Project 1. Presidents’ Inaugural Speeches

Caroline Hao

9/18/2017

Step1. Preparation: Install needed Packages and load the libraries

Step2. Data: Read the speach and convert to the tidy data

Readed all the speeches and convert all the data into a tidy format for future sentiment analysis.

```
#Loaded inauguaral speeches from the following website
main.page <- read_html(x = "http://www.presidency.ucsb.edu/inaugurals.php")
inaug <- f.speechlinks(main.page)
inaug <- inaug[-nrow(inaug),]  # remove the last line, irrelevant due to error.
as.Date(inaug[,1], format="%B %d, %Y")
```

```
##  [1] "1789-04-30" "1793-03-04" "1797-03-04" "1801-03-04" "1805-03-04"
##  [6] "1809-03-04" "1813-03-04" "1817-03-04" "1821-03-04" "1825-03-04"
## [11] "1829-03-04" "1833-03-04" "1837-03-04" "1841-03-04" "1845-03-04"
## [16] "1849-03-05" "1853-03-04" "1857-03-04" "1861-03-04" "1865-03-04"
## [21] "1869-03-04" "1873-03-04" "1877-03-05" "1881-03-04" "1885-03-04"
## [26] "1889-03-04" "1893-03-04" "1897-03-04" "1901-03-04" "1905-03-04"
## [31] "1909-03-04" "1913-03-04" "1917-03-04" "1921-03-04" "1925-03-04"
## [36] "1929-03-04" "1933-03-04" "1937-01-20" "1941-01-20" "1945-01-20"
## [41] "1949-01-20" "1953-01-20" "1957-01-21" "1961-01-20" "1965-01-20"
## [46] "1969-01-20" "1973-01-20" "1977-01-20" "1981-01-20" "1985-01-21"
## [51] "1989-01-20" "1993-01-20" "1997-01-20" "2001-01-20" "2005-01-20"
## [56] "2009-01-20" "2013-01-21" "2017-01-20"
```

```
nrow(inaug)
```

```
## [1] 58
```

```
#58

# Loaded nomination speeches from the following website
main.page=read_html("http://www.presidency.ucsb.edu/nomination.php")
nomin <- f.speechlinks(main.page)
nomin <- nomin[-47,]   #Delete the nomin of Calvin Coolidge
nrow(nomin)
```

```
## [1] 54
```

```
#54

#Loaded farewell speeches from the following website
main.page=read_html("http://www.presidency.ucsb.edu/farewell_addresses.php")
farewell <- f.speechlinks(main.page)
nrow(farewell)
```

```
## [1] 13
```

```r
#13

#Readed the overall information
inaug.list=read.csv("~/Desktop/data/inauglist.csv", stringsAsFactors = FALSE)
nrow(inaug.list)
```

```
## [1] 58
```

```r
nomin.list=read.csv("~/Desktop/data/nominlist.csv", stringsAsFactors = FALSE)
nrow(nomin.list)
```

```
## [1] 54
```

```r
farewell.list=read.csv("~/Desktop/data/farewelllist.csv", stringsAsFactors = FALSE)
nrow(farewell.list)
```

```
## [1] 13
```

```r
#Combined all relevant information together
speech.list=rbind(inaug.list, nomin.list, farewell.list)
speech.list$type=c(rep("inaug", nrow(inaug.list)),
                   rep("nomin", nrow(nomin.list)),
                   rep("farewell", nrow(farewell.list)))
length(speech.list$type)
```

```
## [1] 125
```

```r
speech.url=rbind(inaug, nomin, farewell)
speech.list=cbind(speech.list, speech.url)

speech.list$fulltext=NA
for(i in seq(nrow(speech.list))) {
  text <- read_html(speech.list$urls[i]) %>%
    html_nodes(".displaytext") %>%
    html_text() # get the text
  speech.list$fulltext[i]=text
  filename <- paste0("~/Desktop/data/fulltext/",
                     speech.list$type[i],
                     speech.list$File[i], "-",
                     speech.list$Term[i], ".txt")
  sink(file = filename) %>%
  cat(text)
  sink()
}


# Converted into the tidy format
Speeches_text <- speech.list$fulltext
Presidents_Name <- speech.list $ President

series <- tibble()
for(i in seq_along(Presidents_Name)) {

        clean <- tibble(Term = seq_along(Speeches_text[[i]]),
                        text = Speeches_text[[i]]) %>%
             unnest_tokens(word, text) %>%
             mutate(Name = Presidents_Name[i]) %>%
             select(Name, everything())

        series <- rbind(series, clean)
}


series
```

```
## # A tibble: 401,470 x 3
##                  Name  Term     word
##                 <chr> <int>    <chr>
##  1 George Washington     1   fellow
##  2 George Washington     1 citizens
##  3 George Washington     1       of
##  4 George Washington     1      the
##  5 George Washington     1   senate
##  6 George Washington     1      and
##  7 George Washington     1       of
##  8 George Washington     1      the
##  9 George Washington     1    house
## 10 George Washington     1       of
## # ... with 401,460 more rows
```

Step3. Word frequency: Most frequent Words in latest six presidents

Now we want to get a version of the common words in each of the speech as well as the entire speeches of six presidents.

Step1. We remove the stop words (i.e. the, and, to, of, a, he, . . .) and start to find the top 10 frequent words in each president speech

Step2. Plot these words grouped by eah president

Step3. Calculate the frequency for each word across the entire six president speeches versus within each one of them. This will allow us to compare strong deviations of word frequency within each speech as compared to across the entire version.

Step4. Plot the graph of them

```
# Find the top 10 most frequent words of the last six presidents
series %>%
        anti_join(stop_words) %>%
        group_by(Name) %>%
        count(word, sort = TRUE) %>%
        top_n(10)
```

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

```
## Selecting by n
```

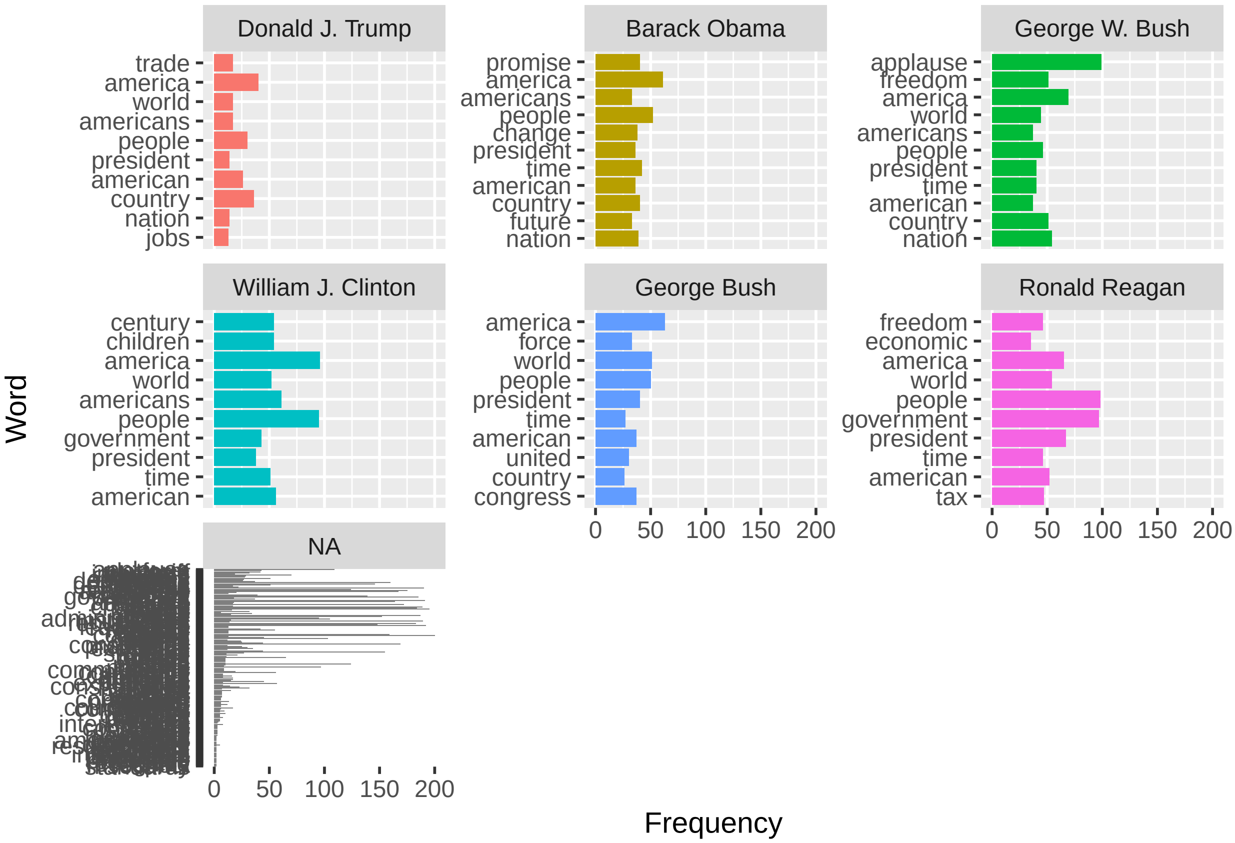
```
## # A tibble: 694 x 3
## # Groups:   Name [61]
##                   Name       word     n
##                  <chr>      <chr> <int>
##  1      Richard Nixon    america   160
##  2      Richard Nixon      world   120
##  3   Albert Gore, Jr.   applause   109
##  4        John McCain   applause   100
##  5     George W. Bush   applause    99
##  6      Ronald Reagan     people    98
##  7      Ronald Reagan government    97
##  8 William J. Clinton    america    96
##  9 William J. Clinton     people    95
## 10      Richard Nixon     people    92
## # ... with 684 more rows
```

```
Last_six = c("Donald J. Trump", "Barack Obama", "George W. Bush", "William J. Clinton", "George Bush",

series %>%
        anti_join(stop_words) %>%
        group_by(Name) %>%
        count(word, sort = TRUE) %>%
        top_n(10) %>%
        ungroup() %>%
        mutate(Name = factor(Name, levels = Last_six),
               text_order = nrow(.):1) %>%
        ggplot(aes(reorder(word, text_order), n, fill = Name)) +
          geom_bar(stat = "identity") +
          ylim(0,200) +
          facet_wrap(~ Name, scales = "free_y") +
          labs(x = "Word", y = "Frequency") +
          coord_flip() +
          theme(legend.position="none")
```

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

```
## Warning: Removed 226 rows containing missing values (geom_bar).
```



```
Word_propotion <- series %>%
        anti_join(stop_words) %>%
        count(word) %>%
        transmute(word, all_words = n / sum(n))
```

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

```
# calculate percent of word use within each speech
frequency <- series %>%
        anti_join(stop_words) %>%
        count(Name, word) %>%
        mutate(Pres_words = n / sum(n)) %>%
        left_join(Word_propotion) %>%
        arrange(desc(Pres_words)) %>%
        ungroup()
```

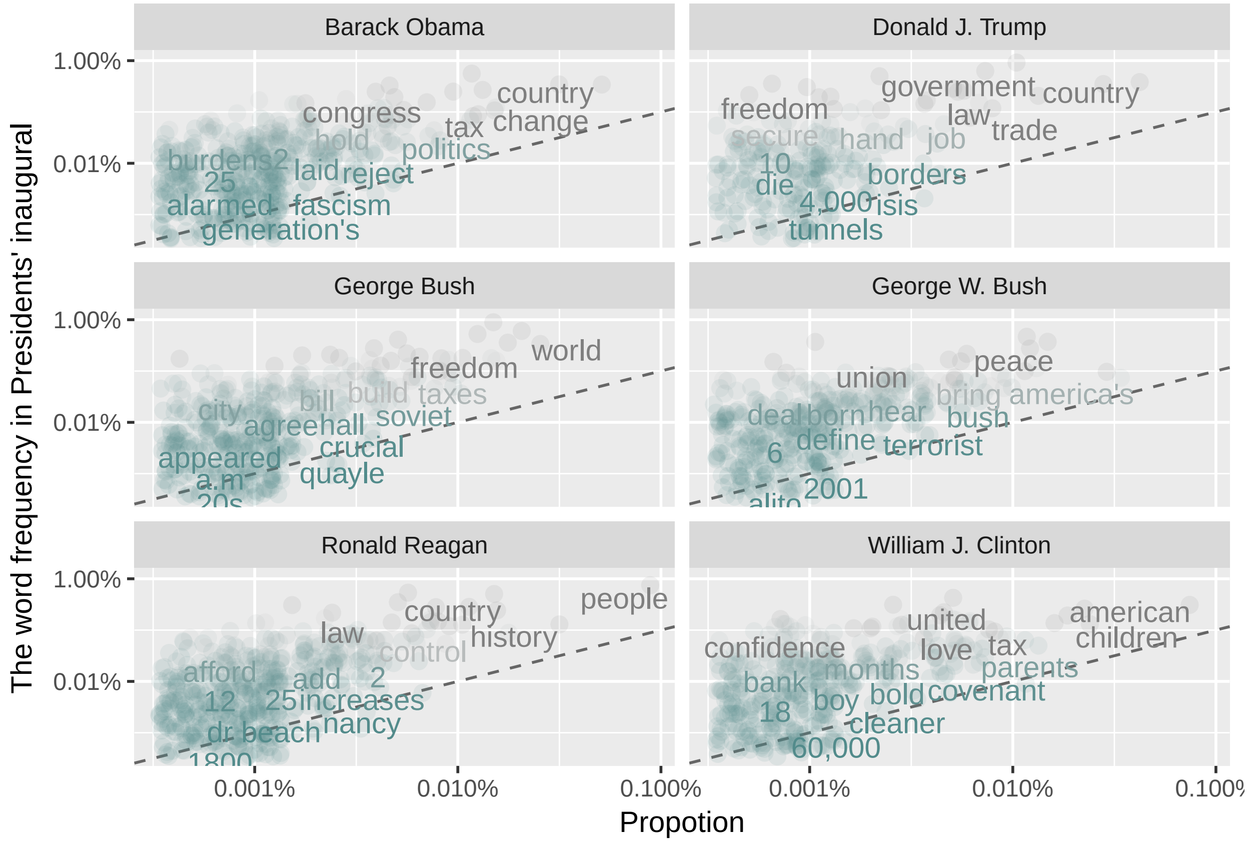
```
## Joining, by = "word"

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

```
Sub <- frequency[frequency$Name == Last_six, ]
```

```
## Warning in frequency$Name == Last_six: longer object length is not a
## multiple of shorter object length
```

```
ggplot(Sub, aes(x = Pres_words, y = all_words, color = abs(all_words - Pres_words))) +
        geom_abline(color = "gray40", lty = 2) +
        geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
        geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
        scale_x_log10(labels = scales::percent_format()) +
        scale_y_log10(labels = scales::percent_format()) +
        scale_color_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray75") +
        facet_wrap(~ Name, ncol = 2) +
        theme(legend.position="none") +
        labs(y = "The word frequency in Presidents' inaugural", x = "Propotion")
```



Words are close to the line such as "crucial", "americas", in these plots have similar frequencies across all the novels.

Words that are far from the line are words that are found more in one text than another. Furthermore, words standing out above the line are common across the entire speeches but not within a specfic one; whereas words below the line are common in that specific speech but not in all the speeches.

For example, "Freedom" stands out above the line in the Donald J. Trump. This means that "Freedom" is fairly common across the entire speeches but is not used as much in Trump's inaugural speech.

Step3. Sentiment Analysis: Sentiment changes of latest six presidents

The tidytext package contains three sentiment lexicons in the sentiments dataset. Here we will use "bing" which categorizes words in a binary fashion into positive and negative categories.

Step1. We break up each speech by 50 words

Step2. Used "bing" to assess the positive vs. negative sentiment of each word

Step3. Counted up how many positive and negative words there are for every 50 words

Step4. Calculated a net sentiment using (positive words - negative words)

Step5. Plotted the final results

```
# Loaded lexicon
get_sentiments("bing")
```

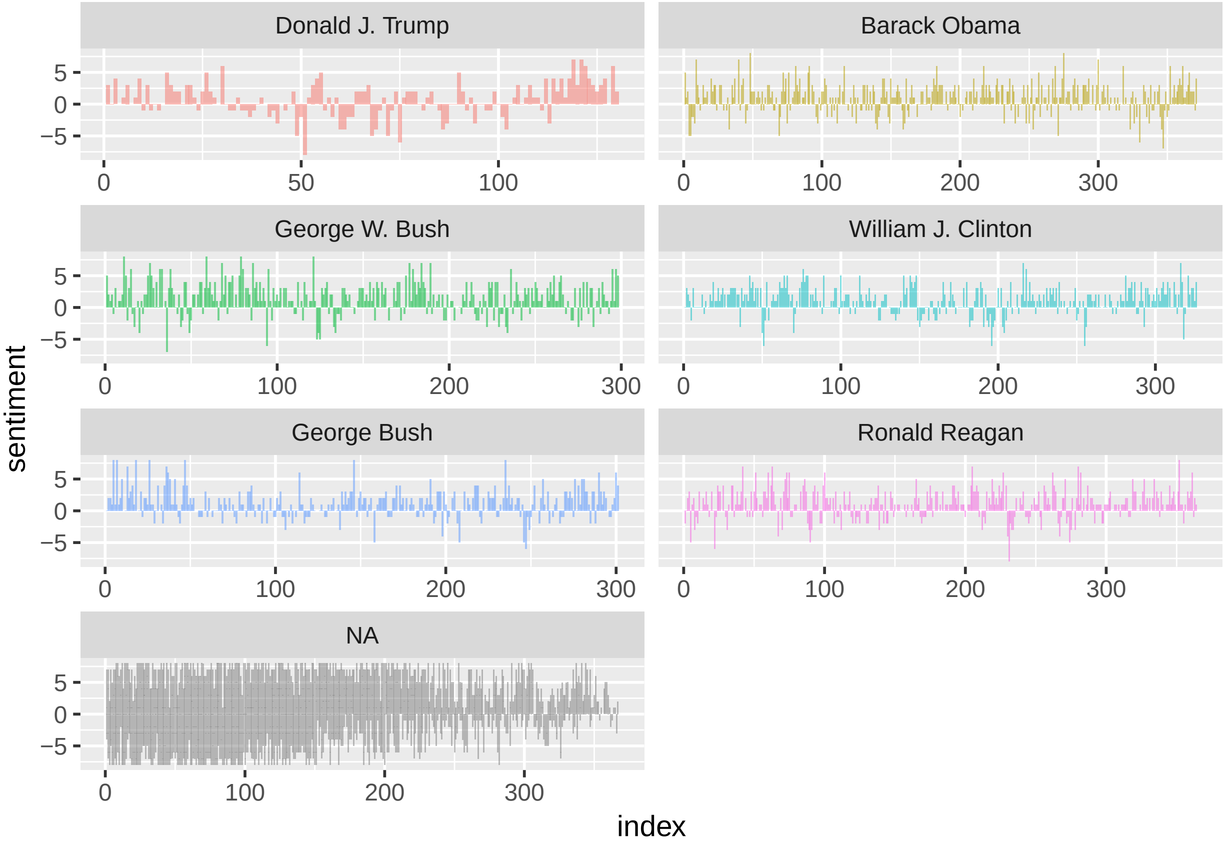
```
## # A tibble: 6,788 x 2
##           word sentiment
##          <chr>     <chr>
##   1     2-faced  negative
##   2     2-faces  negative
##   3          a+  positive
##   4    abnormal  negative
##   5     abolish  negative
##   6  abominable  negative
##   7  abominably  negative
##   8   abominate  negative
##   9 abomination  negative
##  10       abort  negative
## # ... with 6,778 more rows
```

```
series %>%
        group_by(Name) %>%
        mutate(word_count = 1:n(),
               index = word_count %/% 50 + 1) %>%
        inner_join(get_sentiments("bing")) %>%
        count(Name, index = index , sentiment) %>%
        ungroup() %>%
        spread(sentiment, n, fill = 0) %>%
        mutate(sentiment = positive - negative,
               Name = factor(Name, levels = Last_six)) %>%
        ggplot(aes(index, sentiment, fill = Name)) +
          ylim(-8,8) +
          geom_bar(alpha = 0.5, stat = "identity", show.legend = FALSE) +
          facet_wrap(~ Name, ncol = 2, scales = "free_x")
```

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

```
## Warning: Removed 27 rows containing missing values (position_stack).
```

```
## Warning: Removed 3891 rows containing missing values (geom_bar).
```



From above plots we can see that all the speeched move more positive than nagetive and Trump seemed to have a highest propotion of negative words. Also we can see clearly how the emotion of each president changed over their inaugural speach.