Web Scraping Analysis on Presidents Speeches

Yiyun Gong

Part 1

The website The American Presidency Project at UCSB has the text from all of the State of the Union speeches by US presidents. (These are the annual speeches in which the president speaks to Congress to "report" on the situation in the country.)

Tasks:

- Extract the url for each speech
- Extract the year of the speech and boday of the speech
- Stripped out the text that was not spoken by president and saved seperately
- Extract words and sentences
- Count the numbers of words and characters
- Count words frequency
- Store results as well-structured data objects
- Visualizations
- Quantify speeches and illustrates how speeches have changed over time

Website: "https://www.presidency.ucsb.edu/documents/presidential-documents-archive-guidebook/annual-messages-congress-the-state-the-union"

```
base = "https://www.presidency.ucsb.edu/documents/presidential-documents-archive-guidebook/annual-messag
search = read_html(base)
```

```
speech_table =
  search %>%
    html_elements('table') %>%
    html_nodes("a") %>%
    html_attr("href") %>%
    # remove NA
    na.omit() %>%
    # remove the duplicated link
    unique() %>%
    data.frame()
```

```
names(speech_table) = "link"
# get rid of lines that do not contain legit url like "#nixon1973"
speech_table = data.frame(speech_table[grepl("http", speech_table$link),])
names(speech_table) = "link"
```

```r
speech_processor = function(link){
  res = read_html(link)
  # Extract the President name
  President =
    res %>%
    html_elements(".diet-title")%>%
    html_text()

  # Extract the body of the speech
  speech_body =
    res %>%
    html_elements(".field-docs-content") %>%
    html_text()

  # Extract the year of the speech
  year =
    res %>%
    html_elements(".date-display-single") %>%
    html_text() %>%
    str_sub(start= -4)

  # save all text that was not spoken by president to speech_strip
  speech_strip =
    speech_body %>%
    str_extract_all("\\[[^\\]]*\\]")

  # remove all text that was not spoken by president from speech_body
  speech_body =
    speech_body %>%
    str_remove_all("\\[[^\\]]*\\]")

  # count the number of times [lL]aughter and [aA]pplause occurrences in the speech
  speech_strip_str = paste(unlist(speech_strip),collapse=" ")
  laughter = str_count(speech_strip_str, "[lL]aughter")
  applause = str_count(speech_strip_str, "[aA]pplause")

  # extract sentences from each speech as character vectors
  sent_vector =
    speech_body %>%
      gsub("\\.\\s", "\\.\t", .) %>%
      # trim the leading whitespace
      trimws() %>%
      strsplit("\\t")

  # extract words from each speech as character vectors
  word_vector =
    speech_body %>%
      gsub("\\.\\s", "\\.\t", .) %>%
      # special cases: remove all the apostrophe/commas(single word or number seperator)
      str_remove_all("'") %>%
      str_remove_all(",") %>%
      gsub("[[:punct:][:blank:]]+", " ", .)%>%
      trimws() %>%
      str_split(" ")

  # count the number of words
  word_count = str_count(speech_body, '\\w+')
  # count the number of characters
  char_count = nchar(speech_body)
  # compute the average word length
  avg_word_length = char_count / word_count

  # Count the occurrence of I
  I_count = str_count(speech_body, "\\bI\\b")
  # Count the occurrence of we
  we_count = str_count(speech_body, "\\bwe\\b")
  # Count the occurrence of America{,n}
  America_count = str_count(speech_body, "\\bAmerica(n)?\\b")
  # Count the occurrence of democra{cy,tic}
  democracy_count = str_count(speech_body, "\\bdemocra(cy|tic)\\b")
  # Count the occurrence of republic
  republic_count = str_count(speech_body, "\\brepublic\\b")
  # Count the occurrence of Democrat{,ic}
  Democrat_count = str_count(speech_body, "\\bDemocrat(ic)?\\b")
  # Count the occurrence of Republican
  Republican_count = str_count(speech_body, "\\bRepublican\\b")
  # Count the occurrence of free{,dom}
  free_count = str_count(speech_body, "\\bfree(dom)?\\b")
  # Count the occurrence of war
  war_count = str_count(speech_body, "\\bwar\\b")
  # Count the occurrence of God (not including God bless)
  god_count = str_count(speech_body, "\\bGod\\b(?!.*\\bbless\\b)")
  # Count the occurrence of God Bless
  gb_count = str_count(speech_body, "\\bGod [Bb]less\\b")
  # Count the occurrence of {Jesus, Christ, Christian}
  jesus_count = str_count(speech_body, "\\b(Jesus|Christ|Christian)\\b")
  # Additional Word count for 1i to quantify speech
  # Count the occurrence of independen{t,ce}
  indep_count = str_count(speech_body, "\\b[iI]ndependen(t|ce)\\b")
  # Count the occurrence of nation{al}
  nation_count = str_count(speech_body, "\\b[nN]ation(al)?\\b")

  # Organize the results into a table
  result = data.table(
    President = President,
    speech_body = speech_body,
    year = year,
    speech_strip = speech_strip,
    laughter = laughter,
    applause = applause,
    sent_vector = sent_vector,
    word_vector = word_vector,
    word_count = word_count,
    char_count = char_count,
    avg_word_length = avg_word_length,
    I_count = I_count,
    we_count = we_count,
    America_count = America_count,
    democracy_count = democracy_count,
    republic_count = republic_count,
    Democrat_count = Democrat_count,
    Republican_count = Republican_count,
    free_count = free_count,
    war_count = war_count,
    god_count = god_count,
    gb_count = gb_count,
    jesus_count = jesus_count,
    indep_count = indep_count,
    nation_count = nation_count
  )
  return(result)
}
```

```
# Organize the speech text analysis results into a table
n = nrow(speech_table)
speech_result = data.table()
for (i in 1:n){
  speech = speech_processor(speech_table$link[i])
  speech_result = rbind(speech_result, speech)
}
```

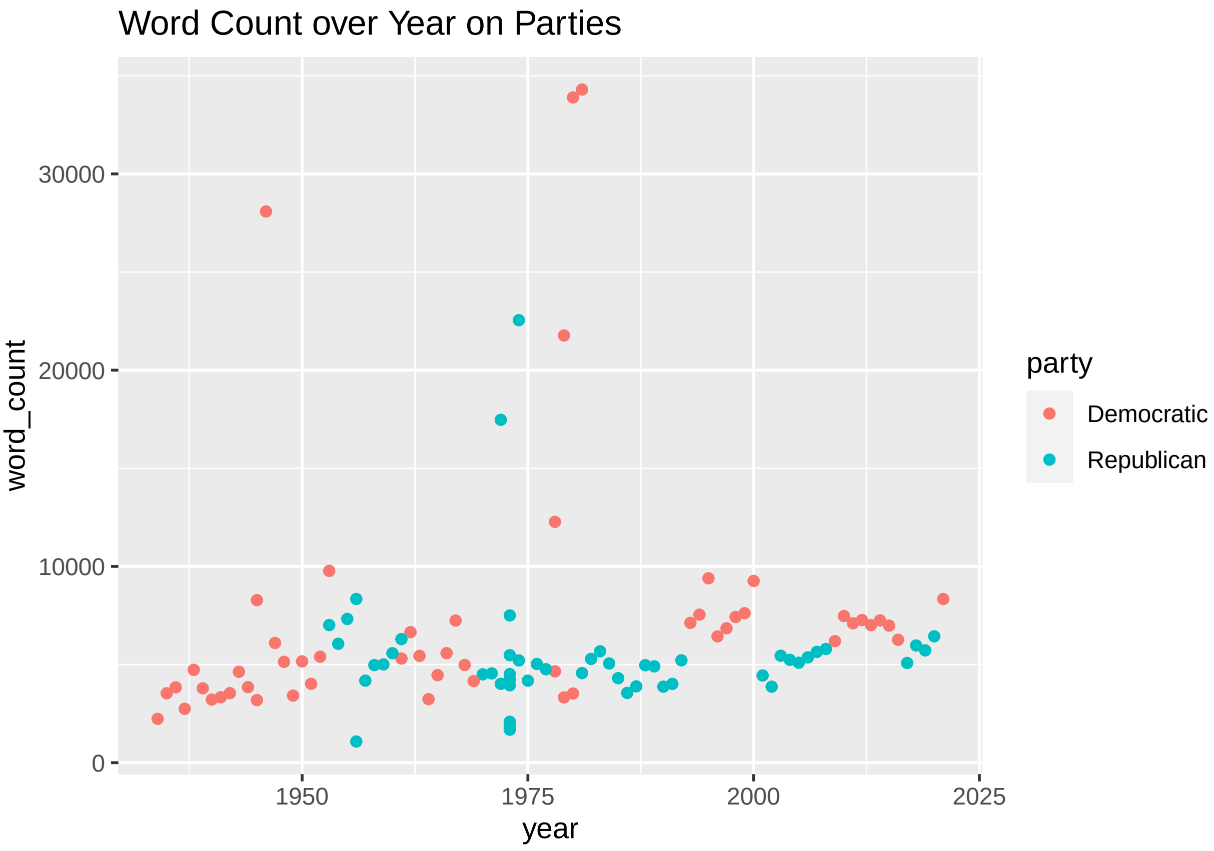
```
# Define Parties of Presidents
Rep_pres = c("Dwight D. Eisenhower", "Richard Nixon", "Gerald R. Ford",
             "Ronald Reagan", "George W. Bush", "George Bush", "Donald J. Trump")
Dem_pres = c("Franklin D. Roosevelt", "Harry S. Truman", "John F. Kennedy",
             "Lyndon B. Johnson", "Jimmy Carter", "William J. Clinton",
             "Barack Obama", "Joseph R. Biden")
speech_result$party = ifelse(speech_result$President %in% Rep_pres, 'Republican',
               ifelse(speech_result$President %in% Dem_pres, 'Democratic', 'Unknown'))
```

```
# Keep a copy of the results
saveRDS(speech_result, file="speech_outcome.RDS")
#speech_result <- readRDS("speech_outcome.RDS")
```

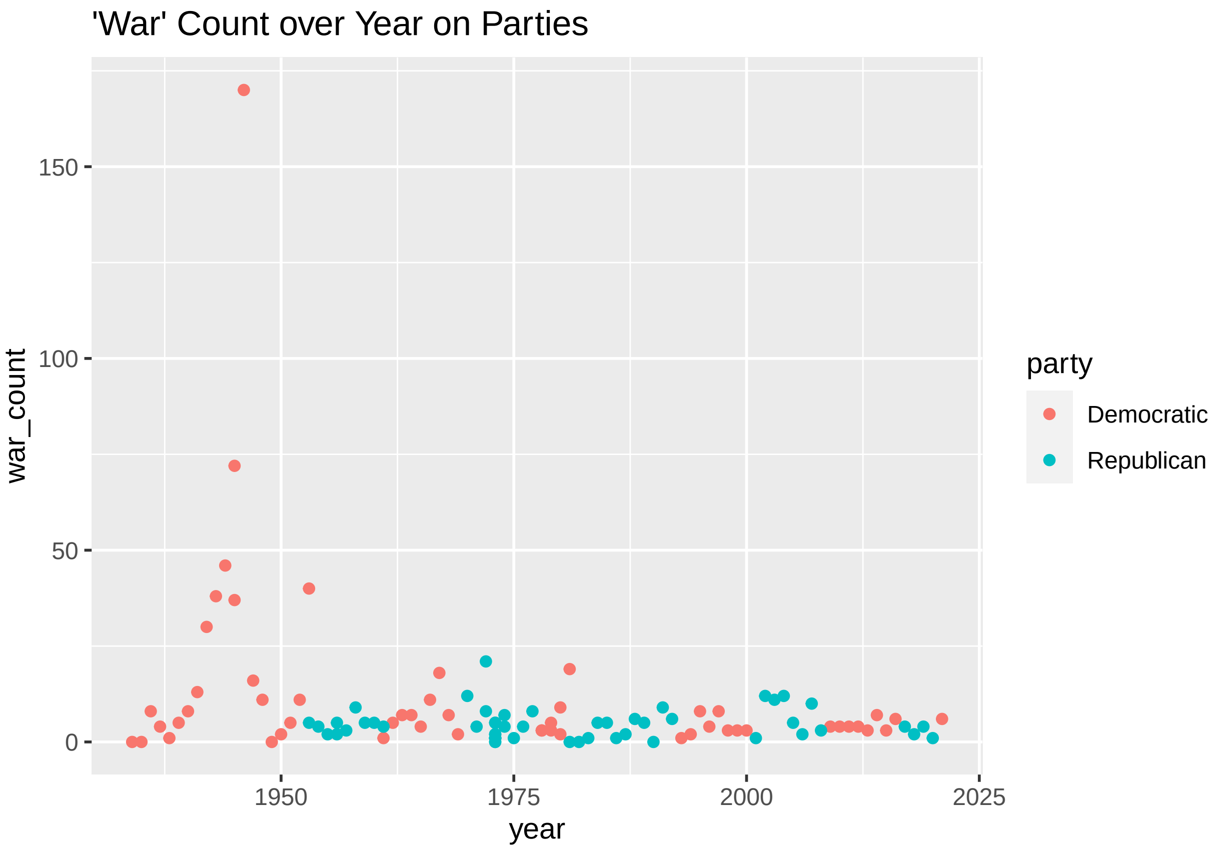
```
# Drop some unused columns
graph_table = speech_result[,c(1,3,5,6,9:26)]
# Subset the table for Presidents since Franklin Roosevelt in 1932
graph_table_party = graph_table[which(graph_table$party != 'Unknown'),]
head(graph_table_party)
```

```
##          President year laughter applause word_count char_count avg_word_length
## 1: Joseph R. Biden 2021        4        1       8341      45718        5.481117
## 2: Donald J. Trump 2017        4        3       5082      29037        5.713695
## 3: Donald J. Trump 2018        6        6       5973      33921        5.679056
## 4: Donald J. Trump 2019        8        8       5724      32898        5.747379
## 5: Donald J. Trump 2020        2        4       6439      37546        5.831030
## 6:    Barack Obama 2013        1        4       7008      39407        5.623145
##    I_count we_count America_count democracy_count republic_count Democrat_count
## 1:      129       123             109              17              0              5
## 2:       41        73              62               0              0              0
## 3:       39       100              62               0              0              0
## 4:       44        81              59               0              1              1
## 5:       60        68              74               1              1              1
## 6:       47       110              47               4              0              0
##    Republican_count free_count war_count god_count gb_count jesus_count
## 1:                4          2         6         2        1           0
## 2:                2          8         4         1        2           0
## 3:                0          7         2         3        1           0
## 4:                1          7         4         3        2           0
## 5:                2         11         1         8        2           0
## 6:                0          9         3         1        2           0
##    indep_count nation_count      party
## 1:           1           21 Democratic
## 2:           1           23 Republican
## 3:           1           10 Republican
## 4:           1           13 Republican
## 5:           2           19 Republican
## 6:           2            8 Democratic
```

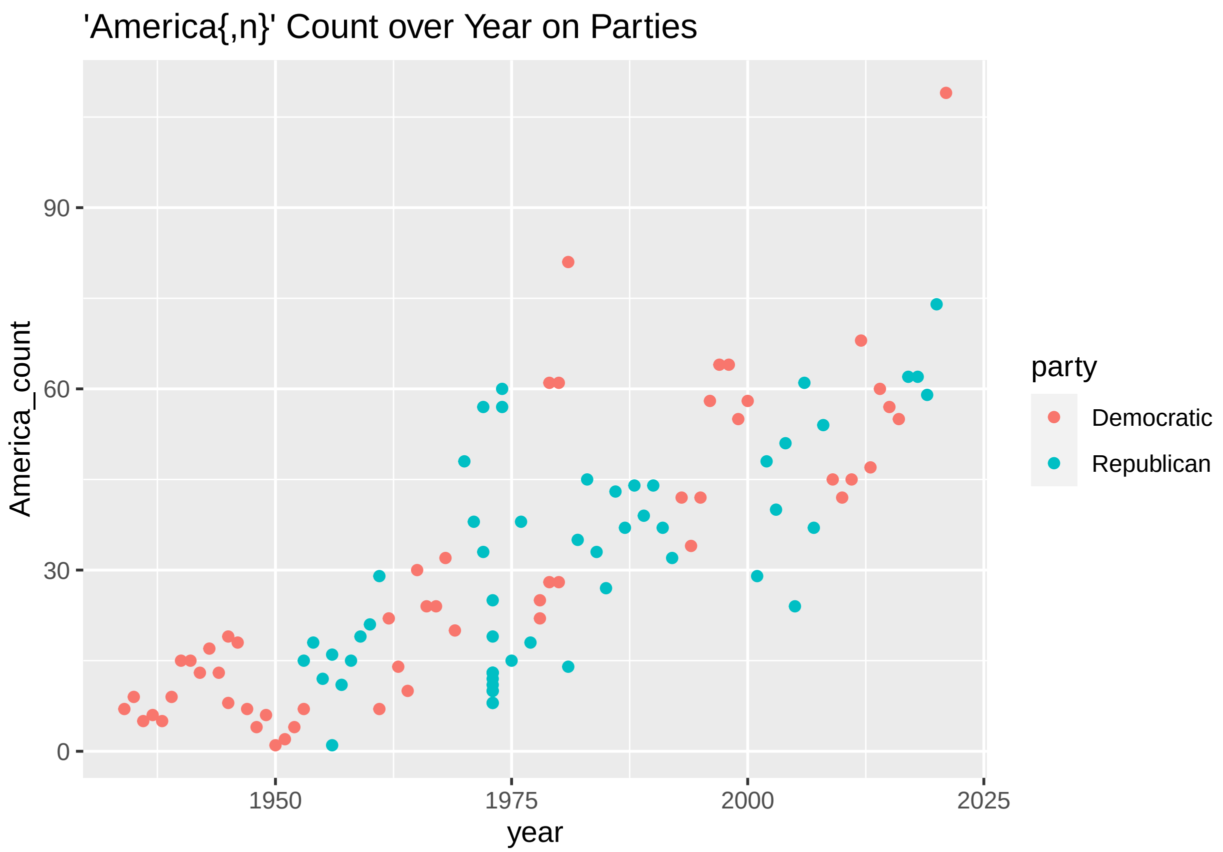
```
graph_table_party$year = as.numeric(graph_table_party$year)
ggplot(graph_table_party, aes(x=year, y=word_count,col=party))+
  geom_point()+ggtitle("Word Count over Year on Parties")
```



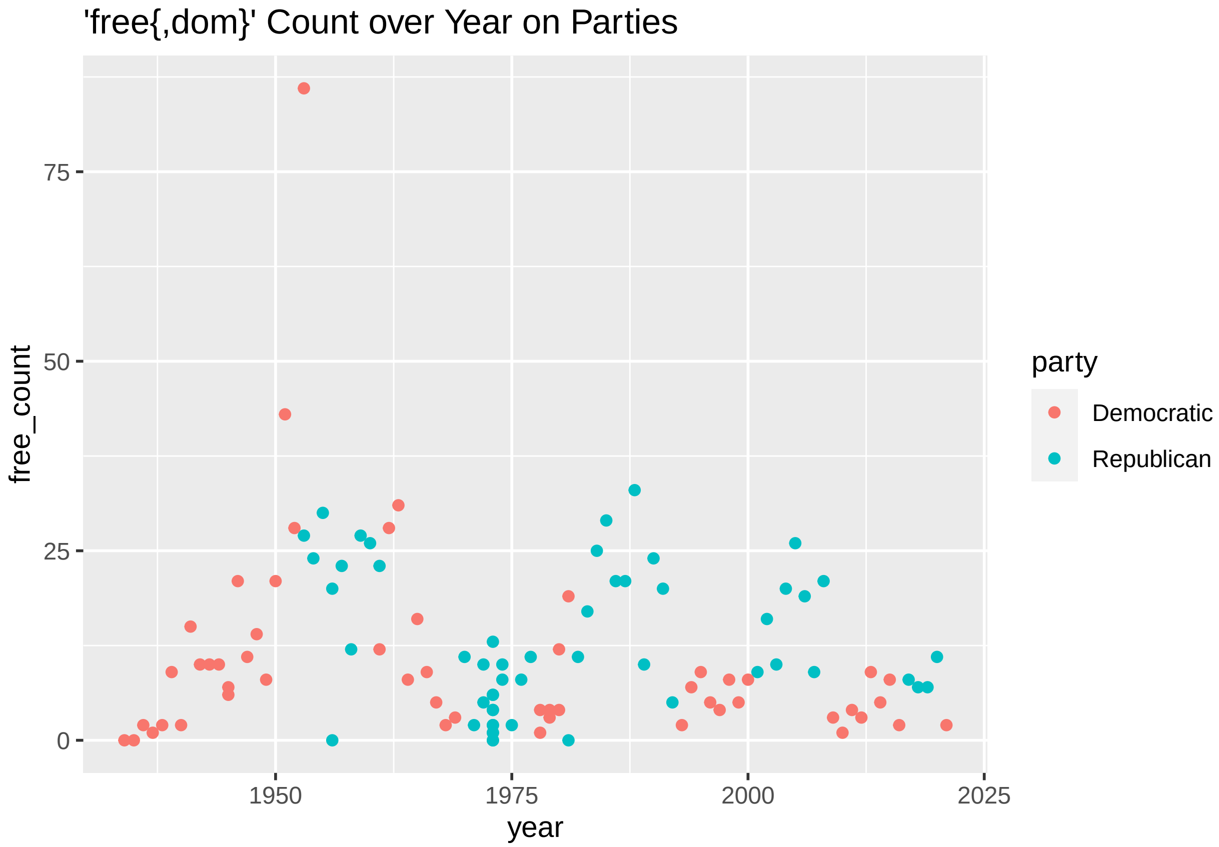
```
ggplot(graph_table_party, aes(x=year, y=war_count, col=party))+
  geom_point()+ggtitle("'War' Count over Year on Parties")
```



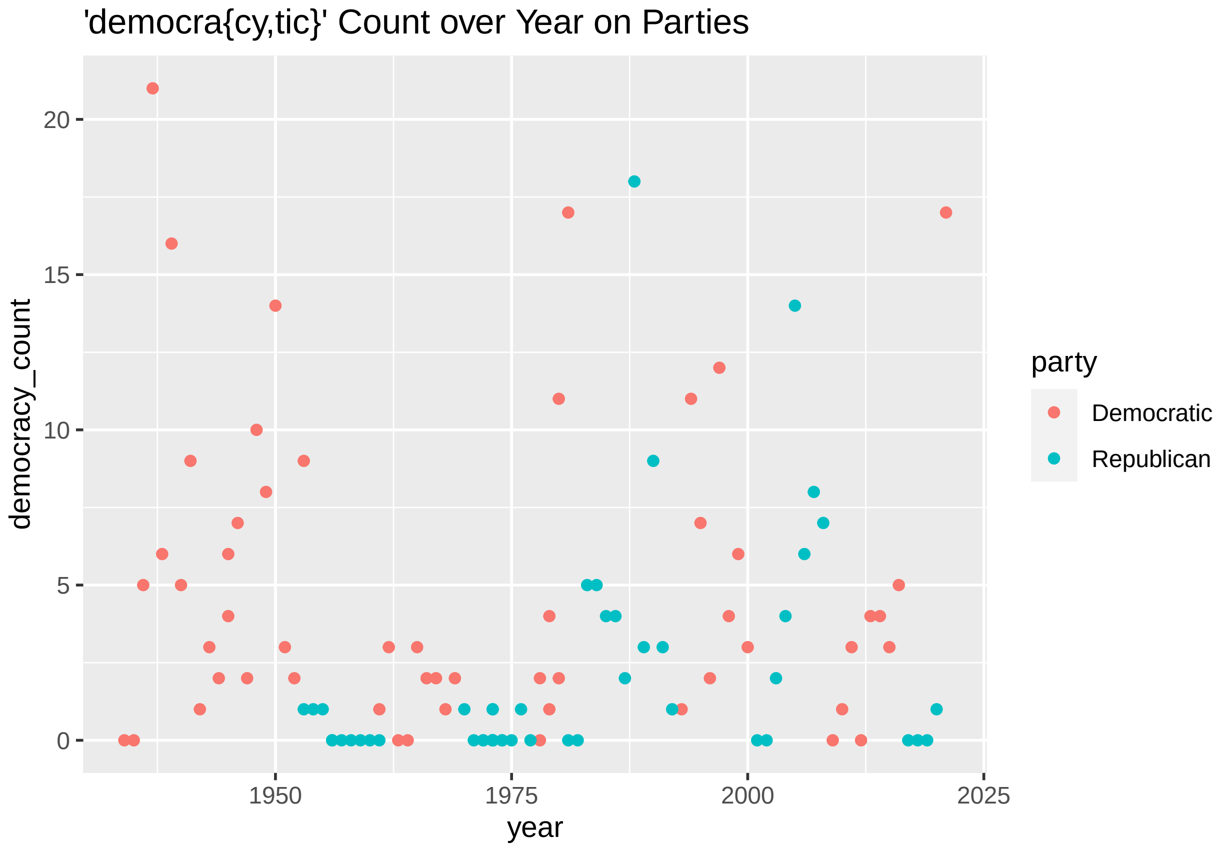
```
ggplot(graph_table_party, aes(x=year, y=America_count, col=party))+
  geom_point()+ggtitle("'America{,n}' Count over Year on Parties")
```



```
ggplot(graph_table_party, aes(x=year, y=free_count, col=party))+
  geom_point()+ggtitle("'free{,dom}' Count over Year on Parties")
```



```
ggplot(graph_table_party, aes(x=year, y=democracy_count, col=party))+
  geom_point()+ggtitle("'democra{cy,tic}' Count over Year on Parties")
```



From the plots

- we can see that word count is evenly distributed between Democratic and Republican parties over the years. There are a few high points around 1975 for both parties.
- For the word count of "War". We can see that the Democratic party use the word of "War" significantly higher between year 1940-1955.
- For the word count of "America{,n}". We can see that the use of this word was increasing over the years for both parties. It also reached a high peak at the year of 2021.
- For the word count of "free{,dom}". We can see that the use of it had a few high peaks at around 1960, 1990, 2010. For all three peaks, the presidents were Republican. In comparison, Democratic presidents used the word of "freedom" less. However, the highest peak was at around 1952 and presented by a Democratic president.
- For the word count of "democra{cy,tic}". We can see that the use of it is evenly distributed between both parties.

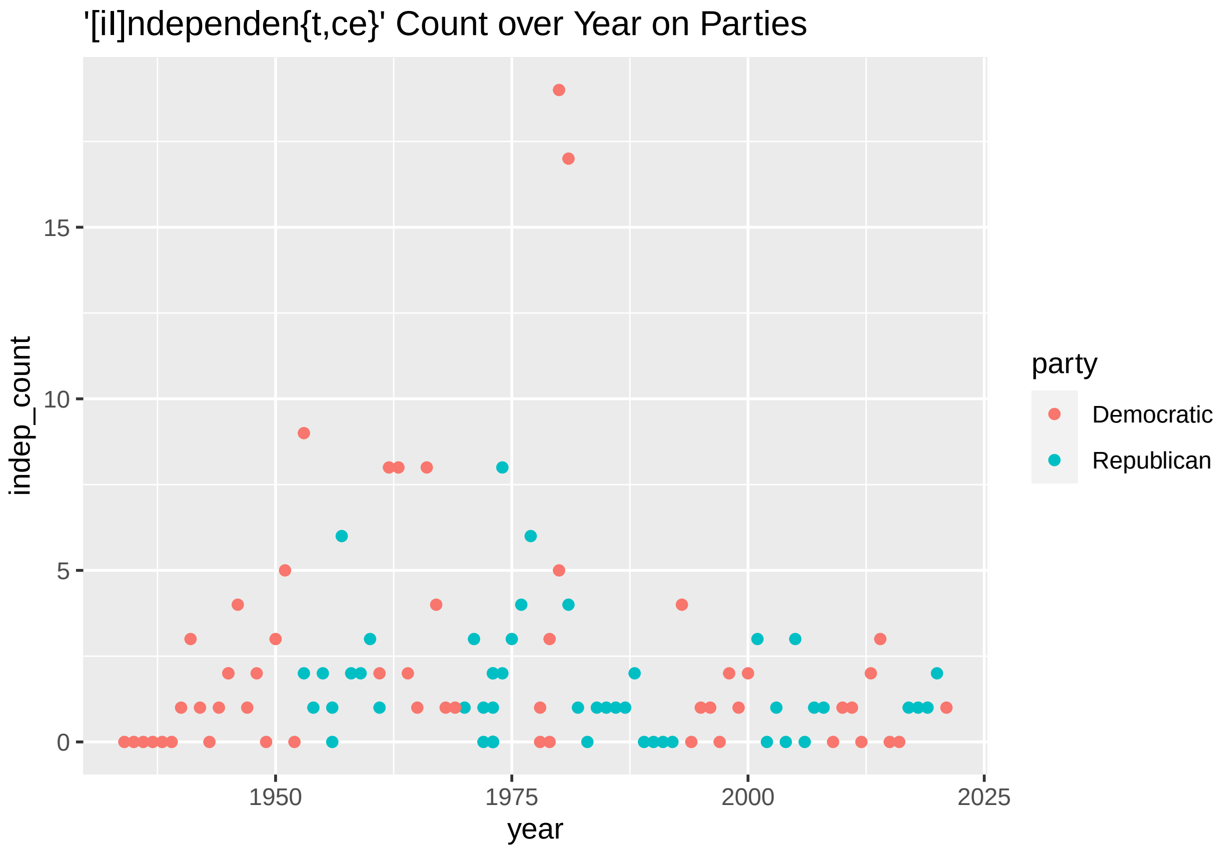
Additional Speech Quantify

I did additional word counts on *Independen{t,ce}* and *Nation{,al}* because I noticed these two words have high frequency of occurrences in many speeches and are very meaningful words. These two variables might be interesting variables that quantify speeches.

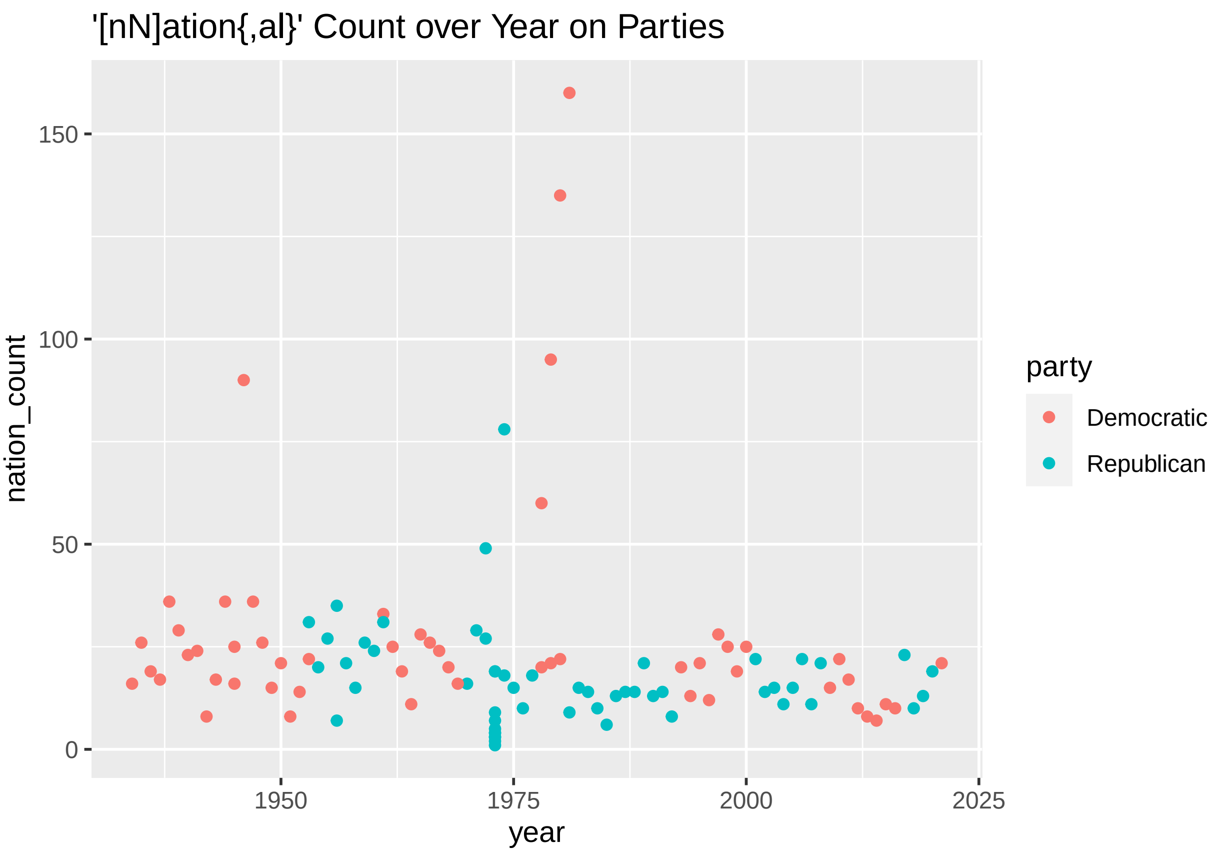
From the following two plots:

- The word "Independent/Independence"are evenly distributed between two parties.
- The word "Nation/National" appears in a high frequency across most of the speeches over the years. It reached a peak around 1975. During the peak, the Democratic president used it more often than the Republican president.

```
ggplot(graph_table_party, aes(x=year, y=indep_count, col=party))+
  geom_point()+ggtitle("'[iI]ndependen{t,ce}' Count over Year on Parties")
```



```
ggplot(graph_table_party, aes(x=year, y=nation_count, col=party))+
  geom_point()+ggtitle("'[nN]ation{,al}' Count over Year on Parties")
```



Part 2

Design an object-oriented programming (OOP) approach to the text analysis in Part 1.

In the context of R6

speechClass works as parent class It contains elements that retrieved directly from the html. Functions are public

speechClass:

- initialize variables self$speech_html with html object
- function president: to get the president's name from speech_html
- function speech_body: to retrieve the speech body from speech_html
- function speech_body_clean: to retrieve the cleaned-up version of speech body (removed parts that were not spoken by the president)
- function year: get the year from speech_html

stripClass works as subclass of speechClass. It serves the purpose of save the stripped parts of speech (parts that were not spoken by the president)

stripClass:

- initialize as subclass of speechClass - inherited from speechClass
- function speech_strip: get the parts of speech that were not spoken by the president

laughterClass works as subclass of stripClass. It serves the purpose of count the number of times that Laughter and Applause occur from the speech_strip variable in stripClass

laughterClass:

- initialize as subclass of stripClass – inherited from stripClass
- function laughter: get the count of Laughter case
- function applause: get the count of applause case

countClass works as a subclass of speechClass. It serves the purpose of extracting words and sentences as character vectors and counting the occurrences of key words

countClass:

- initialize as subclass of speechClass – inherited from speechClass
- function sent_vector: extracting sentences as character vector
- function word_vector: extracting words as character vector
- function avg_word_length: computing the average word length
- A series of functions to count key words: I_count, we_count, America_count, …, indep_count, nation_count

Below is a demo of my R6 class approach to the text analysis with a few test cases.

```r
speechClass <- R6Class("speechClass",
  public = list(
    speech_html = NULL,
    initialize = function(speech_html = NA) {
      self$speech_html <- speech_html
    },
    president = function(){
      self$speech_html %>%
      html_elements(".diet-title")%>%
      html_text()
    },
    speech_body = function(){
      self$speech_html %>%
      html_elements(".field-docs-content") %>%
      html_text()
    },
    speech_body_clean = function(){
      self$speech_html %>%
      html_elements(".field-docs-content") %>%
      html_text() %>%
      str_remove_all("\\[[^\\]]*\\]")
    },
    year = function(){
      self$speech_html %>%
      html_elements(".date-display-single") %>%
      html_text() %>%
      str_sub(start= -4)
    }
  )
)
```

```
stripClass <- R6Class("stripClass",
  inherit = speechClass,
  public = list(
    speech_strip = function() {
      speech_strip_res =
      speech_body = super$speech_body()
      speech_body %>%
      str_extract_all("\\[[^\\]]*\\]")
    }
  )
)

laughterClass <- R6Class("laughterClass",
  inherit = stripClass,
  public = list(
    laughter = function() {
      speech_strip = super$speech_strip()
      speech_strip = paste(unlist(speech_strip),collapse=" ")
      str_count(speech_strip, "[lL]aughter")
    },
    applause = function() {
      speech_strip = super$speech_strip()
      speech_strip = paste(unlist(speech_strip),collapse=" ")
      str_count(speech_strip, "[aA]pplause")
    }
  )
)

countClass <- R6Class("countClass",
  inherit = speechClass,
  public = list(
  sent_vector = function() {
    speech_body = super$speech_body_clean()
    speech_body %>%
      gsub("\\.\\s", "\\.\t", .) %>%
      trimws() %>%
      strsplit("\\t")
  },
  word_vector = function() {
    speech_body = super$speech_body_clean()
    speech_body %>%
      gsub("\\.\\s", "\\.\t", .) %>%
      str_remove_all("'") %>%
      str_remove_all(",") %>%
      gsub("[[:punct:][:blank:]]+", " ", .)%>%
      trimws() %>%
      str_split(" ")
  },
  avg_word_length = function(){
    speech_body = super$speech_body_clean()
    word_count = str_count(speech_body, '\\w+')
    # count the number of characters
    char_count = nchar(speech_body)
    # compute the average word length
    char_count / word_count
  }
  ##### Similar for the rest of the word count functions for key words etc ####
  )
)
```

```
object = read_html(speech_table$link[1])
test_1 = stripClass$new(object)
test_2 = laughterClass$new(object)
test_3 = countClass$new(object)
```

```
test_1$speech_strip()
```

```
## [[1]]
## [1] "[Laughter]" "[Applause]" "[laughter]" "[under]"    "[child]"
## [6] "[Laughter]" "[Laughter]"
```

```
test_2$applause()
```

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

```
test_3$word_vector()[[1]][1:10]
```

```
##  [1] "The"       "President" "Thank"     "you"       "Thank"     "you"
##  [7] "Thank"     "you"       "Good"      "to"
```