SOTU Speech Webscraping

Daniel Huang

28 September, 2021

Writing a function that gets and processes the SOTU speeches from the UCSB website. We extract the president, date, and body for each speech and put it into a dataframe. We then split the body of the speech into spoken and unspoken categories, keeping both saved. Interesting words and wordstems are counted and saved for any potential exploratory analysis. Some example analysis is given, but the data is in a form to be utilized by anyone wanting to explore trends in SOTU speeches over the years.

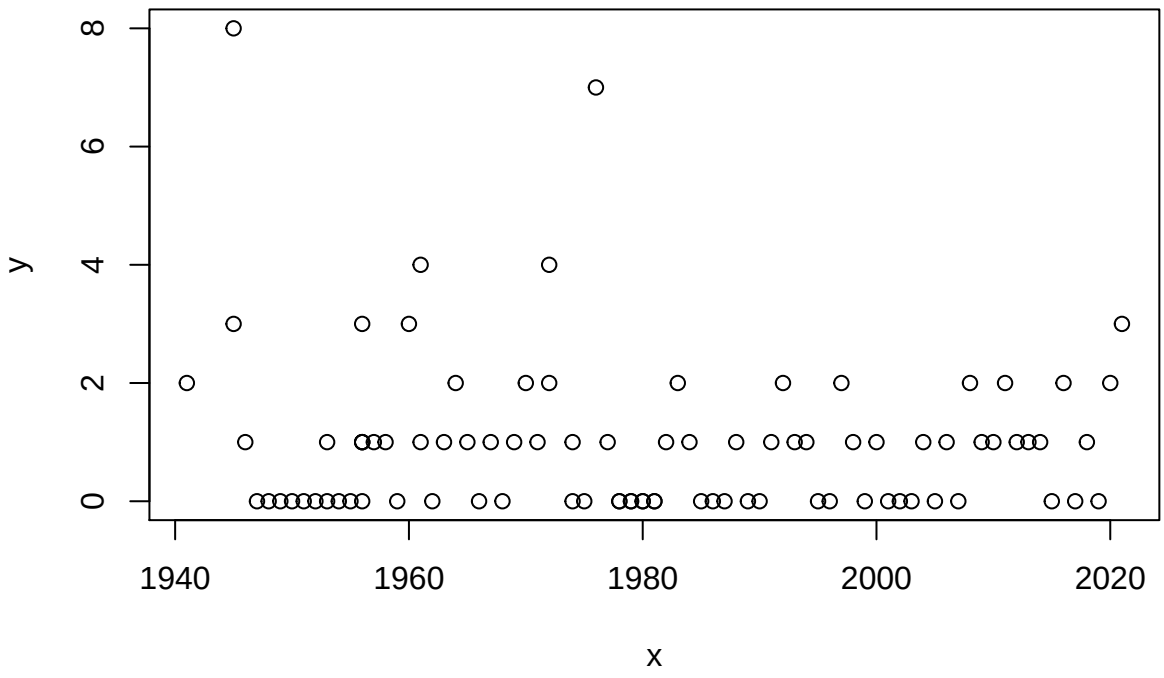
```
#Helper Functions
#Returns counts of interesting word/word stems
get_counts <- function(text, patterns) {
  counts <- lapply(patterns, function(x) {
    return(list(x,str_count(text, x)))
  })
  return(counts)
}

#Returns vectors of the words and sentences from linked text
get_words_sentences <- function(text) {
  words <- str_extract_all(text, "[[:alpha:]]+")
  sentences <- str_extract_all(text, "[^.?!]+[.?!]")
  return(list(words,sentences))
}

#Gets year and body of a speech from link
get_year_body <- function(link) {
  if (is.na(link) || link == "#nixon1973") {
    date <- NA
    body <- NA
    president <- NA
  } else {
    date <- link %>% read_html() %>% html_nodes(".date-display-single") %>% html_text()
    body <- link %>% read_html() %>% html_nodes(".field-docs-content") %>% html_text()
    president <- link %>% read_html() %>% html_nodes(".field-title") %>% html_text()
    president <- str_replace_all(president, "\n ", "")
  }
  return(list(president,date,body))
}

#Strips out things not said by the president and returns both the president's speech + others
get_split_body <- function(text) {
  body1 <- str_remove_all(text, "\\[[^ ]*\\]")
  body2 <- str_extract_all(text, "\\[[^ ]*\\]")
  return(list(body1, body2))
}
```

```r
#Analysis:
URL <- "https://www.presidency.ucsb.edu/documents/presidential-documents-archive-guidebook/annual-messag

#Get links to speeches
links <- URL %>% read_html() %>% html_nodes("td") %>% html_nodes("a") %>% html_attr("href")

#Removing the #nixon1973, nixon speeches, and first NA
links <- links[c(-1, -54, -249:-260)]

#Get year and body from links
year_body_info <- lapply(links, get_year_body)

#Storing original body and year in vectors
pres <- lapply(year_body_info, function(x) {return(x[[1]])})
year <- lapply(year_body_info, function(x) {return(x[[2]])})
orig_body <- lapply(year_body_info, function(x) {return(x[[3]])})

#Split speeches into spoken and unspoken
split_speeches <- lapply(orig_body, get_split_body)
spoken <- lapply(split_speeches, function(x) {return(x[[1]])})
unspoken <- lapply(split_speeches, function(x) {return(x[[2]][[1]])})

#Find the counts of applause and laughter
counts_applause <- unlist(lapply(unspoken, function(x)
  {return(max(table(x)[names(table(x))=="[Applause]"], 0))}), use.names = FALSE)
counts_laughter <- unlist(lapply(unspoken, function(x)
  {return(max(table(x)[names(table(x))=="[Laughter]"], 0))}), use.names = FALSE)

#Get word and sentence character vectors
words_sentences <- lapply(spoken, get_words_sentences)

words <- as.vector(lapply(words_sentences, function(x) {return(x[[1]][[1]])}))
sentences <- as.vector(lapply(words_sentences, function(x) {return(x[[2]][[1]])}))

#Get word and character counts, and average word length
word_count <- lapply(words, function(x) {return(length(x))})
char_count <- lapply(words, function(x) {return(sum(nchar(x)))})
avg_word_length <- mapply(function(words, chars)
  {return(chars[[1]]/words[[1]])}, word_count, char_count)

#Find words/wordstems of interest
stems <- c("I[ |']", "we[[:punct:]]?", "America[ n[:punct:]]",
           "democra(cy|tic)", "republic(?!an)", "Democrat([[:punct:]]|ic| )",
           "Republican", "free( |[[:punct:]]|dom)", "war", "God(?! bless)",
           "God bless", "(Jesus|Christ|Christian)", "tax")

stem_counts <- lapply(spoken, function(x) {
  return(get_counts(x, stems))
})

#Put all vectors together into a usable dataframe
speeches_info <- as.data.frame(cbind(1:length(pres), unlist(pres), unlist(year),
                                     unlist(word_count), unlist(char_count),
```

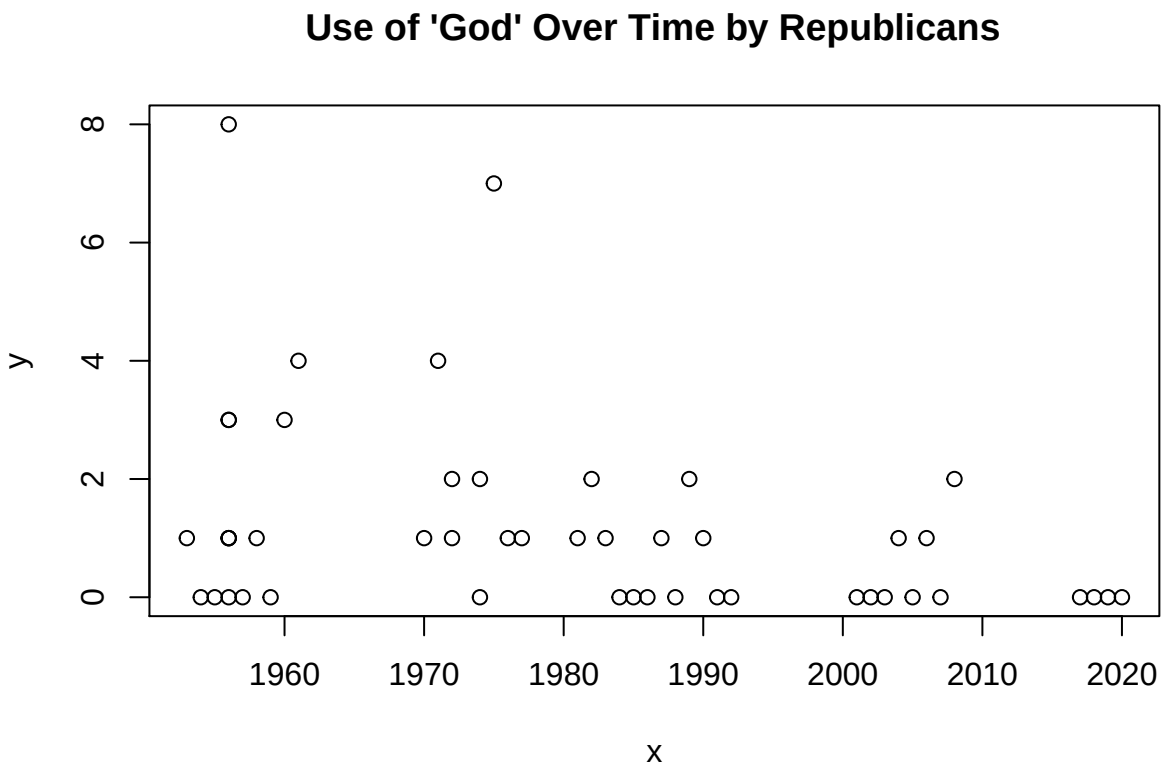
```
                                    unlist(avg_word_length), unlist(counts_laughter),
                                    unlist(counts_applause)))
colnames(speeches_info) <- c("Index", "President", "Year", "Word_Count",
                             "Character_Count", "Avg_Word_Length", "Laughter", "Applause")

#Original body, spoken parts, unspoken parts, word vector and character vector can be
#found by referencing the correct index in the corresponding vector: orig_body, spoken,
#unspoken, sentences, and words
#For example, to find the original body for Trump's 2018 speech I would see the index 3
#and use:
orig_body[3]
```

```
## [[1]]
## [1] "\n    The President. Mr. Speaker, Mr. Vice President, Members of Congress, the First Lady of the
```

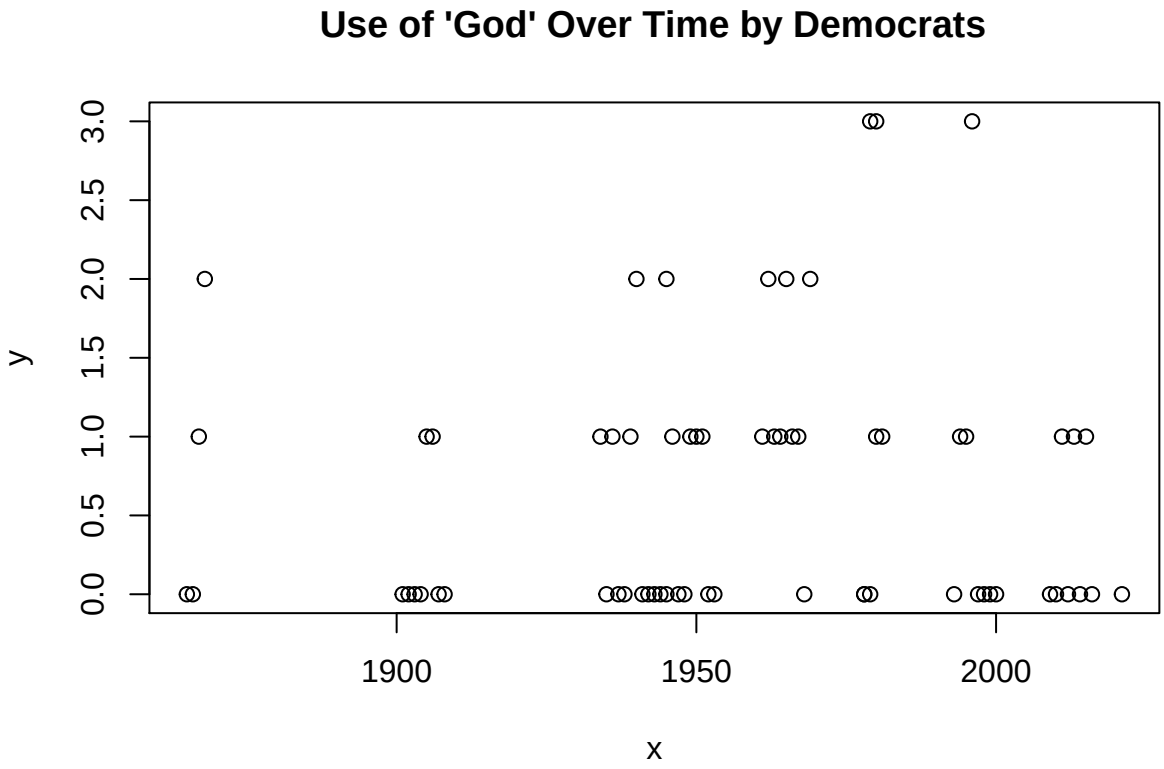
```
#Stem_counts contains information regarding the frequency of various stems. Stem_counts[[i]][[j]]
#returns the corresponding wordstem and frequency of the jth wordstem in the ith speech. i is the
#same index as in speeches_info, and j is the corresponding index in stems.

#Basic exploratory analysis plots:
#Gets counts from stem_counts given the index for a wordstem
get_frequencies <- function(index, counts) {
  result <- lapply(counts, function(x) {
    return(x[[index]][[2]])
  })
}

#Makes plot given the speech_info style dataframe, index for desired wordstem,
#and the stem_counts style list
make_plot <- function(df, index, counts, ...) {
  y <- rev(unlist(get_frequencies(index, counts[as.numeric(df$Index)])[1:nrow(df)]))
  x <- unlist(lapply(df$Year, function(x) {
    return(str_extract(x, "[[:digit:]]{4}"))
  }))
  plot(x, y, ...)
}
#Looking at use of the word "God" over time since 1932:
make_plot(slice(speeches_info, 1:90), 10, stem_counts)
```




```r
#Filtering into Republican and Democrat data
repub <- filter(speeches_info,
      str_detect(speeches_info$President, "(Eisenhower|Nixon|Ford|Reagan|Bush|Trump)")
)
democrat <- filter(speeches_info,
      str_detect(speeches_info$President,
                 "(Roosevelt|Truman|Kennedy|Johnson|Carter|Clinton|Obama| Biden)")
)

#Looking at use of word "God" over time for Republicans vs Democrats
make_plot(repub, 10, stem_counts, main="Use of 'God' Over Time by Republicans")
```




```
make_plot(democrat, 10, stem_counts, main="Use of 'God' Over Time by Democrats")
```




```
#Using make_plot, we could do more analysis for any wordstem we are interested in, and we can
#subset our speeches_info dataframe differently to get different groupings
```