The cost of changing cloture votes

Max H. Feinleib

Table of contents


```
# packages used
library(dplyr)
library(ggplot2)
library(readr)
library(stringr)
library(filibustr)
```

1 Introduction

I measure the distance between the breakpoints on failed cloture votes and the ideal point of the potential pivotal vote for cloture. This distance can be interpreted as a measurement of the cost a new filibuster rule would have to impose to change the outcome of a cloture vote.

2 Data

For this analysis, I used: * failed cloture votes (on final passage, not a motion to proceed) * since 1977 (so that all votes are under the current cloture rules).

```r
# downloading from Voteview
# s_votes_data <- get_voteview_rollcall_votes(chamber = "s", congress = 95:117)
# s_mem_votes_data <- get_voteview_member_votes(chamber = "s", congress = 95:117)
# s_mem_data <- get_voteview_members(chamber = "s", congress = 95:117)

# use local files
s_votes_data <- read_csv("s_votes_data.csv")
```

```
Rows: 16712 Columns: 18
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (6): chamber, bill_number, vote_result, vote_desc, vote_question, dtl_...
dbl  (11): congress, rollnumber, session, clerk_rollnumber, yea_count, nay_c...
date  (1): date

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.
```

```r
s_mem_votes_data <- read_csv("s_mem_votes_data.csv")
```

```
Rows: 1685290 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): chamber
dbl (5): congress, rollnumber, icpsr, cast_code, prob

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.
```

```r
s_mem_data <- read_csv("s_mem_data.csv")
```

```
Rows: 2376 Columns: 22
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): chamber, state_abbrev, bioname, bioguide_id
dbl (17): congress, icpsr, state_icpsr, district_code, party_code, occupancy...
lgl  (1): conditional

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.
```

3 Analysis

First, I filter to failed cloture votes for a final passage vote.

```
## failed cloture votes

# calculate whether cloture threshold is 51 or 60
# TODO: doesn't consider VP tiebreakers
get_cloture_threshold <- function(df) {
  # nuclear option dates
  nuclear_2013 <- as.Date("2013-11-21")
  nuclear_2017 <- as.Date("2017-04-06")

  df |>
    mutate(
      nomination = str_starts(bill_number, "PN[:digit:]"),
      scotus = str_detect(vote_desc, "(Associate|Chief) Justice"),
      threshold = case_when(
        # SCOTUS nominations
        nomination & scotus
        & (date > nuclear_2017 | (congress == 115 & rollnumber == 110)) ~ 51,
        # other nominations
        nomination & !scotus
        & (date > nuclear_2013 | (congress == 113 & rollnumber == 244)) ~ 51,
        # 60 for everything else
        .default = 60),
      .after = nay_count
    ) |>
    select(-nomination, -scotus)
}

s_failed_cvotes <- s_votes_data |>
  filter(vote_result == "Cloture Motion Rejected") |>
  get_cloture_threshold() |>
  mutate(
    votes_needed = threshold - yea_count,
    # use hypotenuse of spread to measure distances
    nominate_spread_dist = sqrt(nominate_spread_1 ** 2 + nominate_spread_2 ** 2),
    .after = threshold
  )

s_failed_cvotes |>
```

```
  mutate(total_votes = yea_count + nay_count, .before = yea_count) |>
  arrange(total_votes)
```

```
# A tibble: 431 x 22
   congress chamber rollnumber date       session clerk_rollnumber total_votes
      <dbl> <chr>         <dbl> <date>       <dbl>            <dbl>       <dbl>
 1      116 Senate           52 2019-03-26       1               52          57
 2      103 Senate          102 1993-04-05       1              102          78
 3      110 Senate          130 2007-04-16       1              130          81
 4      106 Senate          668 2000-11-01       2              294          83
 5      107 Senate          259 2001-07-27       1              259          84
 6      110 Senate          587 2008-06-06       2              145          84
 7      110 Senate          629 2008-07-26       2              187          85
 8      111 Senate          419 2010-02-09       2               22          85
 9      112 Senate          131 2011-09-12       1              131          86
10      112 Senate          227 2011-12-12       1              227          86
# i 421 more rows
# i 15 more variables: yea_count <dbl>, nay_count <dbl>, threshold <dbl>,
#   votes_needed <dbl>, nominate_spread_dist <dbl>, nominate_mid_1 <dbl>,
#   nominate_mid_2 <dbl>, nominate_spread_1 <dbl>, nominate_spread_2 <dbl>,
#   nominate_log_likelihood <dbl>, bill_number <chr>, vote_result <chr>,
#   vote_desc <chr>, vote_question <chr>, dtl_desc <chr>
```

Now, I find the pivotal member on these failed cloture votes. I filter the votes to Nay votes with a probability greater than 50% (so they are explained by pivotal models). I also filter out votes with 100% probability, as these votes are unlikely to change in the face of higher-cost filibustering.

```
## finding pivotal votes

s_mem_positions <- s_mem_data |>
  # remove presidents
  filter(chamber == "Senate") |>
  # relevant columns
  select(congress, chamber, icpsr, bioname, party_code, state_abbrev, nominate_dim1, nominate

# cast codes (source: https://voteview.com/articles/data_help_votes)
# TODO: use `filibustr` version when that's available
voteview_cast_codes <- tibble(cast_code = 0:9,
                              vote_cast = factor(
                                c("Not a Member",
```

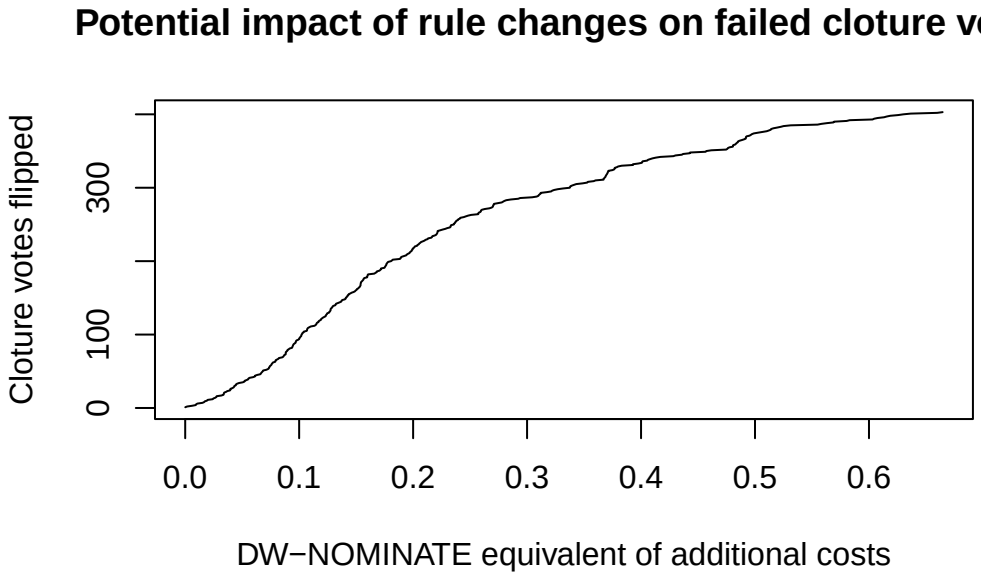
```
                                  "Yea", "Paired Yea", "Announced Yea",
                                  "Announced Nay", "Paired Nay", "Nay",
                                  "Present", "Present", "Not Voting")
                                ))

s_mem_failed_cvotes <- s_mem_votes_data |>
  # filter: sen_mem_votes_data uses the `rollnumber` as a foreign key
  semi_join(s_failed_cvotes, by = c("congress", "rollnumber")) |>
  # add cast_code descriptions for easier reading
  left_join(voteview_cast_codes, by = "cast_code") |>
  # filter: Nay votes with 50-99.9% probability
  filter(str_detect(vote_cast, "Nay"), prob >= 50, prob < 100) |>
  # add member ideologies
  left_join(s_mem_positions, by = c("congress", "chamber", "icpsr")) |>
  arrange(rollnumber, prob)

# find (60-yeas)'th member with lowest probability of Nay vote
pivotal_votes <- s_mem_failed_cvotes |>
  left_join(s_failed_cvotes |>
              select(congress, rollnumber, threshold, votes_needed, nominate_spread_dist, bi
            by = c("congress", "rollnumber")) |>
  group_by(congress, rollnumber, bill_number, threshold, votes_needed) |>
  # ensure there are enough flippable votes
  filter(n() >= votes_needed) |>
  # find pivotal votes
  mutate(rank = min_rank(prob)) |>
  filter(rank <= votes_needed) |>
  filter(rank == max(rank))
```

I multiply the pivotal vote's probability by the DW-NOMINATE spread of the probabilities to estimate the distance.

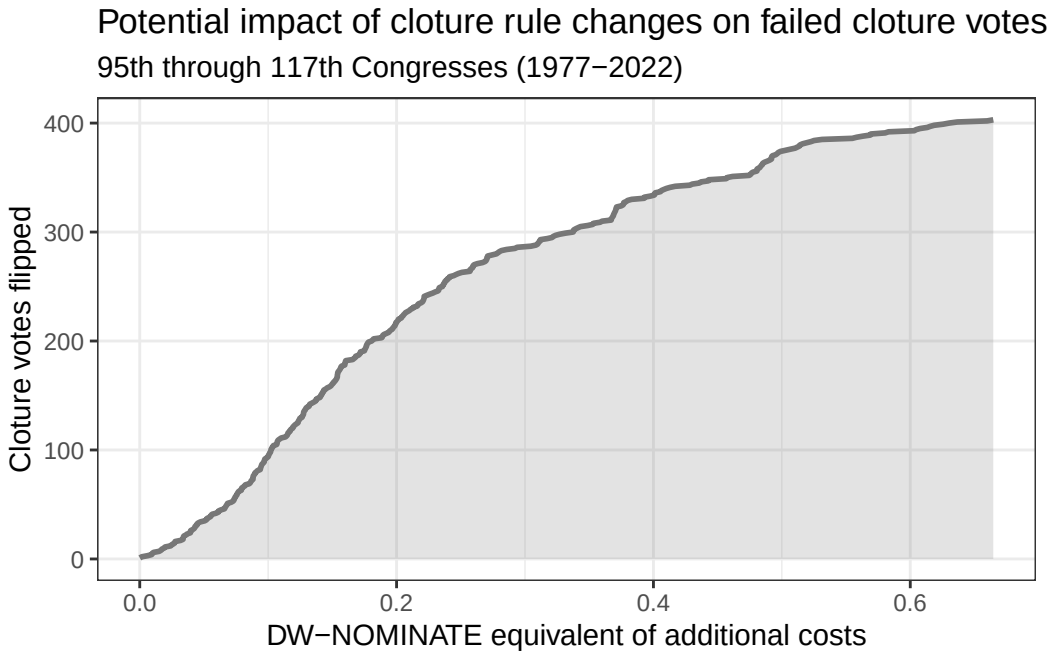
```
vote_costs <- pivotal_votes |>
  summarize(prob = first(prob),
            prob_left = prob - 50,
            nominate_spread_dist = first(nominate_spread_dist),
            n_pivotal_votes = n(),
            .groups = "drop") |>
  # distance you have to move: (prob - 50) * spread
  mutate(probXdist = prob_left * nominate_spread_dist / 100,
         pd_rank = row_number(probXdist)) |>
  arrange(probXdist)
```

4 Findings

Finally, I plot the cumulative distribution function of the vote costs. This shows how many cloture votes would flip as the costs of filibustering increase. Costs are translated into an equivalent movement on the DW-NOMINATE scale.




```
NULL
```



Based on the slope of the graph, the largest impact on cloture votes would appear to come from rule changes that impose costs equivalent to 0.1-0.2 units on the DW-NOMINATE scale.