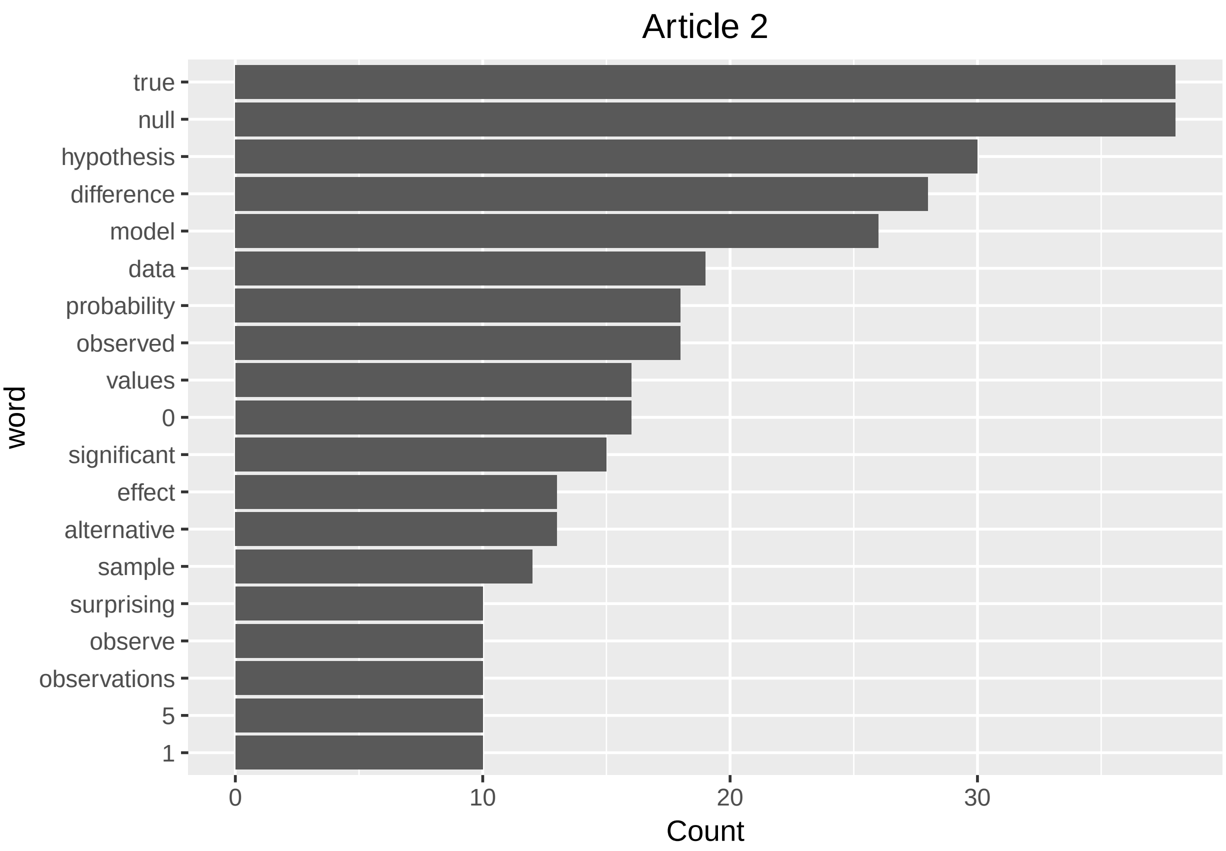
Introduction

In order to better understand the functions of tidytext, we have selected two blog posts to compare. The first article, "Reconstructing Cambridge Analytica's"Psychological Warfare Tool"" discusses how Cambridge Analytic used people's Facebook profiles in order microtarget of ads to the Trump presidential campaign, gives a simplified example of how such a thing can be done in R, and discusses the results and how they effective Cambridge Analytic targeted ads given what we have learned. The second article, "About p-values", discusses common misconseptions about p-values. These articles were selected becayse they both are of a substatnial length for a blog post, and take a stance on a topic.

Chapter 1

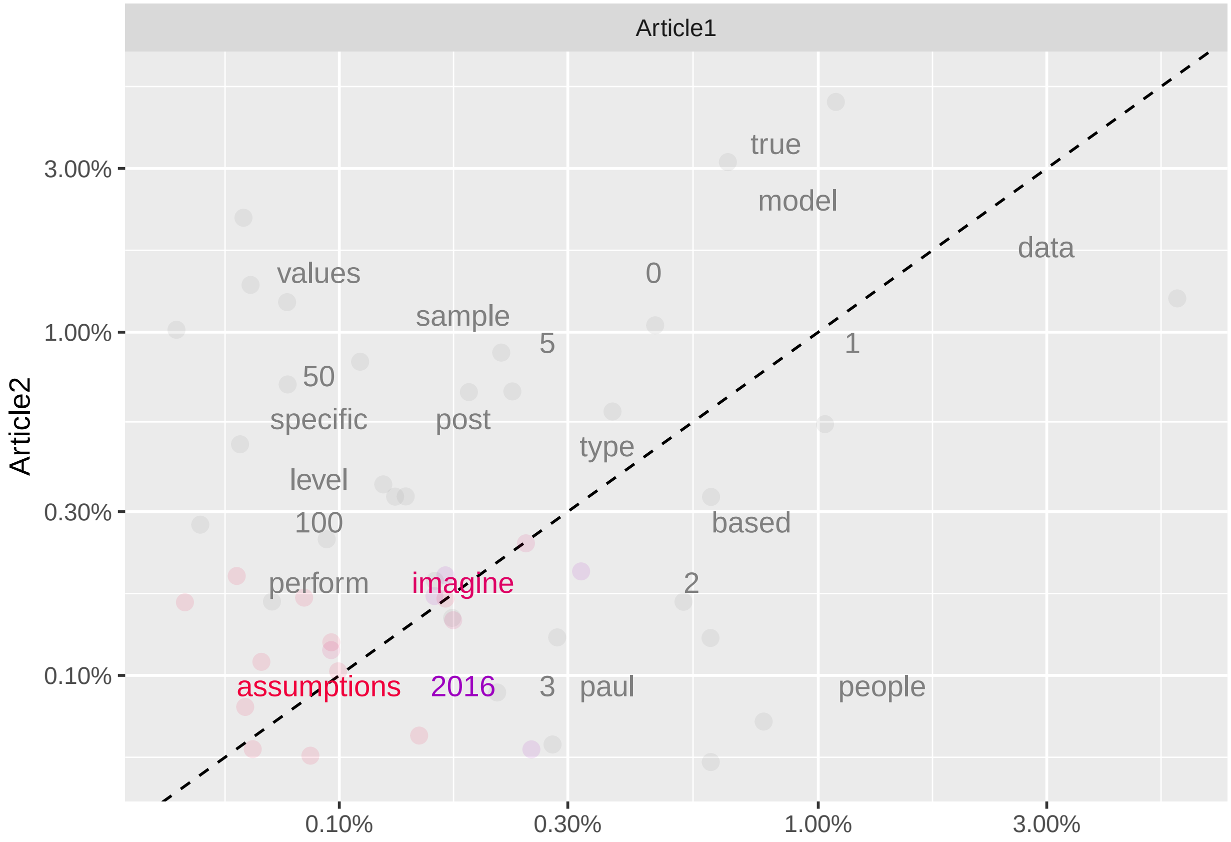
Plots were created to count the occurences of frequently used words in both articles. The biggest suprise for the second article is that "p-value" was not on the list, but because of the hyphen and the common use of "value" and "values", it's count was diluted. The graph showing the corrolation of common words between the two articles show that they are not particularly comparable to eachother, with an R value of 0.45. This is not suprising, as one is much more focused on statistical theory than the other.





```
## Warning: Removed 752 rows containing missing values (geom_point).

## Warning: Removed 752 rows containing missing values (geom_text).
```



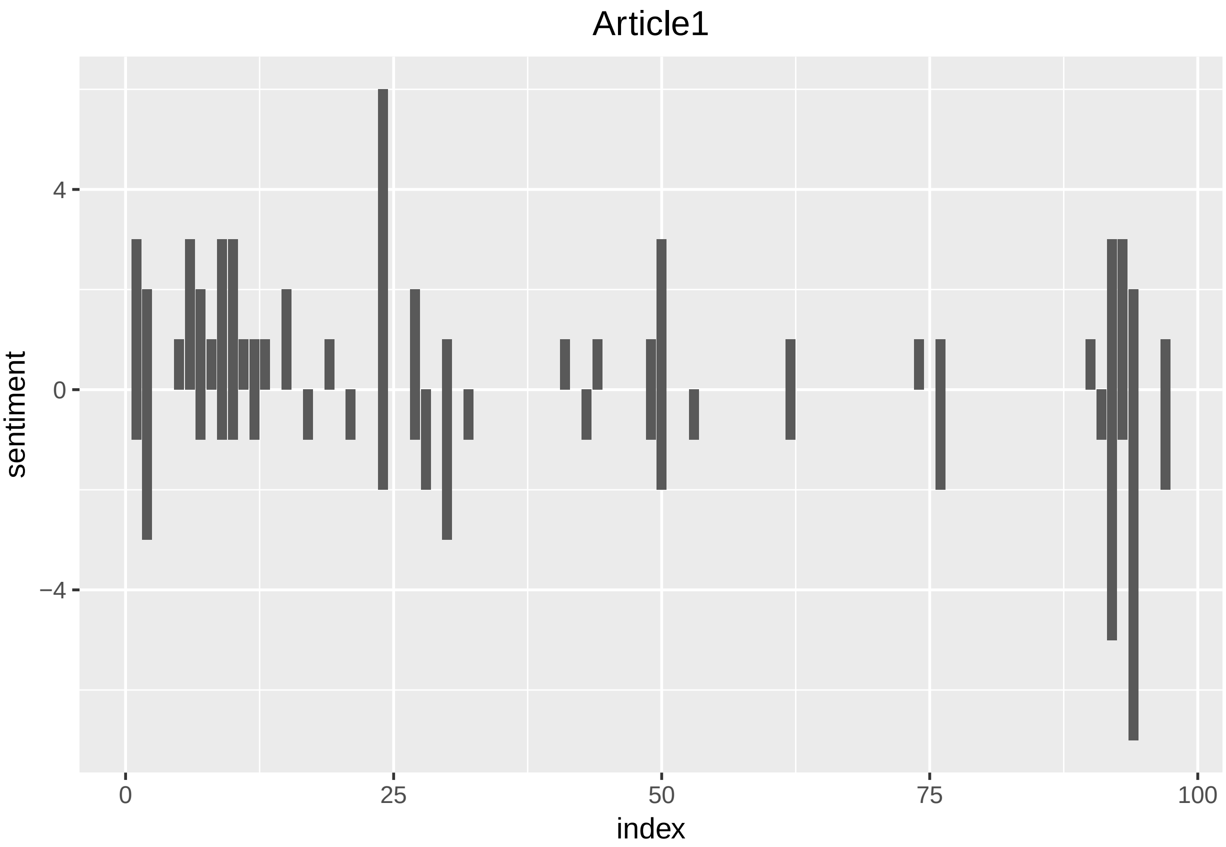
```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Article2
## t = 3.5106, df = 48, p-value = 0.0009835
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1986456 0.6487298
## sample estimates:
##      cor
## 0.452001
```

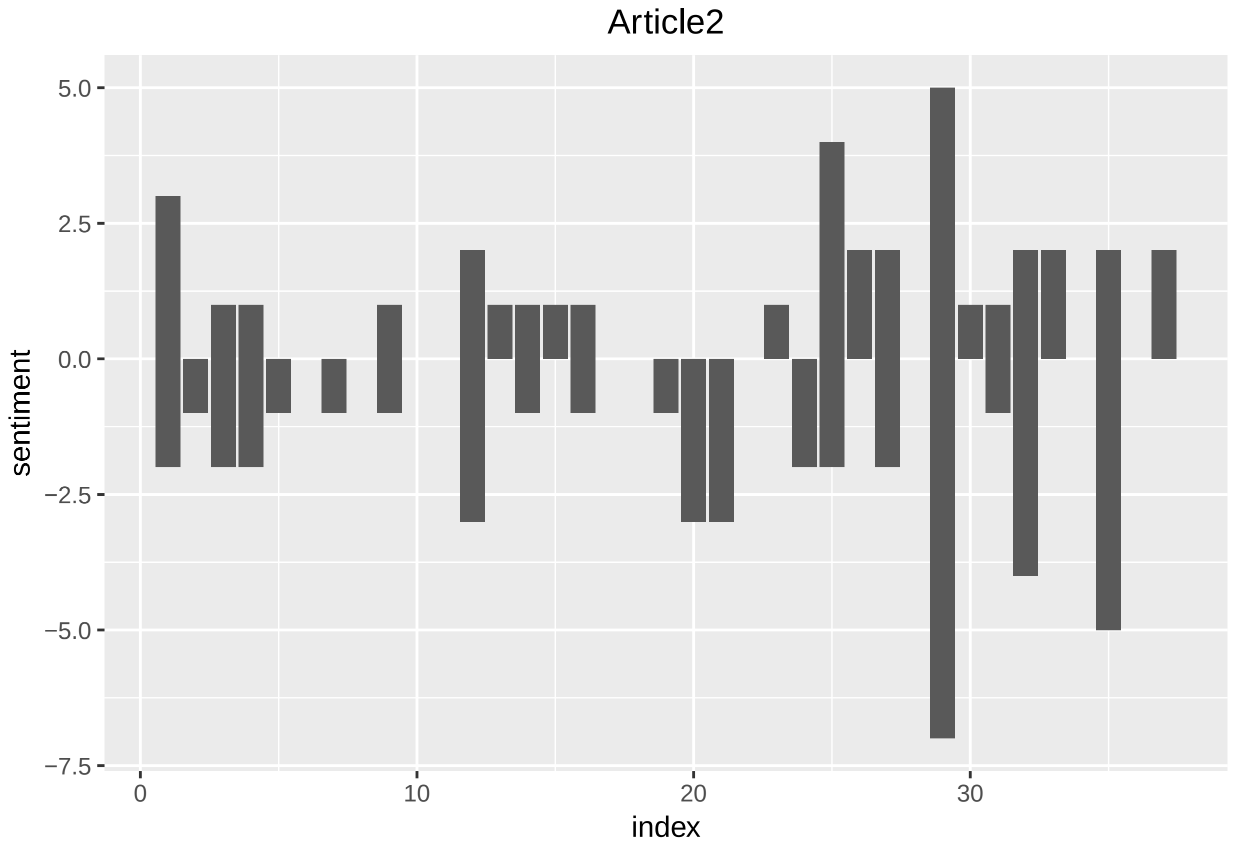
Chapter 2

A sentiment analyis was preformed on each article, indicating the progression in terms of positivity and negativity. Article 1 was longer than Article 2, and less negative, as indicated by the x- and y-axis, respectively. The second article seems to be more negative, not only because it's y-axis spans -5 to 5, as opposed to +/-4 for the first article, but in the frequency of negative bars. The second article was instructional, and did cast doubt on commonly held thoughts on p-value interpritation, but was overall was not opinionated. This negativity is likely due to the commonly word used "mean", here of course refering to average as opposed to being used as an adjective. This is a perfect illustration of the limitations of text analysis misinterpreting words and not being able to take context into account.

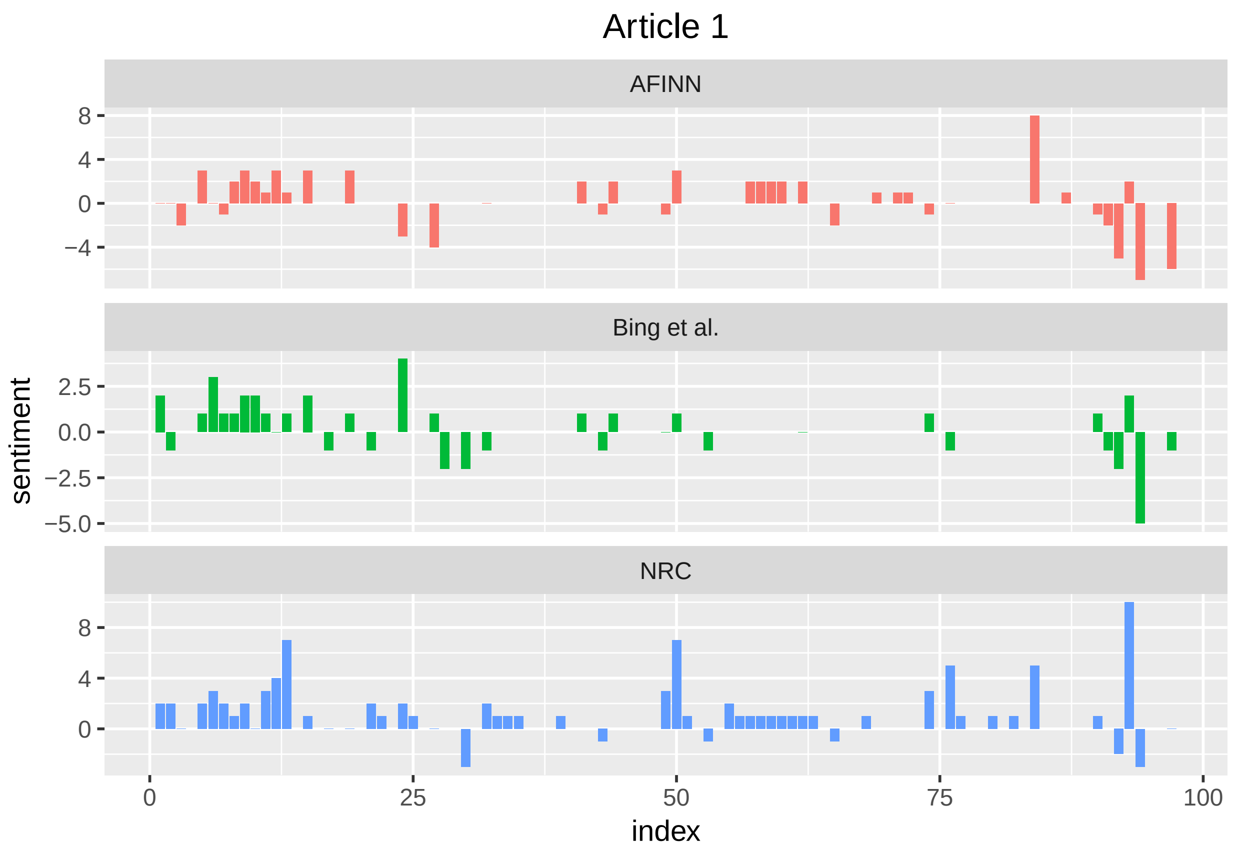
```
## # A tibble: 11 x 2
##    word          n
##    <chr>     <int>
##  1 true          9
##  2 create        3
##  3 simplify      3
##  4 vote          3
##  5 clean         2
##  6 confident     2
##  7 friendly      1
##  8 happy         1
##  9 money         1
## 10 powerful      1
## 11 shopping      1
```

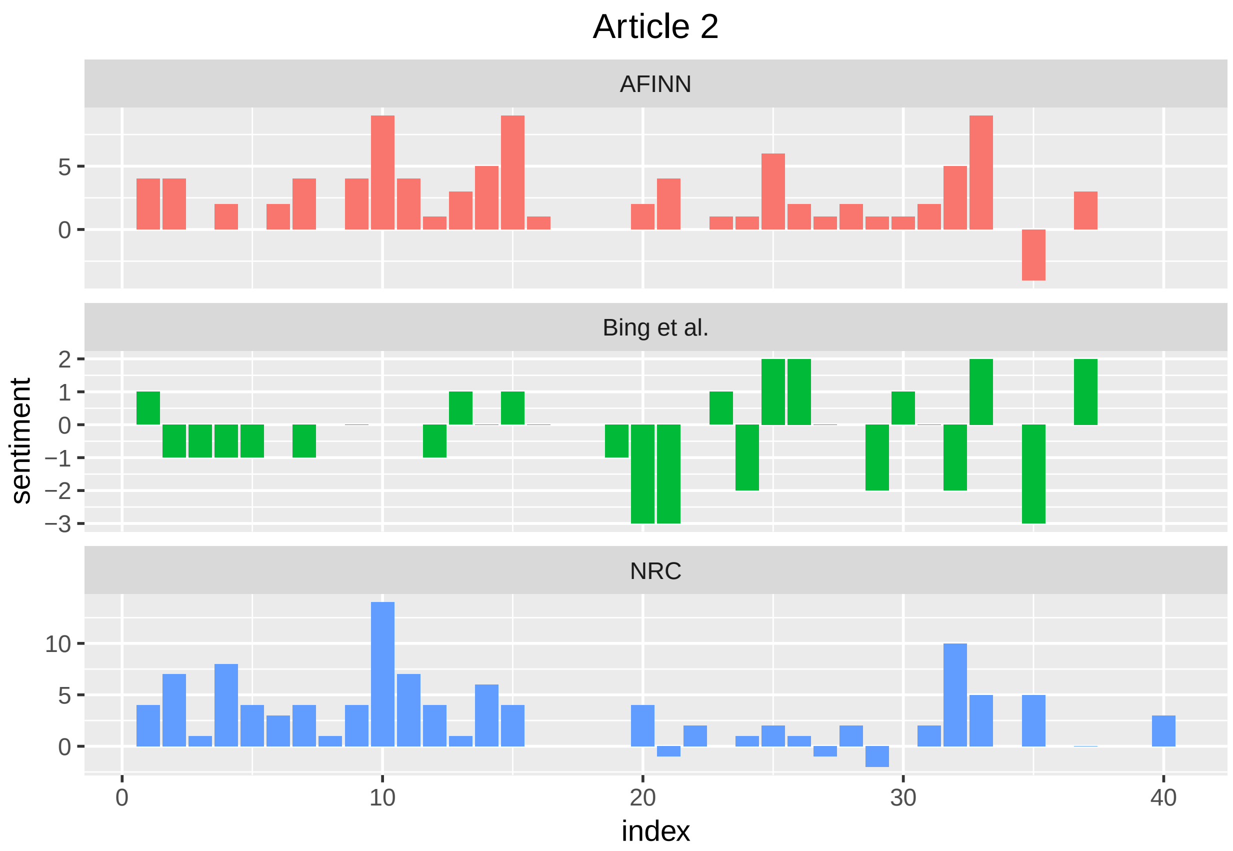
```
## # A tibble: 3 x 2
##   word           n
##   <chr>      <int>
## 1 true          38
## 2 complement     1
## 3 finally        1
```





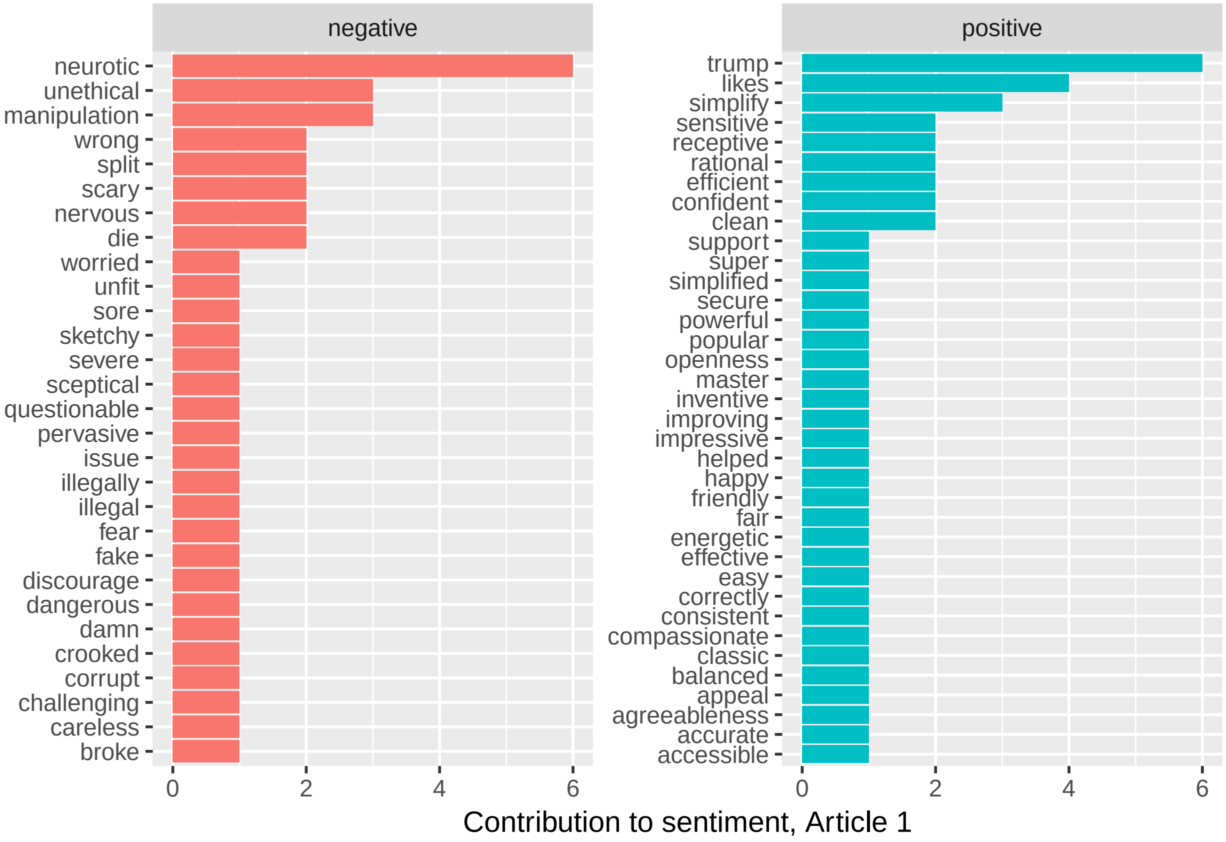
The analysis above was re-preformed comparing all of the sentiment lexicons. Again, the second article is shown to be shorter than the first article. However, the second article seems to be more positive in these results than when the net sentiment was calculated. As in the original analysi, the first article has longer stretches where no sentiment is recorded. This is likely because it contains instructions and code, which cannot be easily interprited in terms of emotion.

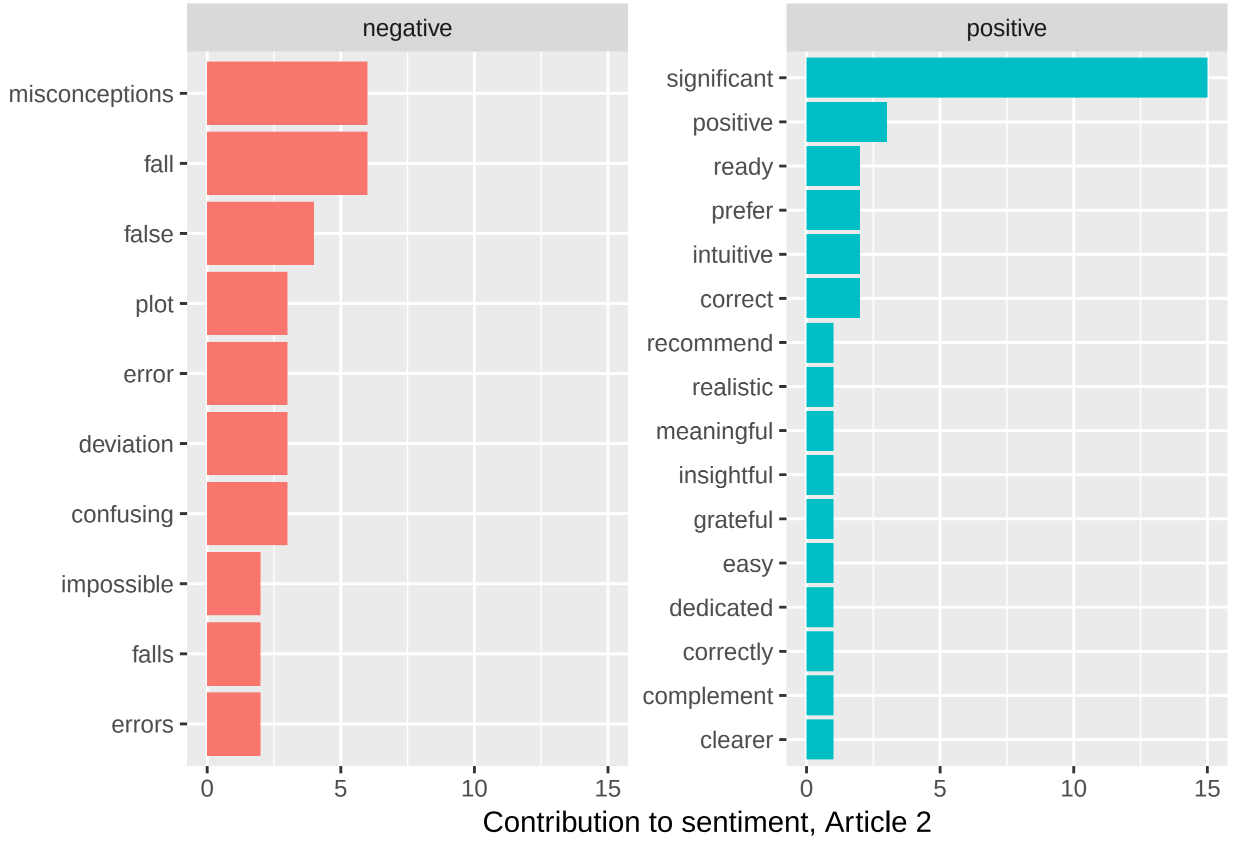

Article 1
AFINN
Bing et al.
NRC
sentiment
index
8
4
0
−4
2.5
0.0
−2.5
−5.0
8
4
0
0
25
50
75
100



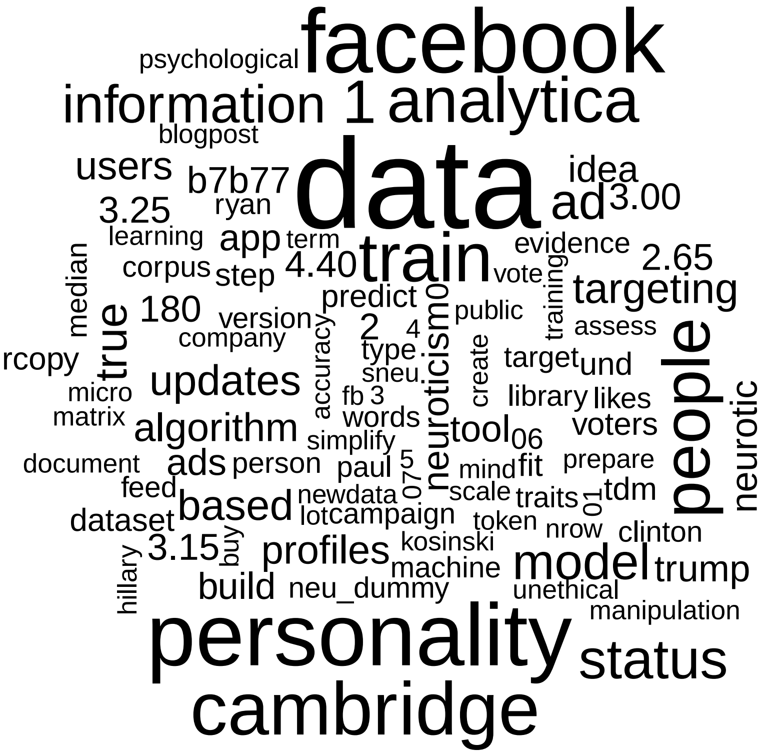
The most frequent words of each sentiment was presented in order of frequency in the graphs below. While some of the previous analyses have indicated that the second article is overall negative, there are more words associated with positive sentiment than negative, and one particular word, “significant”, makes up the majority of the counts. As this was an instructional statistical article, the word “significant” here should take on a neutral connotation. This may explain why some lexicons rated the second article more positive than when it was not broken down into specific indexes.

The results for the first article contains many more words than the second article, likely because it was more opinionated and not constrained by the need to use commonly known phrases related to it’s topic. There seems to be an even balance of negative and positive words, and all seem to be appropriately categorized, with one exception. The word “Trump” here refers to the name of a person, not the verb. While many people have strong opinions on Donald Trump, it should be treated neutrally in this analysis.


negative
neutoric
neurotic
unethical
manipulation
wrong
split
scary
nervous
die
worried
unfit
sore
sketchy
severe
sceptical
questionable
pervasive
issue
illegally
illegal
fear
fake
discourage
dangerous
damn
crooked
corrupt
challenging
careless
broke
positive
trump
likes
simplify
sensitive
receptive
rational
efficient
confident
clean
support
super
simplified
secure
powerful
popular
openness
master
inventive
improving
impressive
helped
happy
friendly
fair
energetic
effective
easy
correctly
consistent
compassionate
classic
balanced
appeal
agreeableness
accurate
accessible
0
2
4
6
0
2
4
6
Contribution to sentiment, Article 1

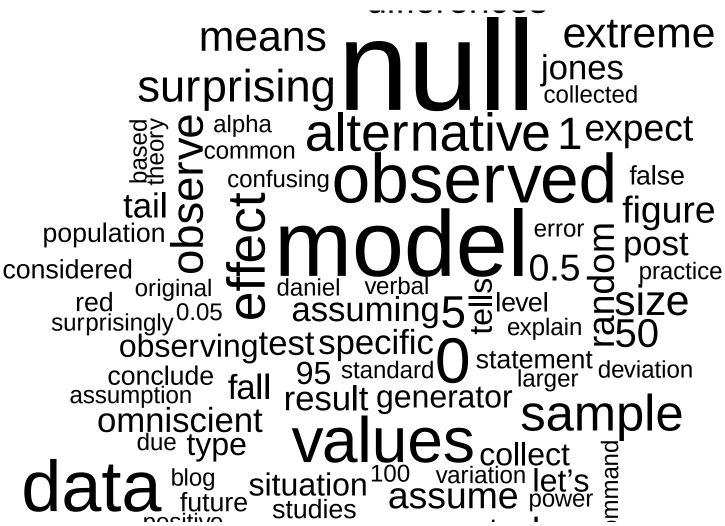


Word clouds for each article were created, showing both their most frequently used words and their most frequently used words that carry a sentiment. Both word clouds are unsuprising for their respective articles, and the main difference between the two that the first article repeats its words more often, as indicated by a higher frequency of large font for its word cloud. The word coulds coded for sentiment for both articles confirm what was presented in the ggplots above.

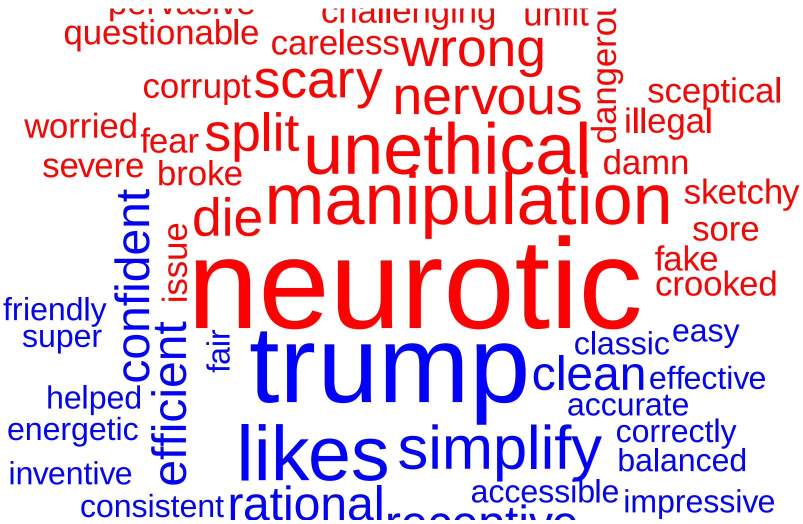


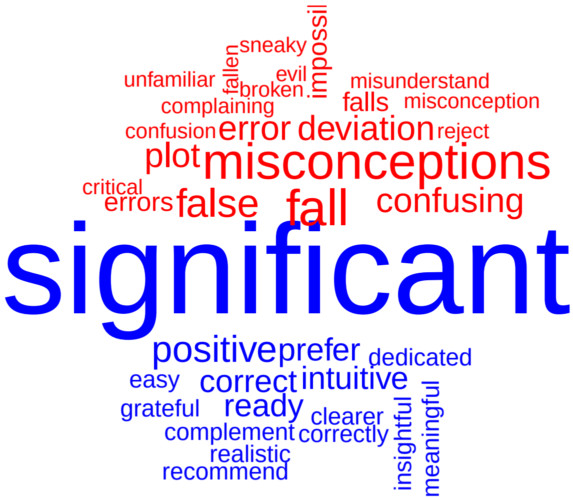
```
## Warning in wordcloud(word, n, max.words = 100): hypothesis could not be fit
## on page. It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): difference could not be fit
## on page. It will not be plotted.
```



```
## Warning in comparison.cloud(., colors = c("red", "blue"), max.words = 100):
## compassionate could not be fit on page. It will not be plotted.
```


pervasive
challenging
unfit
questionable
careless
wrong
dangerous
corrupt
scary
nervous
sceptical
worried
fear
split
unethical
illegal
severe
broke
damn
manipulation
sketchy
die
sore
confident
issue
neurotic
fake
crooked
friendly
super
fair
easy
classic
trump
efficient
clean
effective
helped
accurate
energetic
correctly
likes
simplify
balanced
inventive
accessible
consistent
rational
receptive
impressive

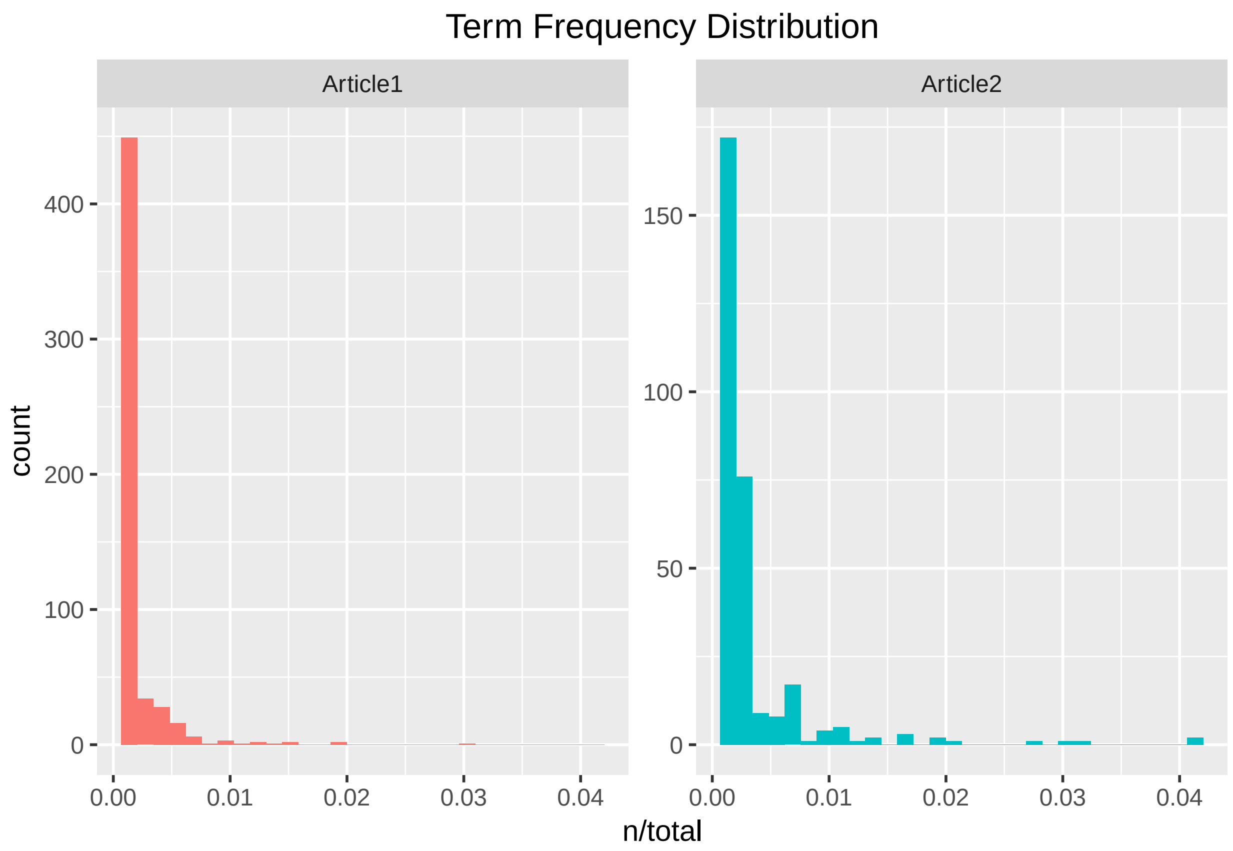


Chapter3

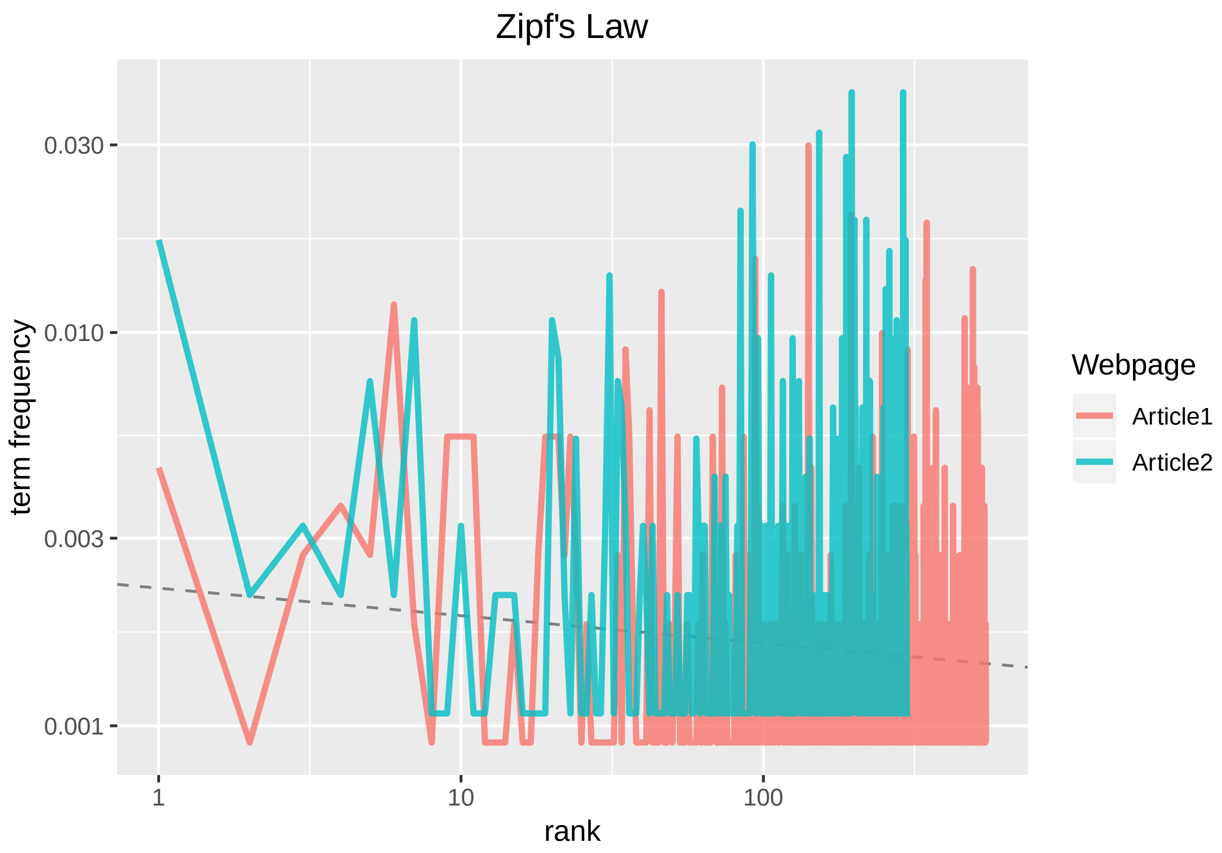
The term frequency distribution plots the count of the number of words who's frequency of use increases as the graph moves to the right. This results in the peak towards the left of the graph indicating the number of words that are not used very often. The results for article 1 indicate that it has more words than article 2. It also shows that it uses a greater variety of words, as there are few data points as the frequency of use increases. However, based on the results of the previously produced graphs, it is suprising that there are a few more peaks, though admitidly small ones, towards the left of the x-axis.

Zipf's law states that the frequency of a word's use is inversly proportioned to its rank. In other words, commonly used words that hold little meaning, such as "and" or "the" have a low rank, and scarcely used words have a higher rank. This would, in theory, cause a negative trend line. The two articles were graphed together, and though they overlap eachother fairly well, I would not say they are particularly comparable. The trendline is indeed negative, though the slope is not particularly steep. Both of these characteristics can be atributed to the shortness of the texts, and one being much more instructionally centered than the other.

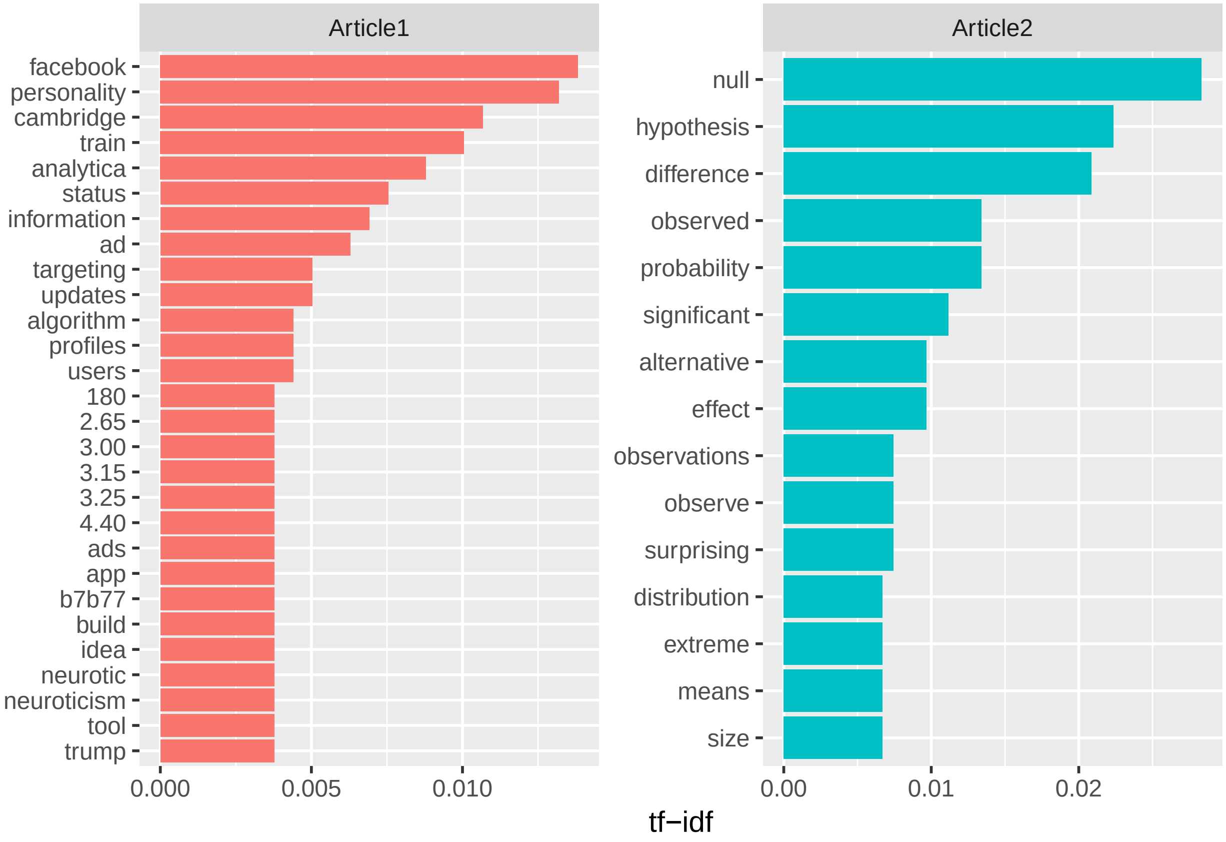
Term frequency-inverse document frequency (tf-idf), is used to measure how important a particular word is in a document out of a colection of documents, with more weight given to infrequently used words, and less weight given to commonly used words, implementing the theory of Zipf's law. Thus, the function will find words that are common to a text, but not so common that they have little significance. The results of the Cambridge Analytica article are not suprising, and are words that any article on the topic would contain. However, some of the results of the code run in the article are present in this analysis, causing sevral numbers to be included in the results. The resutls of the second article are also appropriate for the topic, and are largely words that describe statistical terms. Both results show words that are prominent in the word cloud. The results of the tf-idf analysis can again be seen with the highest scoring words in the final plot, though it should be noted that their scales are different before once compares them.

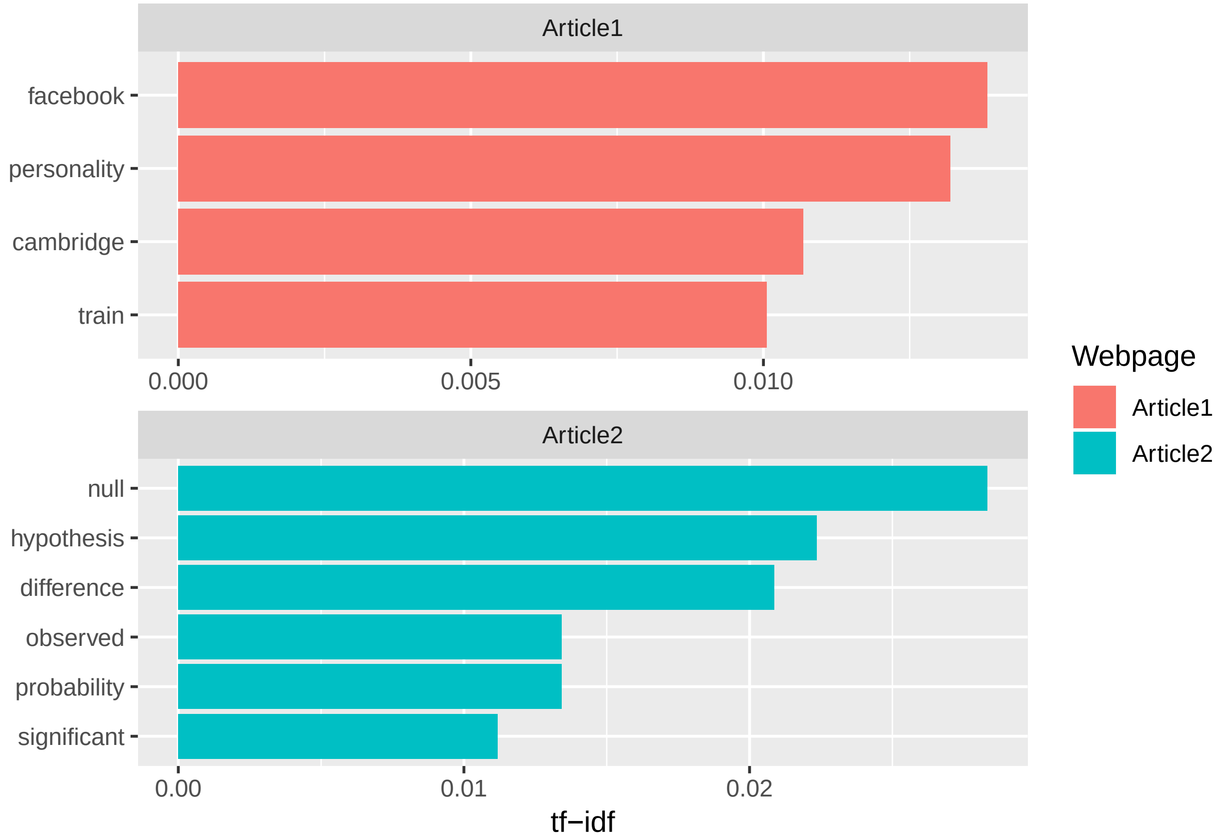


```
##
## Call:
## lm(formula = log10(`term frequency`) ~ log10(rank), data = rank_subset)
##
## Coefficients:
## (Intercept)  log10(rank)
##    -2.65425     -0.07384
```



```
## # A tibble: 852 x 6
##    Webpage  word             n     tf   idf tf_idf
##    <chr>    <chr>        <int>  <dbl> <dbl>  <dbl>
##  1 Article2 null            38 0.0409 0.693 0.0283
##  2 Article2 hypothesis      30 0.0323 0.693 0.0224
##  3 Article2 difference      28 0.0301 0.693 0.0209
##  4 Article1 facebook        22 0.0199 0.693 0.0138
##  5 Article2 observed        18 0.0194 0.693 0.0134
##  6 Article2 probability     18 0.0194 0.693 0.0134
##  7 Article1 personality     21 0.0190 0.693 0.0132
##  8 Article2 significant     15 0.0161 0.693 0.0112
##  9 Article1 cambridge       17 0.0154 0.693 0.0107
## 10 Article1 train           16 0.0145 0.693 0.0101
## # ... with 842 more rows
```


Article1
facebook
personality
cambridge
train
analytica
status
information
ad
targeting
updates
algorithm
profiles
users
180
2.65
3.00
3.15
3.25
4.40
ads
app
b7b77
build
idea
neurotic
neuroticism
tool
trump
0.000
0.005
0.010
Article2
null
hypothesis
difference
observed
probability
significant
alternative
effect
observations
observe
surprising
distribution
extreme
means
size
0.00
0.01
0.02
tf−idf

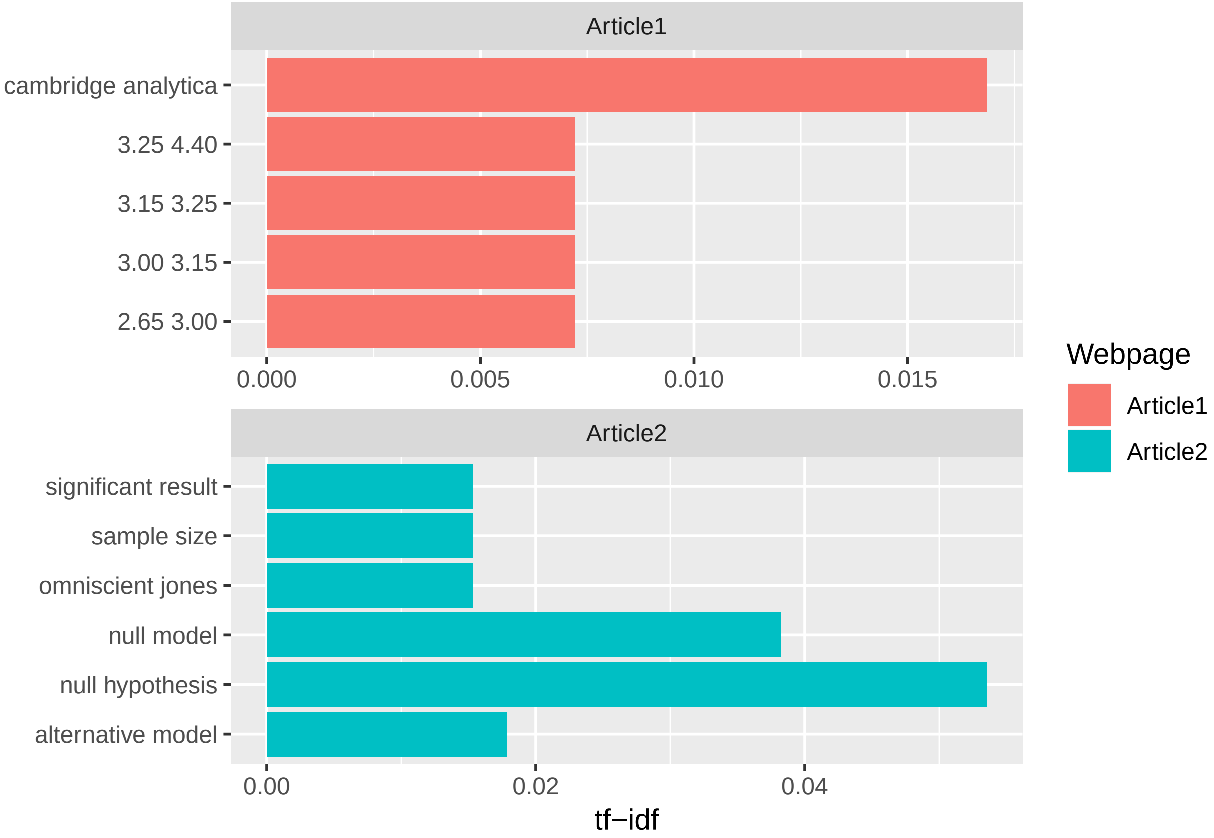


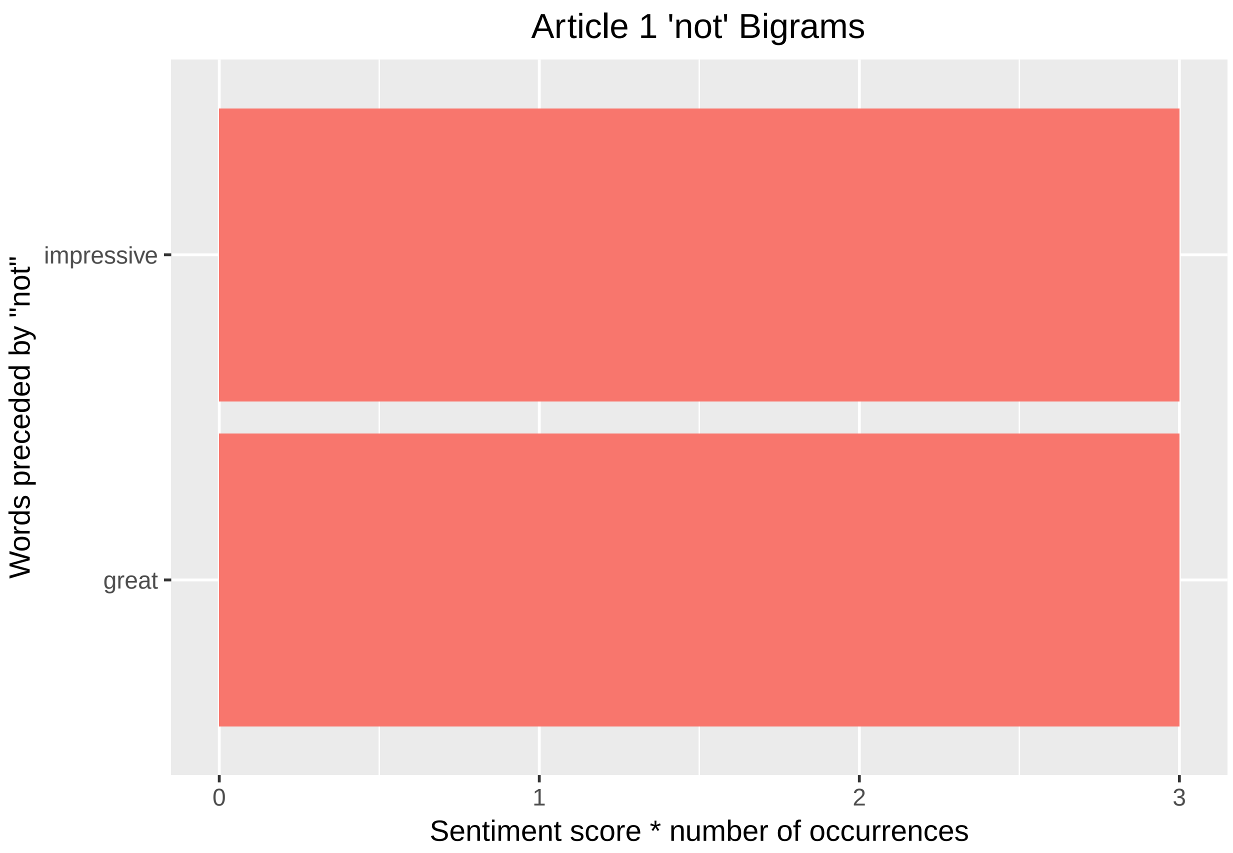
Chapter4

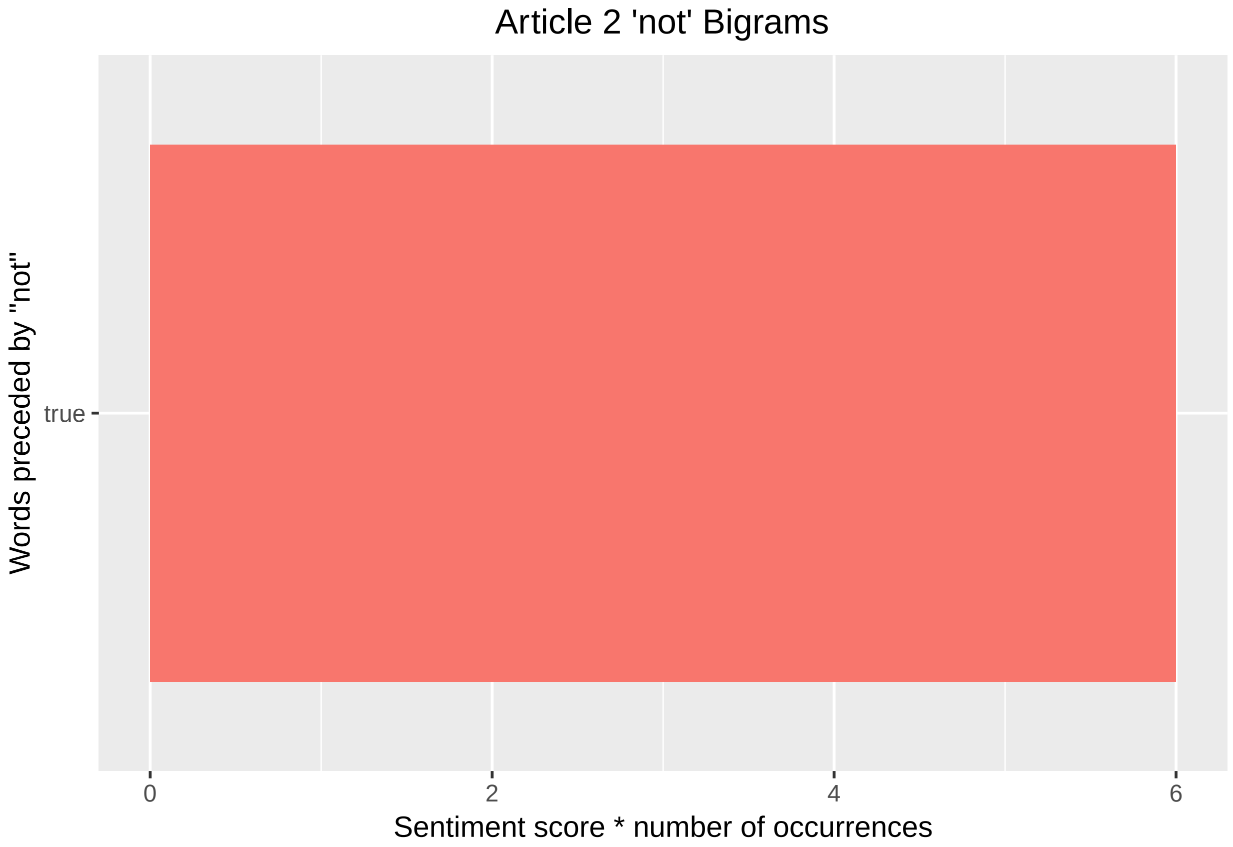
The tf-idf analysis was repeated for bigrams. Unfortunatly, due to the example code being printed in the first article and analyzed by the text reader, many of the bigrams are numbers, and do not actually have any significance. The results of the second article are as expected.

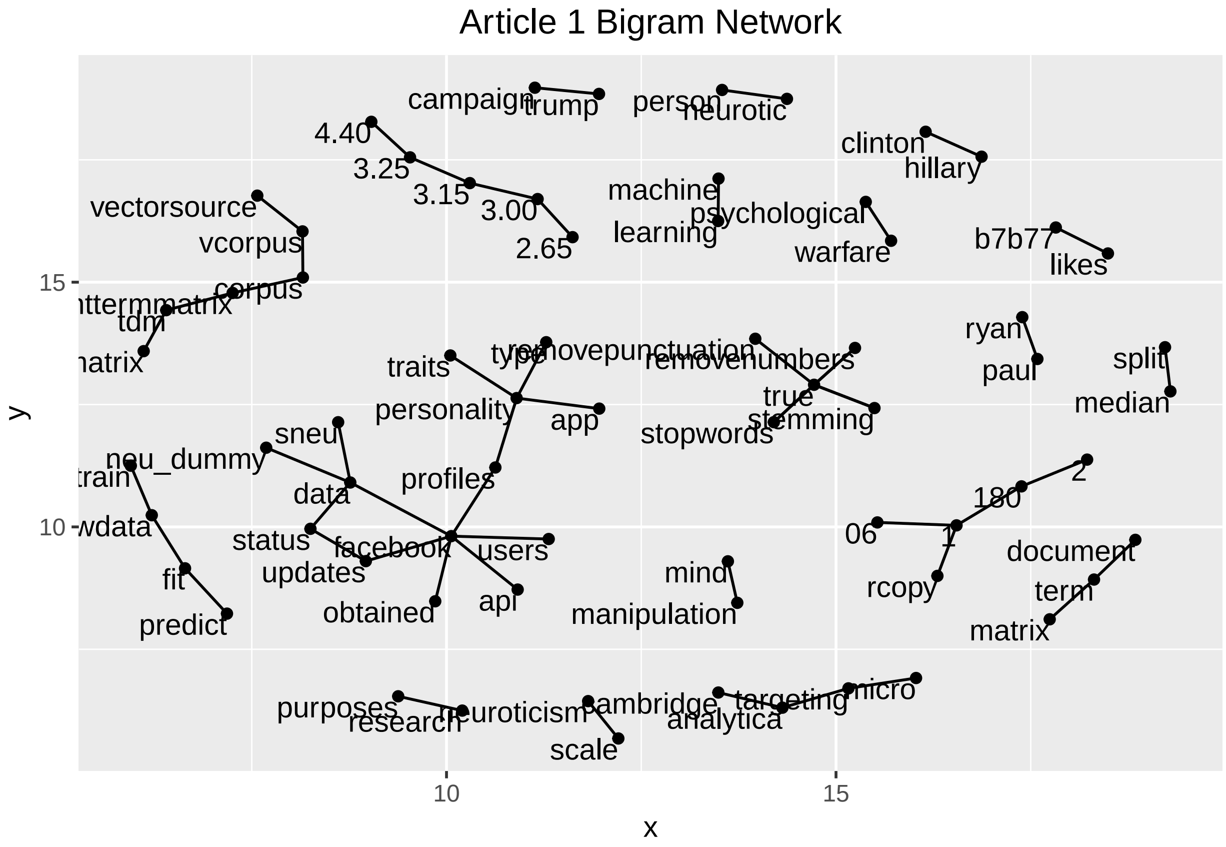
The “Article 1 ‘not’ Bigrams” shows the word most commonly following the word ‘not’. This was repeated for article 2. The results are not particularly varied, due to the shortness of the texts.

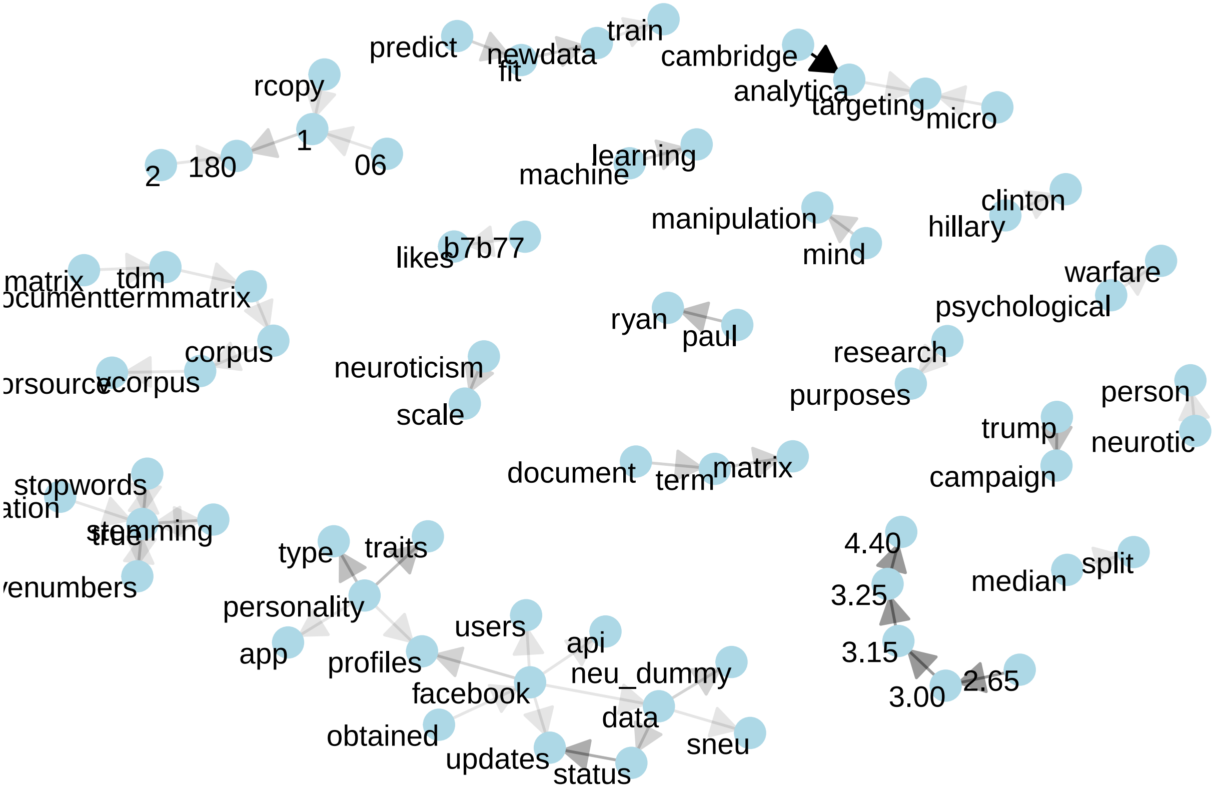
Bigram networks were created for each article, which indicate which commonly used words were followed by another commonly used word. Some words, such as “facebook” in the first article, were often followed by a variety of words, such as “data”, “users”, and “profiles”. In this example, “facebook” was the center of the node, and the link connecting it to other words is the edge. In order to create a more informative graphic, arrows were added to show the direction the edge was moving, and the arrows were colored to show the weight of the bigram. The results for article 2 show that there are a few commonly used chains of words, and not many nodes. Article 1 has slightly more interesting results as there are more nodes with centers. However, there are again many numbers included, which are not meaningful.


Article1
cambridge analytica
3.25 4.40
3.15 3.25
3.00 3.15
2.65 3.00
0.000
0.005
0.010
0.015
Article2
significant result
sample size
omniscient jones
null model
null hypothesis
alternative model
0.00
0.02
0.04
tf−idf
Webpage
Article1
Article2

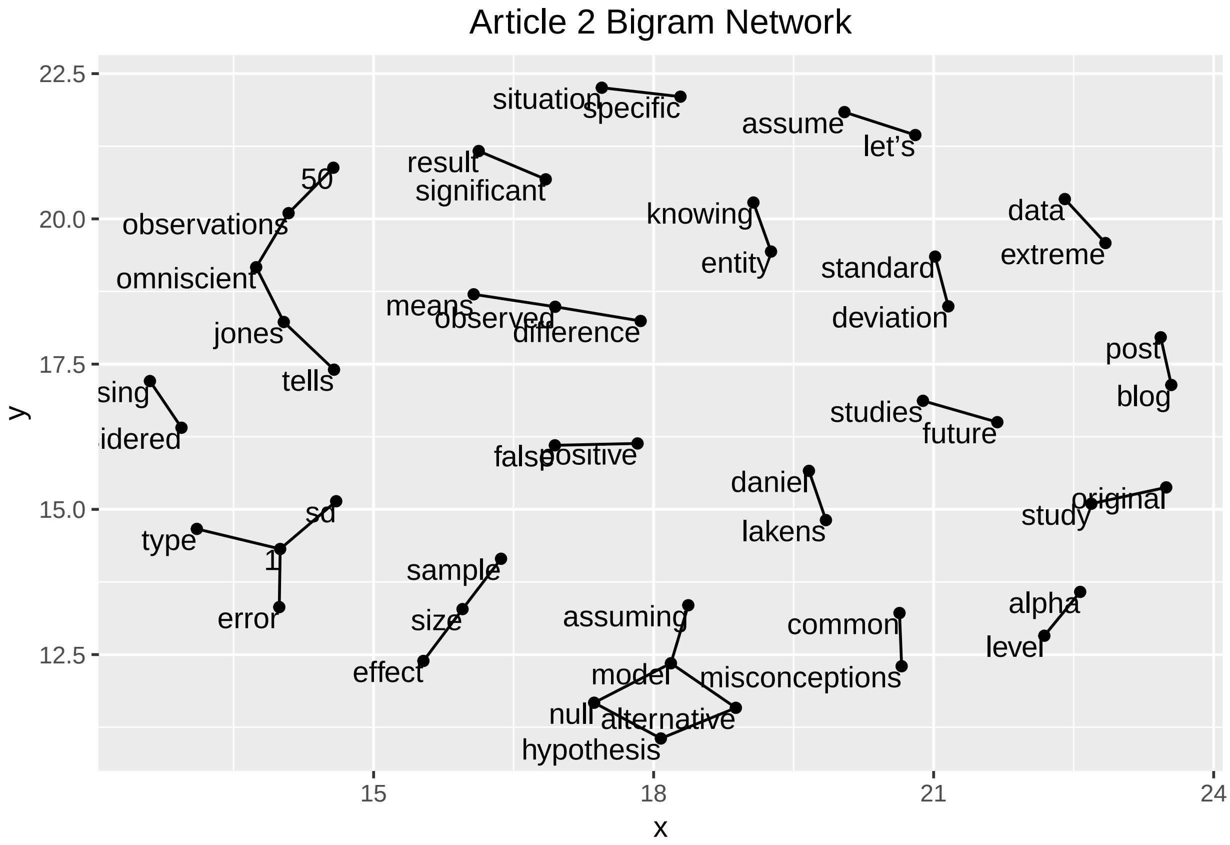

Article 1 'not' Bigrams
Words preceded by "not"
impressive
great
0
1
2
3
Sentiment score * number of occurrences

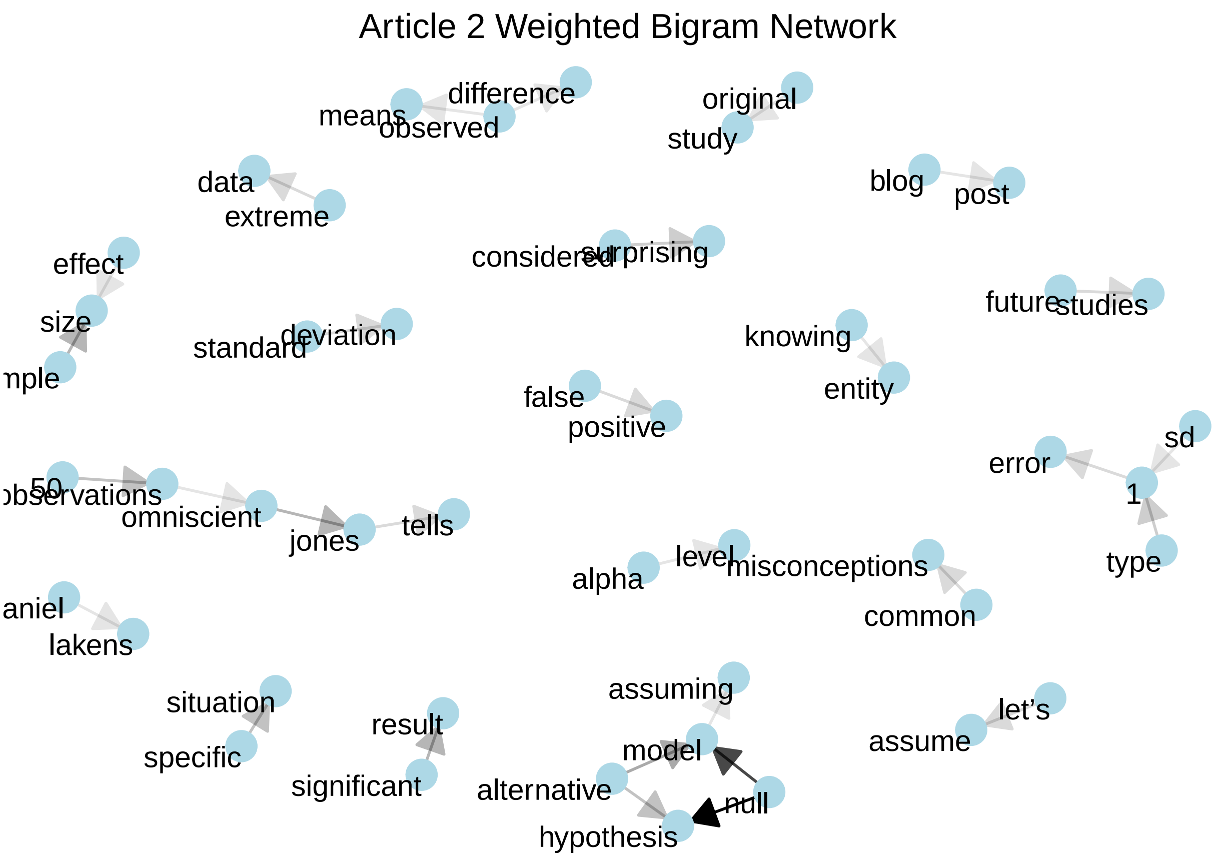

Article 2 'not' Bigrams
Words preceded by "not"
true
0
2
4
6
Sentiment score * number of occurrences


Article 1 Bigram Network
campaign
trump
person
neurotic
4.40
3.25
3.15
3.00
2.65
clinton
hillary
machine
learning
psychological
warfare
b7b77
likes
vectorsource
vcorpus
corpus
ntermmatrix
tdm
natrix
ryan
paul
split
median
traits
type
removepunctuation
removenumbers
true
stemming
stopwords
personality
app
sneu
neu_dummy
train
data
profiles
wdata
status
facebook
users
updates
obtained
api
fit
predict
mind
manipulation
180
2
06
1
rcopy
document
term
matrix
purposes
research
neuroticism
cambridge
scale
analytica
targeting
micro
15
10
y
10
15
x



Article 1 Weighted Bigram Network


Article 2 Bigram Network



Conclusion

Text mining is a great way to systematically analyze and compare text documents in R. While there are limitations, such as lexicons not taking the different meanings of words into account, tidytext is still a versitile method of analysis.