ANALYSIS OF THE STATE OF THE UNION LANGUAGE AND EFFECT ON CONGRESSIONAL SEATS

Data 607 Spring 2025

Maxfield Raynolds

A LANGUAGE ANALYSIS OF THE STATE OF THE UNION

Single most important political speech in the US each year (arguably)

Televised since 1947

Frequently projects the Executive Branch's (and their party's) agenda for the coming year.

Does the language of the State of the Union have a lasting political impact on the legislative branch's composition?

DOES THE LANGUAGE OF THE STATE OF THE UNION HAVE A LASTING POLITICAL IMPACT ON THE LEGISLATIVE BRANCH'S COMPOSITION?

How to examine this question?

Acquire the text of the speeches & subsequent changes in congressional makeup

Perform language analysis: Sentiment analysis

Analyze results against changes in congressional makeup following the next election

DATA ACQUISITION

Import Congressional Seat Changes

The following code imports the changes in congress house seats by year from The Brookings Institute.

```
cong_raw <- read_csv("https://www.brookings.edu/wp-
content/uploads/2024/11/2-3.csv", show_col_types = FALSE)
```

CONGRESSIONAL DATA SORTED, FILTERED AND CLEANED

The congressional data is then sorted, filtered, cleaned a little, and placed into tidy format.

```
cong_tidy <- cong_raw |>
  filter(
    Year >= 1947,
    ElectionType == "General",
  ) |>
  select(!NumSpecialElections) |>
  rename(gainingparty = GaingingParty) |>
  clean_names()
```

IMPORT LIST OF PRESIDENTS

A list of Presidents is imported into the work environment from the OpenIntro.

This data is appended with the two most recent presidents and is then cleaned, filtered and sorted into a more functionally usable tidy format.

```
## # A tibble: 10 × 4
##     year potus                  party      party_abbrev
##    <dbl> <chr>                  <chr>      <chr>
##  1  1945 Harry S. Truman        Democratic D
##  2  1946 Harry S. Truman        Democratic D
##  3  1947 Harry S. Truman        Democratic D
##  4  1948 Harry S. Truman        Democratic D
##  5  1949 Harry S. Truman        Democratic D
##  6  1950 Harry S. Truman        Democratic D
##  7  1951 Harry S. Truman        Democratic D
##  8  1952 Harry S. Truman        Democratic D
##  9  1953 Dwight David Eisenhower Republican R
## 10  1954 Dwight David Eisenhower Republican R
```

A FALSE START: SPEECH DATA

Searched and obtained some speech data through govinfo.gov api

API acquistion successful but:

Speeches labelled and notated in many different ways

- Made data acquisition very granular

API DATA ACQUISITION

```
govinfo_apikey <- key_get("govinfo.gov")

query <- "PRESIDENTIAL ADDRESS BEFORE A JOINT SESSION OF CONGRESS"
collection <- "PPP"

govinfo_url <- "https://api.govinfo.gov/search"

header <- add_headers(
  `X-Api-Key` = govinfo_apikey,
  `Content-Type` = "application/json",
  `Accept` = "application/json"
)

body <- list(
  query = query,
  pageSize = 1000,
  offsetMark = "*",
  sorts = list(list(
    field = "relevancy",
    sortOrder = "DESC"
  )),
  historical = TRUE,
  resultLevel = "default"
)

search <- POST(govinfo_url,
               header,
               encode = "raw",
               body = toJSON(body, auto_unbox = TRUE)
)

content_json <- content(search, as = "text", encoding = "UTF-8")
results <- fromJSON(content_json)

data <- results$results

data1 <- data |> filter(dateIssued >= 1947-01-01,
                        str_detect(title, regex("joint session", ignore_case = TRUE))) |> arrange(dateIssued)

links <- as.data.frame(results$results$download$txtLink)
first_url <- links$`results$results$download$txtLink`[1]

detail_url <- paste0(first_url, "?api_key=", govinfo_apikey)

res_detail <- GET(detail_url)

detail_content <- read_html(content(res_detail, as = "text"))

html_speech <- detail_content |>
  html_elements("pre") |>
  html_text()
```

A BETTER SOURCE

Web Scrape: The American Presidency Project

Performed two scrapes for URLs that linked to the text of speeches while also extracting their dates and the President that delivered them.

Cleaned, prepped and compiled them into a single dataframe

WEB SCRAPE CODE

```
speech_urls_html <- read_html("https://www.presidency.ucsb.edu/advanced-search?field-keywords=%22Address%20Before%20A%20Joint%22&field-keywords2=&field-keywords3=&from%5Bdate%5D=&to%5Bdate%5D=&person2=&category2%5B0%5D=406&category2%5B1%5D=8&category2%5B2%5D=45&items_per_page=100&order=field_docs_start_date_time_value&sort=desc")

date_node <- html_elements(speech_urls_html, ".views-field-field-docs-start-date-time-value")
president_node <- html_elements(speech_urls_html, ".views-field-field-docs-person")
url_node <- html_elements(speech_urls_html, ".views-field-title a")

date_txt <- (xml_text(date_node, trim = TRUE))[-1]
pres_txt <- (xml_text(president_node, trim = TRUE))[-1]
url_txt <- html_attr(url_node, "href")

url_table <- tibble(
  date = date_txt,
  pres = pres_txt,
  url = url_txt
)
```

WEB SCRAPE CODE 2

```
sou_urls_html <- read_html("https://www.presidency.ucsb.edu/documents/app-
categories/spoken-addresses-and-remarks/presidential/state-the-union-
addresses?items_per_page=100")

date_node <- html_elements(sou_urls_html, ".date-display-single")
president_node <- html_elements(sou_urls_html, ".col-sm-4 p")
url_node <- html_elements(sou_urls_html, ".field-title a")

date_txt <- (xml_text(date_node, trim = TRUE))
pres_txt <- (xml_text(president_node, trim = TRUE))
url_txt <- html_attr(url_node, "href")

sou_table <- tibble(
  date = date_txt,
  pres = pres_txt,
  url = url_txt
)

url_table <- url_table |> rbind(sou_table)
```

URL TIBBLE

```
head(url_table, n = 10)
## # A tibble: 10 × 3
##    date         pres                      url
##    <chr>        <chr>                     <chr>
##  1 Mar 04, 2025 Donald J. Trump (2nd Term) /documents/address-before-joint-sess…
##  2 Mar 07, 2024 Joseph R. Biden, Jr.       /documents/address-before-joint-sess…
##  3 Feb 07, 2023 Joseph R. Biden, Jr.       /documents/address-before-joint-sess…
##  4 Dec 21, 2022 U.S. Congress              /documents/address-before-joint-sess…
##  5 Mar 16, 2022 U.S. Congress              /documents/address-before-joint-sess…
##  6 Mar 01, 2022 Joseph R. Biden, Jr.       /documents/address-before-joint-sess…
##  7 Apr 28, 2021 Joseph R. Biden, Jr.       /documents/address-before-joint-sess…
##  8 Feb 04, 2020 Donald J. Trump (1st Term) /documents/address-before-joint-sess…
##  9 Feb 05, 2019 Donald J. Trump (1st Term) /documents/address-before-joint-sess…
## 10 Jan 30, 2018 Donald J. Trump (1st Term) /documents/address-before-joint-sess…
```

CLEAN AND PREP THE SCRAPES

The following code cleans the data collected when scraping for the URLs and prepares the urls to be scraped for the actual text of the speeches.

```
url_table1 <- url_table |>
  mutate(
    date = as_date(date, format = "%B %e, %Y"),
    pres = str_trim(pres),
    pres = str_replace_all(pres, "-", " "),
    url = paste0("https://www.presidency.ucsb.edu",url)
  ) |> arrange(desc(date)) |>
  filter(pres != "U.S. Congress",
         date >= as.Date("2001-01-01")) |>
  distinct(url, .keep_all = TRUE)
```

FUNCTION TO SCRAPE THE ACTUAL SPEECH TEXT

```
speech_conversion <- function(url) {
speech <- read_html(url)

date_node <- html_elements(speech, ".date-display-single")
pres_node <- html_elements(speech, ".diet-title")
speech_node <- html_elements(speech, ".field-docs-content")

date_txt <- xml_text(date_node, trim = TRUE)
pres_txt <- xml_text(pres_node, trim = TRUE)
speech_txt <- xml_text(speech_node, trim = TRUE)

speech_tbl <- tibble(
  date = date_txt,
  pres = pres_txt,
  speech = speech_txt)
}
```

WEB SCRAPE THE LIST OF URLS

The list of urls was then scraped and combined into a dataframe using the function written above and the code below utilizing the map_dfr() function.

```
raw_speech_data <- map_dfr(url_table1$url, speech_conversion)
```

```
head(raw_speech_data, n=5)
## # A tibble: 5 × 3
##   date              pres                       speech
##   <chr>             <chr>                      <chr>
## 1 March 04, 2025    Donald J. Trump (2nd Term) "The President. Thank you. Thank…
## 2 March 07, 2024    Joseph R. Biden, Jr.       "[Before speaking, the President…
## 3 February 07, 2023 Joseph R. Biden, Jr.       "The President. Mr. Speaker——\n[…
## 4 March 01, 2022    Joseph R. Biden, Jr.       "The President. Thank you all ve…
## 5 April 28, 2021    Joseph R. Biden, Jr.       "The President. Thank you. Thank…
```

CLEAN & TIDY SPEECH DATA

The acquired speech text was then formatted and cleaned to allow for consistent referencing during the analysis process.

```
speech_data <- raw_speech_data |>
  mutate(
    date = as.Date(date, format = "%B %e, %Y"),
     pres = str_remove_all(pres, regex("\\(1st Term\\)|\\(2nd Term\\)")),
    pres = trimws(pres),
    speech = str_remove_all(speech, "\\[.*?\\]"),
    pres = str_remove_all(pres, " Jr."),
    pres = str_remove_all(pres, "\\.")
  ) |>
  separate_wider_delim(
    pres,
    delim = " ",
    names = c("first", "last"),
    too_many = "merge",
    cols_remove = FALSE
  ) |>
  separate_wider_delim(
    last,
    delim = " ",
    names = c("initial", "last"),
    too_few = "align_end",
  ) |>
  mutate(
    id = paste0(year(date), "_", last)
  ) |> select(!first:last) |>
  relocate(id, .before = speech) |>
  filter(
    year(date) %in% cong_tidy$Year,
    !between(date, as.Date("1960-02-01"),as.Date("1960-12-31")),
    !date %in% c(as.Date("1948-04-19"), as.Date("1976-01-31"), as.Date("1978-09-18"), as.Date("1982-02-09"),
as.Date("1982-03-15"), as.Date("1982-03-16"), as.Date("1984-06-04"), as.Date("1990-09-11"))
  )
```

SPEECH TEXT SPLIT IN TO SENTENCES TO TRACK PROGRESSION OF SENTIMENT OVER SPEECH

The speech text was then split into individual sentences so that the sentences could be tracked for analysis.

```
sentence_data <- speech_data |>
  arrange(date) |>
  mutate(
    speech_number = row_number()
  ) |>
  separate_longer_delim(speech, delim = ". ") |>
  separate_longer_delim(speech, delim = "! ") |>
  separate_longer_delim(speech, delim = "? ") |>
  separate_longer_delim(speech, delim = "!") |>
  separate_longer_delim(speech, delim = "?") |>
  group_by(speech_number) |>
  mutate(
    sentence_number = row_number()) |> ungroup()
```

LOADED STOP WORDS

A list of stop words was loaded and the text of the speeches were tokenized.

```
data(stop_words)

tidy_speech <- sentence_data |>
  unnest_tokens(word, speech) |>
  anti_join(stop_words, by = "word")
```

SPEECH TEXT TOKENIZED AND STOP WORDS REMOVED

Following toeknization and removal of common stop words, an inspection of the remaining words by frequency reveals that the first ten words are fairly common and non-descriptive.

```
tidy_speech |> group_by(word) |>
  summarize(
    count = n(),
  ) |> arrange(desc(count)) |>
  slice_max(n = 10, order_by = count)
```

```
## # A tibble: 0 × 2
## # ℹ 2 variables: word <chr>, count <int>
```

```
head(tidy_speech, n=10)
```

```
## # A tibble: 0 × 6
## # ℹ 6 variables: date <date>, pres <chr>, id <chr>, speech_number <int>,
## #   sentence_number <int>, word <chr>
```

CREATE A LIST OF CUSTOM STOP WORDS…

A list of custom stop words were created to eliminate the most commonly used words.

```
custom_stop_words <- tidy_speech |> group_by(word) |>
  summarize(
    count = n(),
  ) |> arrange(desc(count)) |> slice_max(n = 10, order_by = count)
|>
  mutate(lexicon = "custom") |>
  select(!count)
```

...AND REMOVED THEM

```
tidy_speech <- tidy_speech |>
  anti_join(custom_stop_words, by = "word")
```

NEW LIST OF MOST COMMON WORDS

The current ten most common words. The counts of these words are much more similar than the previously removed words.

```
tidy_speech |> group_by(word) |>
  summarize(
    count = n(),
  ) |> arrange(desc(count)) |>
  slice_max(n = 10, order_by = count)
```

```
## # A tibble: 0 × 2
## # ℹ 2 variables: word <chr>, count <int>
```

SENTIMENT ANALYSIS

For sentiment analysis the AFINN lexicon was selected as it gives sentiments with an ordinal magnitude from -5 to 5. The following code joins the speech tokens with the AFINN lexicon.

```
afinn <- tidy_speech |>
  inner_join(get_sentiments("afinn"), by = join_by(word)) |>
  group_by(id, index = sentence_number %/% 5) |>
  summarise(sentiment = sum(value)) |>
  mutate(method = "AFINN")
```

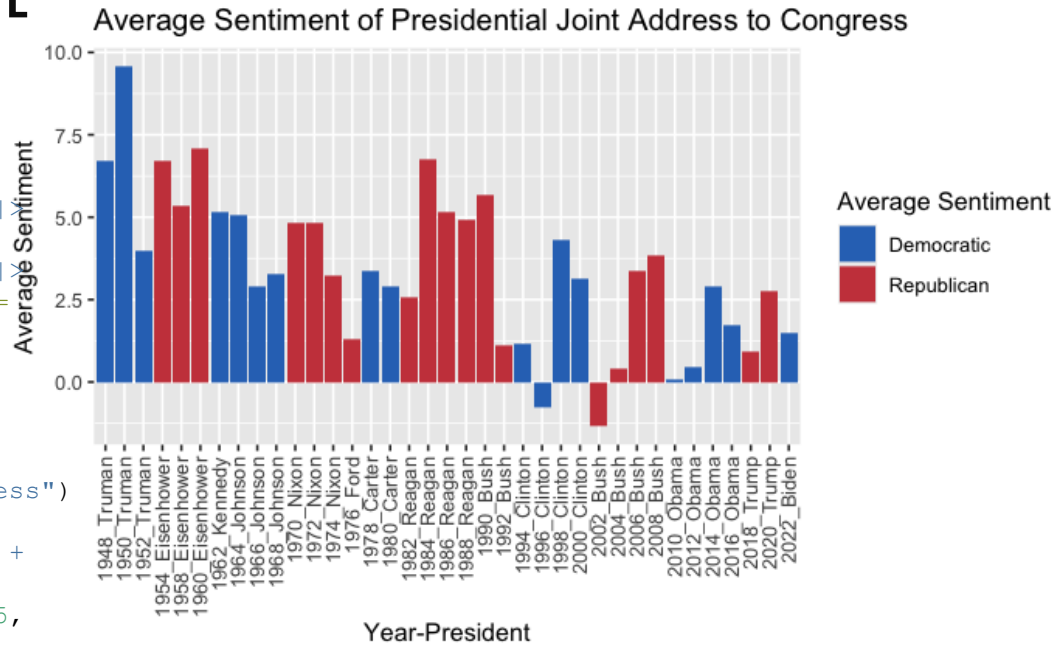
CREATED A DATAFRAME TO AID IN PLOTTING THE RESULTS

The following code then prepares the data to be plotted.

```
afinn_plot <- afinn |>
  separate_wider_delim(id,
                       delim = "_",
                       names = c("year", "pres"),
                       cols_remove = FALSE) |>
  mutate(year = as.numeric(year)) |>
  left_join(us_pres, by = join_by(year))
```

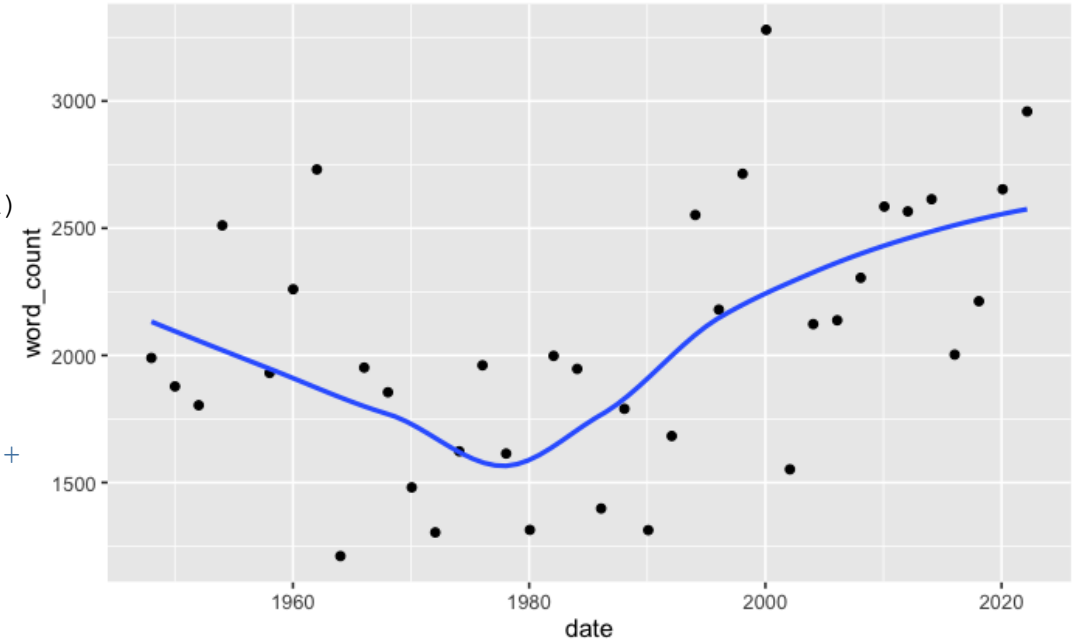
A PLOT OF THE AVERAGE SENTIMENT OVER TIME

```
 afinn_plot |> group_by(id, party) |>
summarise(avg_sentiment =
mean(sentiment), .groups = "drop") |>
ggplot(aes(id, avg_sentiment, fill =
party)) +
  geom_col() +
   xlab("Year-President") +
   ylab("Average Sentiment") +
   ggtitle("Average Sentiment of
Presidential Joint Address to Congress")
+
   labs(fill = "Average Sentiment") +
   theme(axis.text.x =
element_text(angle = 90, vjust = 0.5,
hjust = 1)) +
   scale_fill_manual(values =
c("Democratic" = "#2E74C0", "Republican"
= "#CB454A"))
```



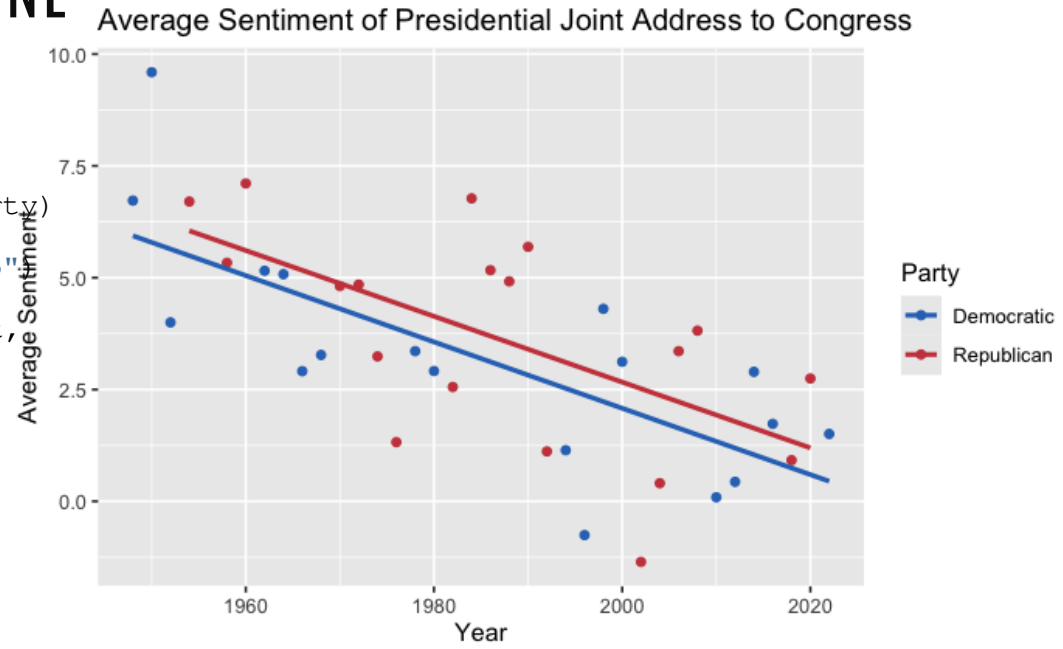
WORD COUNT FOR SPEECHES

```
tidy_speech |> group_by(date, id)
|>
  summarise(
    word_count = n(),
    .groups = "drop"
  ) |>
  ggplot(aes(date, word_count)) +
  geom_point() +
  geom_smooth(method = "loess",
se = FALSE)
```



SCATTERPLOT OF DECLINE IN SENTIMENT

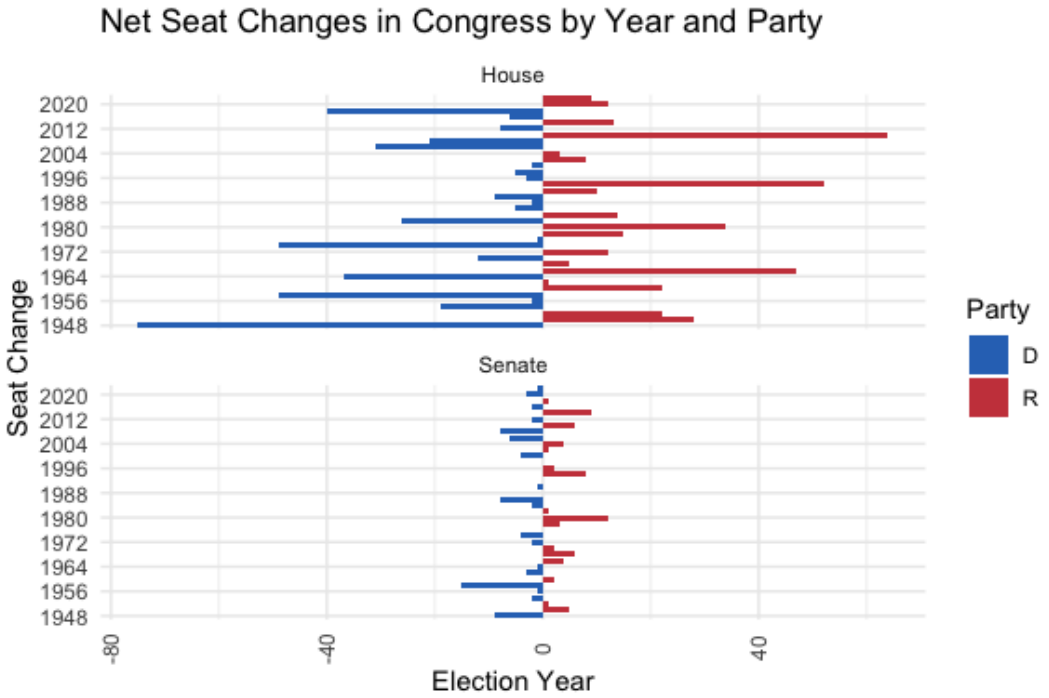
```
afinn_plot |> group_by(year, party)
|> summarise(avg_sentiment =
mean(sentiment), .groups = "drop")
|>
  ggplot(aes(year, avg_sentiment,
color = party)) +
  geom_point() +
  xlab("Year") +
  ylab("Average Sentiment") +
  ggtitle("Average Sentiment of
Presidential Joint Address to
Congress") +
  labs(color = "Party") +
  scale_color_manual(values =
c("Democratic" = "#2E74C0",
"Republican" = "#CB454A")) +
  geom_smooth(method = "lm", se =
FALSE)
```



Seat Changes in Congress

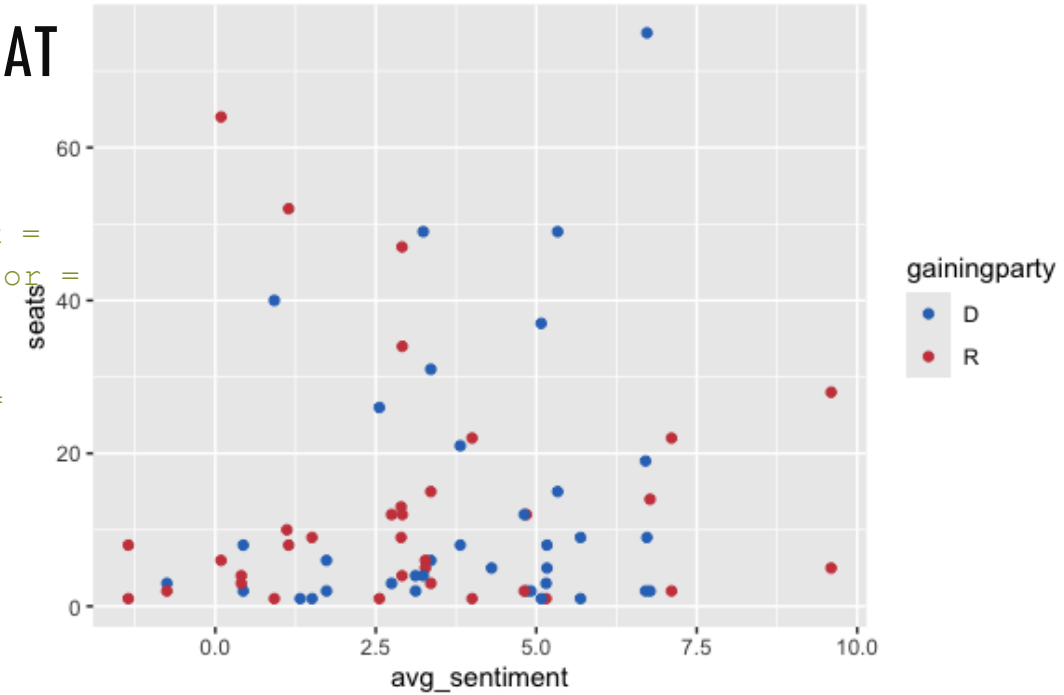
```
election_results <- cong_tidy |>
  mutate(
    seats = as.numeric(seats),
    seat_change_sign = case_when(
      gainingparty == "D" ~ -seats,
      gainingparty == "R" ~ seats,
      TRUE ~ 0
    )
  )

ggplot(election_results, aes(y = factor(year
seat_change_sign, fill = gainingparty)) +
  geom_col() +
  geom_hline(yintercept = 0, color = "black"
  facet_wrap(~ chamber, ncol = 1, scales =
"free_y") +
  scale_fill_manual(values = c("D" = "#2E74C
= "#CB454A")) +
  labs(
    x = "Election Year",
    y = "Seat Change",
    title = "Net Seat Changes in Congress by
and Party",
    fill = "Party"
  ) +
  scale_y_discrete(breaks = function(x) x[se
length(x), by = 4)]) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 9
vjust = 0.5))
```



SCATTERPLOT OF AVG_SENTIMENT VS SEAT CHANGES BY CHAMBER

```
ggplot(afinn_avg_senti, aes(x =
avg_sentiment, y = seats, color =
gainingparty)) +
  geom_point() +
  scale_color_manual(values =
c("D" = "#2E74C0", "R" =
"#CB454A"))
```



LINEAR REGRESSION OF THE RELATIONSHIP BETWEEN SENTIMENT AND SEAT CHANGES

```
Call:
lm(formula = change ~ avg_sentiment, data = afinn_avg_senti)

Residuals:
    Min      1Q  Median      3Q     Max
-54.257  -6.052   2.408   9.891  77.123

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -9.8455     4.1994  -2.345    0.022 *
avg_sentiment  1.1485     0.9965   1.152    0.253
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.05 on 68 degrees of freedom
Multiple R-squared:  0.01916,   Adjusted R-squared:  0.004734
F-statistic: 1.328 on 1 and 68 DF,  p-value: 0.2532
```

THANKS FOR LISTENING

These slides were produced directly from R Studio.