DATA607 Natural Language Processing

2021-10-31

Contents

Introduction **1**

Chapter 2 Sentient Analysis Example **1**

Import data **9**

Tiddy and Tranform Data **9**

Modeling **9**

Visualize (full speach) **10**

visualization of word choice **12**

comparison of lexicons (afinn vs syuzhet) **14**

Conclusion **15**

References **15**

Introduction

In this assignment I will be replicating the Sentiment Analysis example from Chapter 2. I will also be extending the example to include another corpus using a new lexicon. For the purposes of this assignment I have selected a the 2009 State of the Union Address and the Syuzhet lexicon.

- Corpus - I selected my corpus from a Kaggle data set that captures State of the Union Addresses from 1790 - 2018. https://www.kaggle.com/rtatman/tutorial-sentiment-analysis-in-r/data
- Lexicon - I selected the from the syuzhet sentiment extractor from the Syuzhet package https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html

Chapter 2 Sentient Analysis Example

This section recreates the sentiment analysis example in chapter 2 of the Text Minging with R textbook. It includes examples of 3 different lexicons bing, afinn and nrc.

```
#*********************************************************************************************
#    Title: Text Mining with R : 02-sentiment-analysis
#    Author: Julia Silge and David Robinson
#    Date: Apr 6, 2021
#    Code version: #93
#    Availability: https://github.com/dgrtwo/tidy-text-mining/blob/master/02-sentiment-analysis.Rmd
#*********************************************************************************************/

get_sentiments("afinn")
```

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # ... with 2,467 more rows
```

```
get_sentiments("bing")
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # ... with 6,776 more rows
```

```
get_sentiments("nrc")
```

```
## # A tibble: 13,875 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
```

```
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## # ... with 13,865 more rows
```

```
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
        linenumber = row_number(),
        chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]",
        ignore_case = TRUE)))) %>%
  ungroup() %>%
  unnest_tokens(word, text)

nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

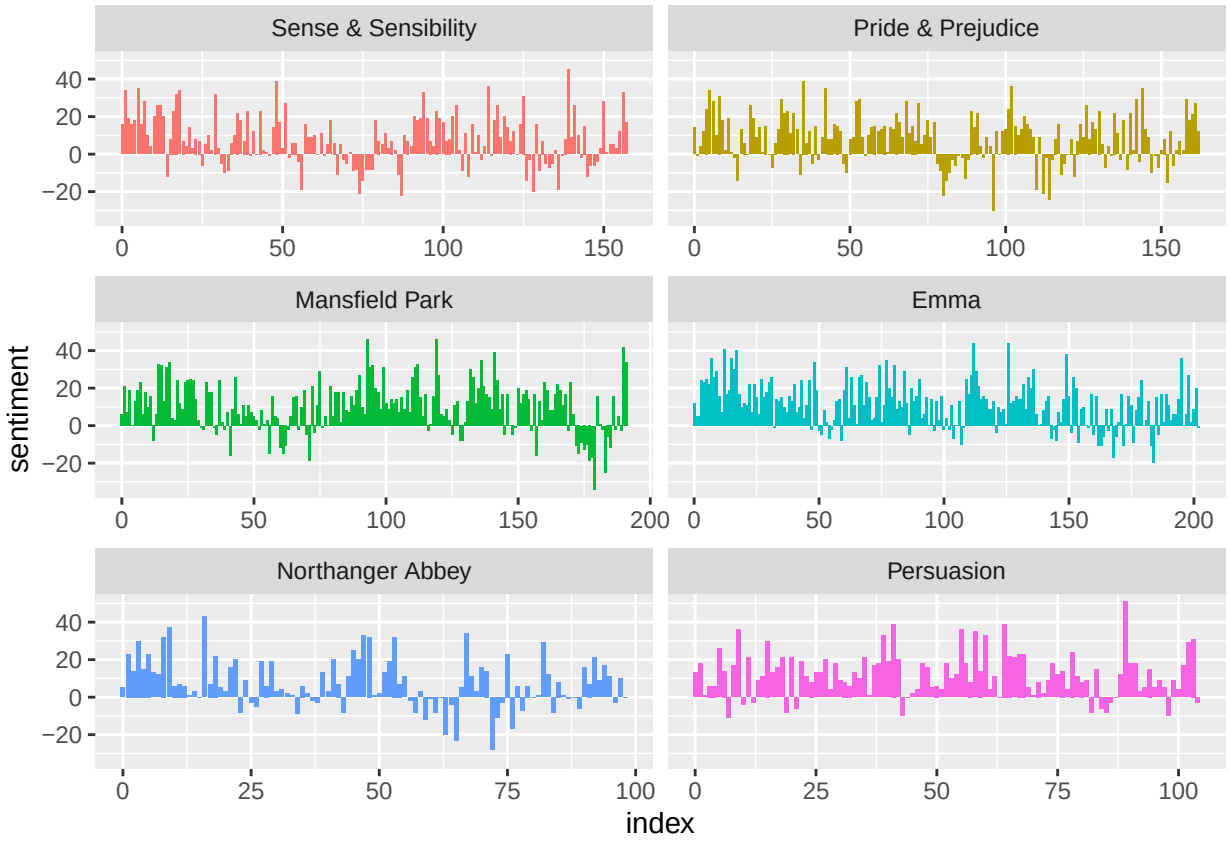
```
## Joining, by = "word"
```

```
## # A tibble: 301 x 2
##    word          n
##    <chr>     <int>
##  1 good        359
##  2 friend      166
##  3 hope        143
##  4 happy       125
##  5 love        117
##  6 deal         92
##  7 found        92
##  8 present      89
##  9 kind         82
## 10 happiness    76
## # ... with 291 more rows
```

```
janeaustensentiment <- tidy_books %>%
    inner_join(get_sentiments("bing")) %>%
    count(book, index = linenumber %/% 80, sentiment) %>%
    spread(sentiment, n, fill = 0) %>%
    mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
ggplot(janeaustensentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")
```




```
pride_prejudice <- tidy_books %>%
  filter(book == "Pride & Prejudice")

pride_prejudice
```

```
## # A tibble: 122,204 x 4
##    book               linenumber chapter word
##    <fct>                   <int>   <int> <chr>
##  1 Pride & Prejudice           1       0 pride
##  2 Pride & Prejudice           1       0 and
##  3 Pride & Prejudice           1       0 prejudice
##  4 Pride & Prejudice           3       0 by
##  5 Pride & Prejudice           3       0 jane
##  6 Pride & Prejudice           3       0 austen
##  7 Pride & Prejudice           7       1 chapter
##  8 Pride & Prejudice           7       1 1
##  9 Pride & Prejudice          10       1 it
## 10 Pride & Prejudice          10       1 is
## # ... with 122,194 more rows
```

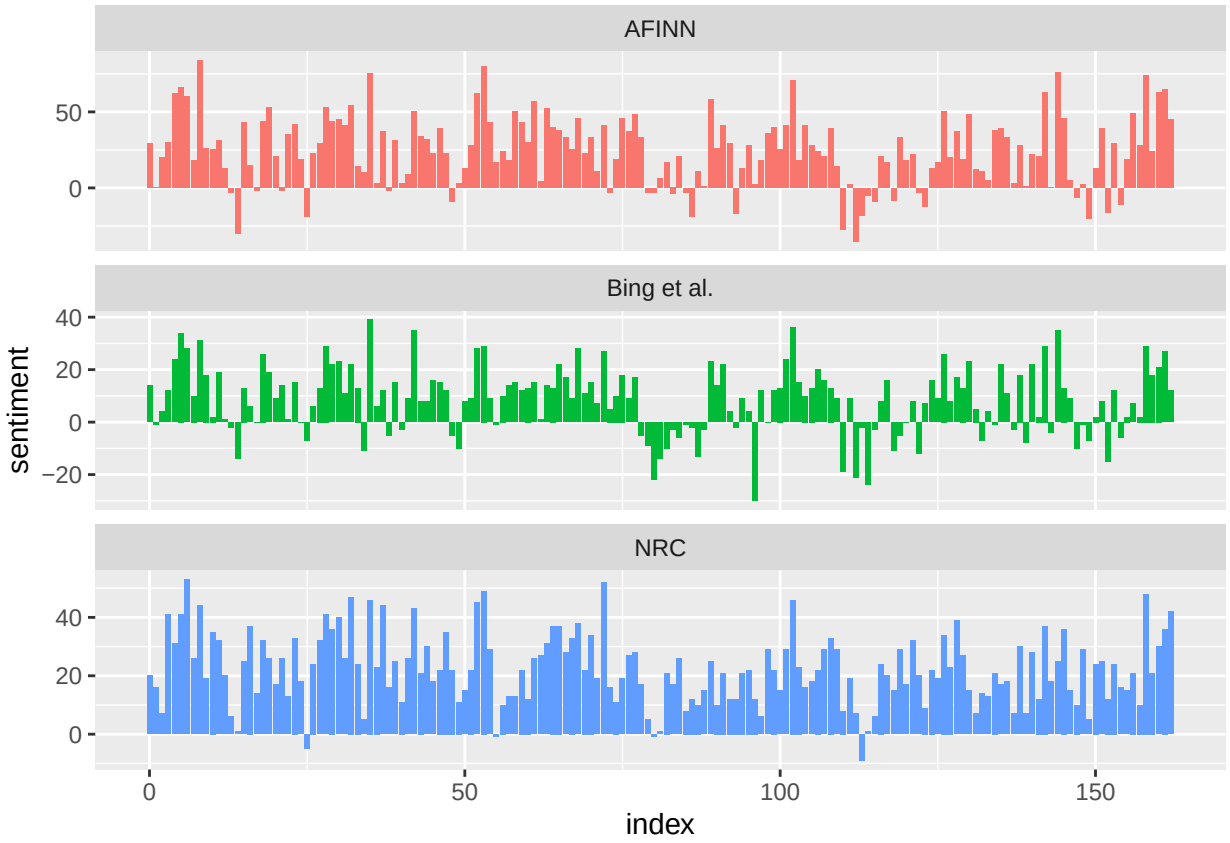
```
afinn <- pride_prejudice %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
bing_and_nrc <- bind_rows(
    pride_prejudice %>%
        inner_join(get_sentiments("bing")) %>%
        mutate(method = "Bing et al."),
    pride_prejudice %>%
        inner_join(get_sentiments("nrc") %>%
                filter(sentiment %in% c("positive",
                                        "negative"))) %>%
    mutate(method = "NRC")) %>%
    count(method, index = linenumber %/% 80, sentiment) %>%
    spread(sentiment, n, fill = 0) %>%
    mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



```
get_sentiments("nrc") %>%
    filter(sentiment %in% c("positive","negative")) %>%
    count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   3318
## 2 positive   2308
```

```
get_sentiments("bing") %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4781
## 2 positive   2005
```

```
bing_word_counts <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

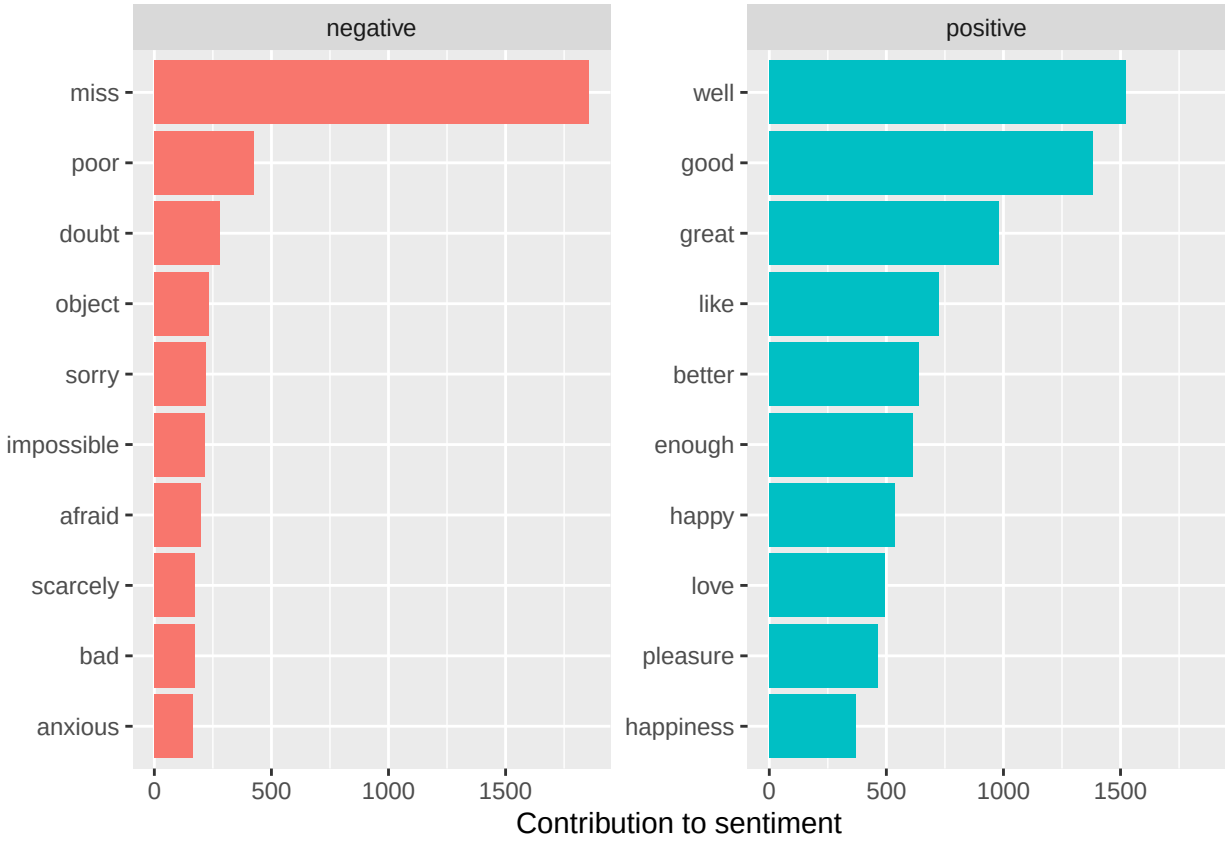
```
## Joining, by = "word"
```

```
bing_word_counts
```

```
## # A tibble: 2,585 x 3
##    word      sentiment     n
##    <chr>     <chr>     <int>
##  1 miss      negative   1855
##  2 well      positive   1523
##  3 good      positive   1380
##  4 great     positive    981
##  5 like      positive    725
##  6 better    positive    639
##  7 enough    positive    613
##  8 happy     positive    534
##  9 love      positive    495
## 10 pleasure  positive    462
## # ... with 2,575 more rows
```

```
bing_word_counts %>%
    group_by(sentiment) %>%
    top_n(10) %>%
    ungroup() %>%
    mutate(word = reorder(word, n)) %>%
    ggplot(aes(word, n, fill = sentiment)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~sentiment, scales = "free_y") +
    labs(y = "Contribution to sentiment",x = NULL) +
    coord_flip()
```

```
## Selecting by n
```




```
custom_stop_words <- bind_rows(data_frame(
                            word = c("miss"),
                            lexicon = c("custom")),
                            stop_words)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
custom_stop_words
```

```
## # A tibble: 1,150 x 2
##    word        lexicon
##    <chr>       <chr>
##  1 miss        custom
##  2 a           SMART
##  3 a's         SMART
##  4 able        SMART
##  5 about       SMART
##  6 above       SMART
##  7 according   SMART
```

```
##  8 accordingly SMART
##  9 across      SMART
## 10 actually    SMART
## # ... with 1,140 more rows
```

```
tidy_books %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

```
## Joining, by = "word"
```

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

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


```
#*****************************************************************************************
#    Title: Text Mining with R : 02-sentiment-analysis
#    End of Code block
#*****************************************************************************************/
```

Import data

I imported the data directly from the Kaggle's Google repository. I divided the speech up into sentences for the initial analysis.

```
file <- "https://storage.googleapis.com/kagglesdsdata/datasets/1660/2921/Obama_2009.txt?X-Goog-Algorithm

state_df <- read_delim(file, delim = "\\." , col_names = c("value"))
```

```
## Rows: 95 Columns: 1
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "\\."
## chr (1): value
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Tiddy and Tranform Data

The text data is clean but I added a line number for future reference. - add line number - unnest tokens - each row will represent a single word - filter out stop words

```
# add columns
state_df <- state_df %>%
    mutate(
        line_num = row_number()
    ) %>%
  unnest_tokens(word, value)



# filter stop words
pres_stop_words <- bind_rows(
    tibble(word = c("miss"),
    lexicon = c("pres")),
    stop_words
)

state_df <- state_df %>%
  anti_join(pres_stop_words)
```

```
## Joining, by = "word"
```

Modeling

The Syuzhut lexicon returns a vector of sentiment scores that includes 0.0 values. I filtered these scores out because they did not add to the analysis. I also tried to map the same sentiment scores to sentiment scores

for afinn lexicon - get sentiment from syuzhet lexicon - get sentiment from afinn lexicon - get stem word sentiment from syuzhet lexicon - calculate word data frame - calculate z scores for comparison purposes

```
# add the syuzhet sentiment
state_df$value_syuzhet <-  get_sentiment(state_df$word, method="syuzhet")
state_df <- state_df %>% filter(value_syuzhet != 0)


# add sentiments and stem words (afinn)
state_df <- state_df %>%
    left_join(get_sentiments("afinn")) %>%
    mutate(
      stem = wordStem(word)
    )
```

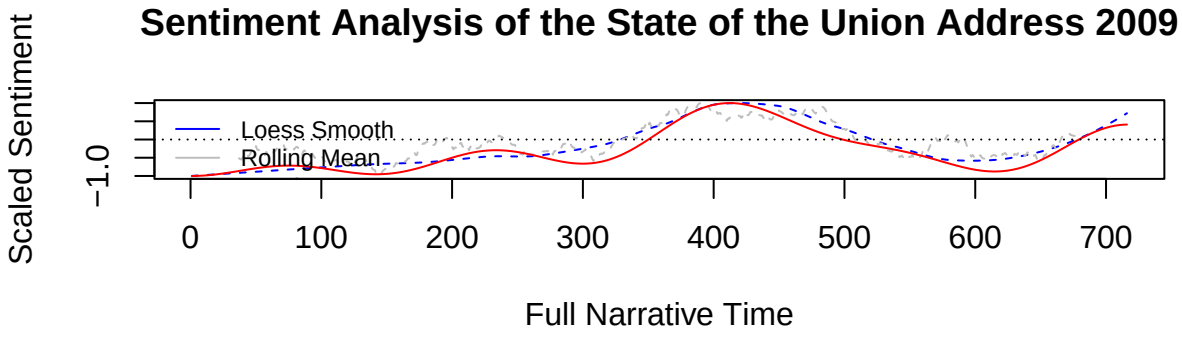
```
## Joining, by = "word"
```

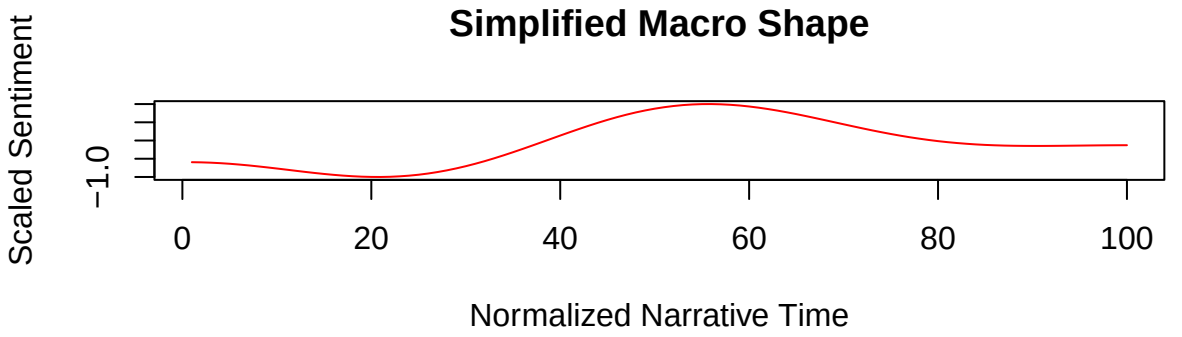
```
# add sentiments for stem words (afinn)
state_df$value_stem <-  get_sentiment(state_df$stem, method="syuzhet")


# create a word based dataframe
word_df <- state_df %>%
    select(-c(line_num,stem, value_stem)) %>%
    group_by(word) %>%
    mutate (
        freq = n(),
        val_freq = value * freq,
    ) %>%
    distinct()


# calculate the z values for comparision
m_value <- mean(word_df$value, na.rm = TRUE)
sd_value  <- sd(word_df$value, na.rm = TRUE)
m_value_syuzhet <- mean(word_df$value_syuzhet)
sd_value_syuzhet <- sd(word_df$value_syuzhet)

word_df <- word_df %>%
    mutate(
        z_value = (value - m_value) / sd_value,
        z_value_syuzhet = (value_syuzhet - m_value_syuzhet) / sd_value_syuzhet,
        z_diff = z_value - z_value_syuzhet
    )
```

Visualize (full speach)

The sentiment for the full state of the union narrative seems to capture the somber tone that was in the country at the time. With the Financial system and the housing market still in crisis the speech starts on a somber note but increases in positive before settling at a neutral tone. This is also reflected in the word cloud with a balance of negative words debt, crisis, recession with positive words such as opportunity, confidence and care

```
simple_plot(state_df$value_syuzhet,
            title = "Sentiment Analysis of the State of the Union Address 2009",
            legend_pos = "topleft"
            )
```






```
state_df %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, random.color=TRUE))
```

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

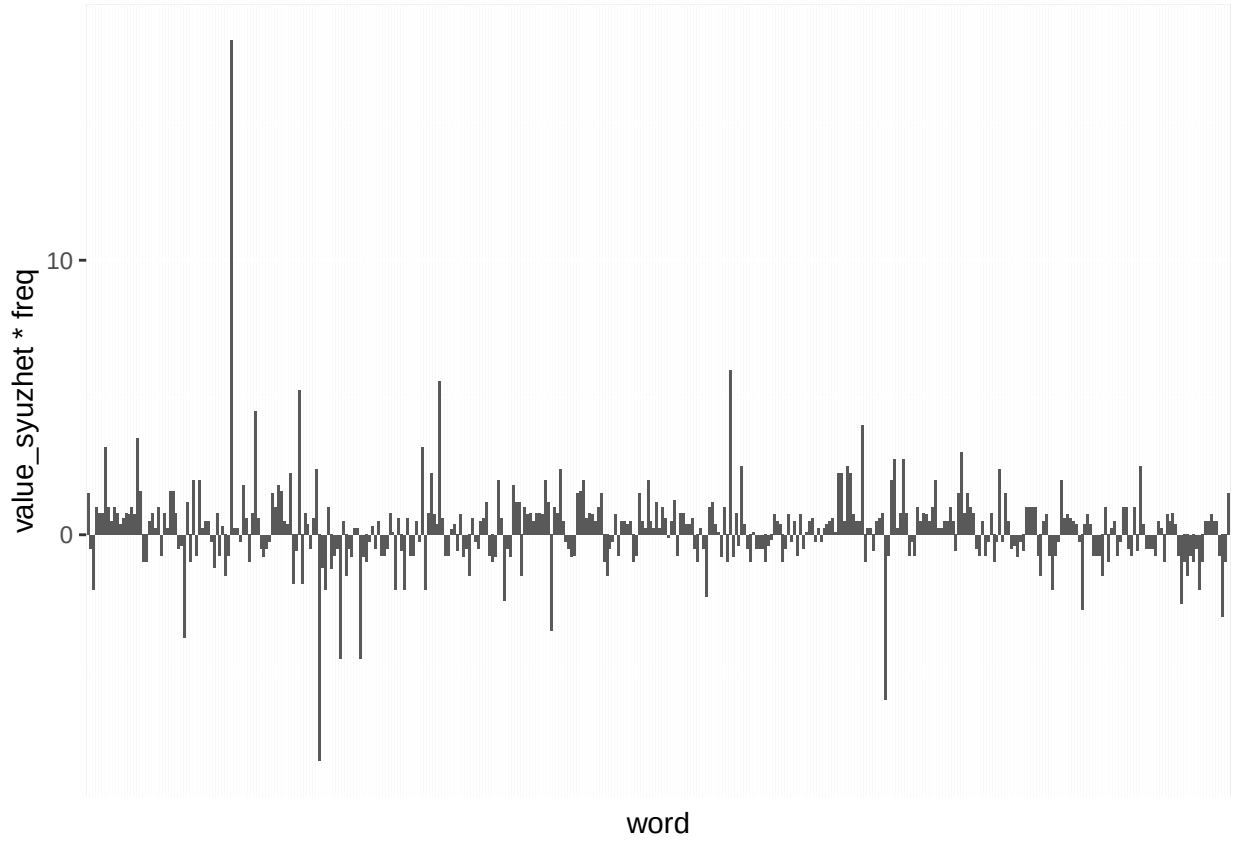
```
## Warning in wordcloud(word, n, max.words = 100, random.color = TRUE): government
## could not be fit on page. It will not be plotted.
```

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

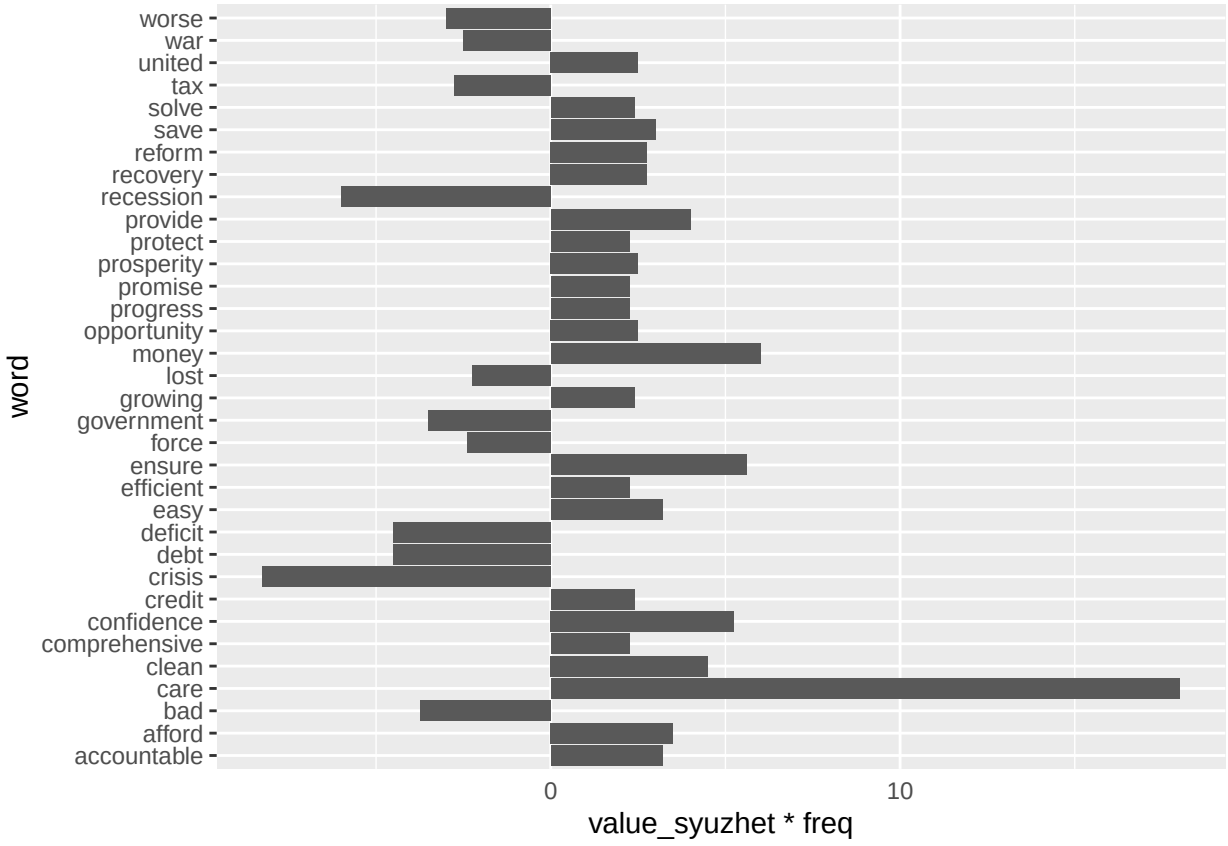
visualization of word choice

The whole speech had an emphasis on Care. It is the biggest work in the word cloud and if you look a frequency and sentiment it is by far the focus of the speech follow by crisis and recession.

```
word_df %>%
ggplot(aes(word, value_syuzhet*freq)) +
    geom_col(show.legend = FALSE) +
    scale_fill_viridis_c() +
    theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())
```



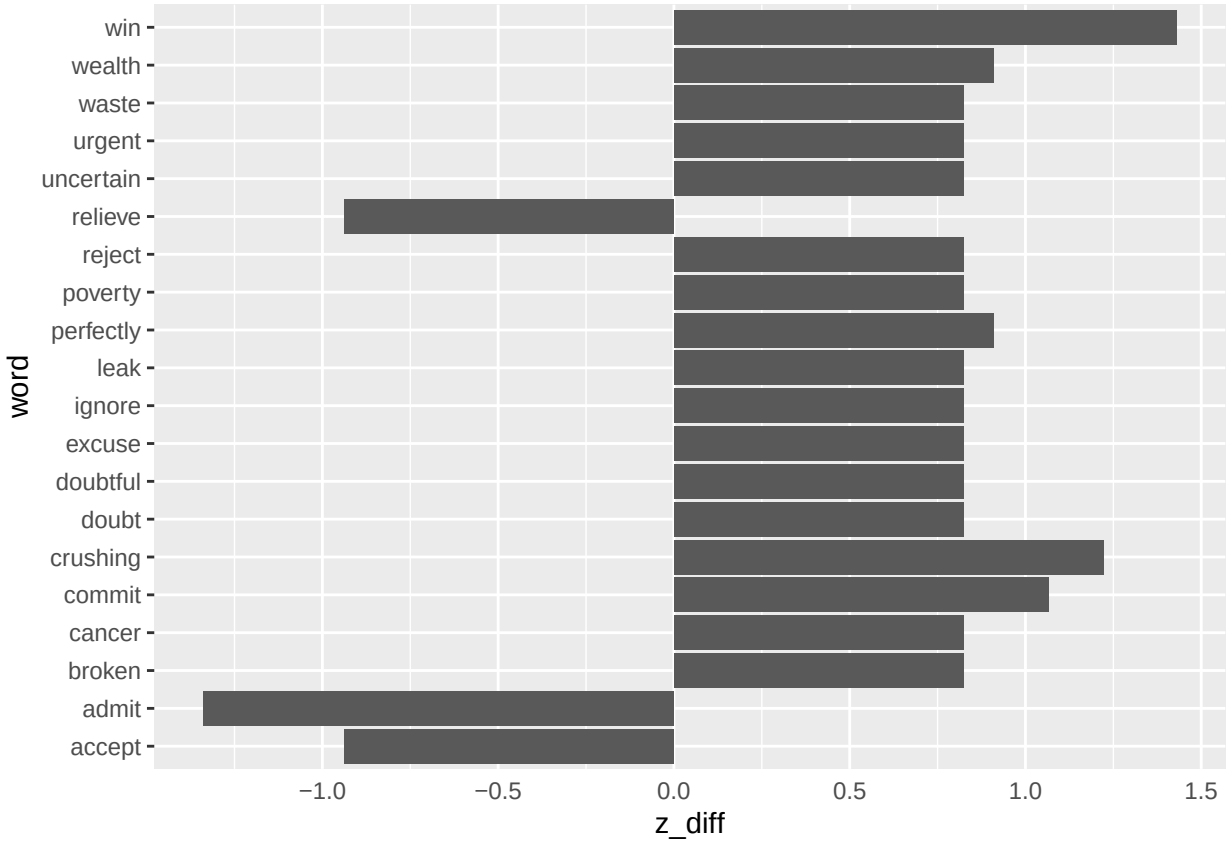

```
# words that are outliers in terms of sentiment and frequencey
word_df %>%
filter(abs(value_syuzhet*freq) > 2) %>%
ggplot(aes(word, value_syuzhet*freq)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    scale_fill_viridis_c() +
    theme(axis.ticks.x = element_blank(),
        axis.text.x = element_text())
```



comparison of lexicons (afinn vs syuzhet)

Calculating the z scores for each word sentiment values allow us to compare the relative sentiment scores by each lexicon using the same scale. What we can see is that the difference lexicons had discrepancies with the following words. With Afinn lexicon being more positive for words to the right of 0 and Syushet being more positive for word on the left of zero. Given this variability it is important to test more than on lexicon.

```
# words that are outliers in terms of sentiment and frequencey
word_df %>%
filter(abs(z_diff) > .8) %>%
ggplot(aes(word, z_diff)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    scale_fill_viridis_c() +
    theme(axis.ticks.x = element_blank(),
        axis.text.x = element_text())
```



Conclusion

The exercise highlighted a few areas of sentiment analysis that I wanted to focus on. Since I was working with a speech I started with the overall flow. Looking at how the sentiment changed overtime as the speech progressed. The sentiment scores seem consistent with the narrative arch of the speech. I also looked at the content overall focusing on word choice and frequency of word usage. It was interesting to see the focus on "Care" as a consistent theme across the speech. It seems like stem analysis added very little to the overall analysis. There was a low percentage of matches for the stem words. I would like to explore ways to increase the relevance of this analysis going forward.

References

- Silge, J and Robinson, D (2017) Text Mining with R: 02-sentiment-analysis (3) [source Code]. https://github.com/dgrtwo/tidy-text-mining/blob/master/02-sentiment-analysis.Rmd
- Tatman, R (2017) Tutorial: Sentiment Analysis in R: State of the Union Corpus (1790 - 2018) (8) [Dataset]. https://www.kaggle.com/rtatman/tutorial-sentiment-analysis-in-r/notebook