

2024 General Election Forecasting Model

POLSCI 239 - Assignment Four

Jack Regan

Data

The data for this model is borrowed from ABC's 538 general election state polling dataset. (Full citation in README)

```
raw_polling_data <- read_csv("data/president_polls.csv", show_col_types = FALSE)
glimpse(raw_polling_data)
```

```
Rows: 15,971
Columns: 52
$ poll_id          <dbl> 88806, 88806, 88836, 88836, 88817, 88817, 88~
$ pollster_id      <dbl> 770, 770, 1895, 1895, 1741, 1741, 770, 770, ~
$ pollster         <chr> "TIPP", "TIPP", "Quantus Insights", "Quantus~
$ sponsor_ids      <dbl> NA, NA, 2184, 2184, NA, NA, NA, NA, NA, NA, ~
$ sponsors          <chr> NA, NA, "TrendingPolitics", "TrendingPolitic~
$ display_name     <chr> "TIPP Insights", "TIPP Insights", "Quantus I~
$ pollster_rating_id <dbl> 144, 144, 859, 859, 721, 721, 144, 144, 338,~
$ pollster_rating_name <chr> "TIPP Insights", "TIPP Insights", "Quantus I~
$ numeric_grade    <dbl> 1.8, 1.8, NA, NA, NA, NA, 1.8, 1.8, 0.7, 0.7~
$ pollscore        <dbl> -0.4, -0.4, NA, NA, NA, NA, -0.4, -0.4, 0.6,~
$ methodology      <chr> "Online Panel", "Online Panel", "Online Pane~
$ transparency_score <dbl> 3.0, 3.0, 5.5, 5.5, 8.0, 8.0, 3.0, 3.0, 4.0,~
$ state            <chr> NA, NA, "Pennsylvania", "Pennsylvania", "Flo~
$ start_date       <chr> "10/18/24", "10/18/24", "10/17/24", "10/17/2~
$ end_date         <chr> "10/20/24", "10/20/24", "10/20/24", "10/20/2~
$ sponsor_candidate_id <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ sponsor_candidate <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ sponsor_candidate_party <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ endorsed_candidate_id <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ endorsed_candidate_name <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

\$ endorsed_candidate_party	<lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ question_id	<dbl> 213459, 213459, 213538, 213538, 213472, 2134~
\$ sample_size	<dbl> 1244, 1244, 840, 840, 400, 400, 1254, 1254, ~
\$ population	<chr> "lv", "lv", "lv", "lv", "lv", "lv", "lv", "lv", "lv", ~
\$ subpopulation	<lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ population_full	<chr> "lv", "lv", "lv", "lv", "lv", "lv", "lv", "lv", "lv", ~
\$ tracking	<lgl> TRUE, TRUE, NA, NA, NA, NA, TRUE, TRUE, NA, ~
\$ created_at	<chr> "10/21/24 08:43", "10/21/24 08:43", "10/21/2~
\$ notes	<chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ url	<chr> "https://tippinsights.com/tipp-tracking-poll~
\$ url_article	<chr> "https://tippinsights.com/tipp-tracking-poll~
\$ url_topleft	<chr> NA, NA, "https://docs.google.com/document/d/~
\$ url_crosstab	<chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ source	<dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ internal	<lgl> NA, NA, FALSE, FALSE, FALSE, FALSE, NA, NA, ~
\$ partisan	<chr> NA, NA, "REP", "REP", NA, NA, NA, NA, "REP", ~
\$ race_id	<dbl> 8914, 8914, 8872, 8872, 8778, 8778, 8914, 89~
\$ cycle	<dbl> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 20~
\$ office_type	<chr> "U.S. President", "U.S. President", "U.S. Pr~
\$ seat_number	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
\$ seat_name	<lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ election_date	<chr> "11/5/24", "11/5/24", "11/5/24", "11/5/24", ~
\$ stage	<chr> "general", "general", "general", "general", ~
\$ nationwide_batch	<lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
\$ ranked_choice_reallocated	<lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
\$ ranked_choice_round	<dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
\$ hypothetical	<lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
\$ party	<chr> "DEM", "REP", "DEM", "REP", "DEM", "REP", "D~
\$ answer	<chr> "Harris", "Trump", "Harris", "Trump", "Harri~
\$ candidate_id	<dbl> 16661, 16651, 16661, 16651, 16661, 16651, 16~
\$ candidate_name	<chr> "Kamala Harris", "Donald Trump", "Kamala Har~
\$ pct	<dbl> 47.0, 48.0, 48.2, 50.3, 45.4, 54.6, 47.0, 49~

Data Cleaning

The model will only calculate win percentages for toss up states.

```
toss_up_states <- c("Michigan", "Nevada",
  "Arizona", "New Mexico",
  "Wisconsin", "Pennsylvania",
  "North Carolina", "Georgia")

polling_data <- raw_polling_data |>
  select(
    poll_id,
    state,
    end_date,
    sample_size,
    candidate_name,
    pct
  ) |>
  rename("dem_pct" = pct) |>
  filter(candidate_name == "Kamala Harris" & state %in% toss_up_states) |>
  mutate(end_date = as.Date(end_date, format = "%m/%d/%y")) |>
  arrange(end_date) |>
  drop_na(sample_size)

glimpse(polling_data)
```

Rows: 847

Columns: 6

```
$ poll_id      <dbl> 84542, 84542, 84543, 84543, 84544, 84544, 84545, 84545, ~
$ state        <chr> "Arizona", "Arizona", "Georgia", "Georgia", "Michigan", ~
$ end_date     <date> 2023-11-03, 2023-11-03, 2023-11-03, 2023-11-03, 2023-1~
$ sample_size  <dbl> 603, 603, 629, 629, 616, 616, 611, 611, 600, 600, 603, ~
$ candidate_name <chr> "Kamala Harris", "Kamala Harris", "Kamala Harris", "Kam~
$ dem_pct      <dbl> 43.0, 43.0, 44.0, 44.0, 45.0, 48.0, 42.0, 42.0, 44.0, 4~
```

Summary Statistics

```
options(pillar.sigfig = 7)
```

```
polling_data |>
  group_by(state) |>
  summarise(
    poll_count = n(),
    raw_harris_approval = mean(dem_pct),
    earliest_poll = min(end_date),
    most_recent_poll = max(end_date)
  )
```

```
# A tibble: 8 x 5
```

	state <chr>	poll_count <int>	raw_harris_approval <dbl>	earliest_poll <date>	most_recent_poll <date>
1	Arizona	111	46.50595	2023-11-03	2024-10-18
2	Georgia	119	46.41235	2023-11-03	2024-10-18
3	Michigan	124	47.50210	2023-11-03	2024-10-18
4	Nevada	80	46.96812	2023-11-03	2024-10-18
5	New Mexico	10	48.96	2024-08-03	2024-10-18
6	North Carolina	111	47.18766	2024-02-16	2024-10-18
7	Pennsylvania	166	47.52759	2023-11-03	2024-10-20
8	Wisconsin	126	48.46984	2023-11-03	2024-10-18

Weighting Data by Sample Size

Each poll was weighted using a function based on its sample size. Specifically, I take the square root of the median sample size for each state and then multiplied the Harris approval percentage for each poll by the square root of the poll's sample size divided by that state's square-rooted median sample size. This methodology was adopted from 538's weighting guidelines and then adjusted to fit the specifications of the dataset. A new "adjusted_pct" variable was applied to each poll in the dataset.

```
options(pillar.sigfig = 7)

square_root_median_sample_size_by_state <- polling_data |>
  group_by(state) |>
  summarize(
    square_root_median_sample_size = sqrt(median(sample_size, na.rm = TRUE))
  )
as_tibble(square_root_median_sample_size_by_state)
```

```
# A tibble: 8 x 2
  state          square_root_median_sample_size
  <chr>              <dbl>
1 Arizona          27.85678
2 Georgia          28.26659
3 Michigan         26.22975
4 Nevada          26.01922
5 New Mexico       22.94559
6 North Carolina   28.28427
7 Pennsylvania     28.33725
8 Wisconsin       26.45751
```

```
polling_data <- polling_data |>
  mutate(adjusted_pct = case_when(
    state == "Arizona" ~ sqrt(sample_size)/27.85678*dem_pct,
    state == "Georgia" ~ sqrt(sample_size)/28.26659*dem_pct,
    state == "Michigan" ~ sqrt(sample_size)/26.22975*dem_pct,
    state == "Nevada" ~ sqrt(sample_size)/26.01922*dem_pct,
    state == "New Mexico" ~ sqrt(sample_size)/22.94559*dem_pct,
    state == "North Carolina" ~ sqrt(sample_size)/28.28427*dem_pct,
    state == "Pennsylvania" ~ sqrt(sample_size)/28.33725*dem_pct,
    state == "Wisconsin" ~ sqrt(sample_size)/26.45751*dem_pct
  )
```

```
)
```

```
glimpse(polling_data)
```

```
Rows: 847
```

```
Columns: 7
```

```
$ poll_id      <dbl> 84542, 84542, 84543, 84543, 84544, 84544, 84545, 84545, ~
$ state        <chr> "Arizona", "Arizona", "Georgia", "Georgia", "Michigan", ~
$ end_date     <date> 2023-11-03, 2023-11-03, 2023-11-03, 2023-11-03, 2023-1~
$ sample_size  <dbl> 603, 603, 629, 629, 616, 616, 611, 611, 600, 600, 603, ~
$ candidate_name <chr> "Kamala Harris", "Kamala Harris", "Kamala Harris", "Kam~
$ dem_pct      <dbl> 43.0, 43.0, 44.0, 44.0, 45.0, 48.0, 42.0, 42.0, 44.0, 4~
$ adjusted_pct <dbl> 37.90497, 37.90497, 39.03953, 39.03953, 42.58030, 45.41~
```

Exponentially Weighted Moving Average

In an EWMA calculation, recent data points are assigned more weight than older points. This makes the average more responsive to recent changes in the data. The lambda variable controls how much weight is assigned to more recent data points. I assign a lambda value of 0.95 in order to assign greater weight to more recent polls. The smoothed average provides a single value for each state that represents the democrat win percentage. This value is then used in my forecast prediction. More documentation on the EWMA can be found here (<https://www.investopedia.com/articles/07/ewma.asp>).

```
options(pillar.sigfig = 7)

calculate_ewma <- function(data, raw_average, lambda) {

  ewma <- numeric(length(data[[raw_average]]))
  ewma[1] <- data[[raw_average]][1]

  for (i in 2:length(data[[raw_average]])) {
    ewma[i] <- lambda * data[[raw_average]][i] + (1 - lambda) * ewma[i - 1]
  }
  return(ewma[length(ewma)])
}

polling_data |>
  group_by(state) |>
  summarise(
    ewma_adjusted_pct = calculate_ewma(cur_data(), "adjusted_pct", 0.85)
  )
```

```
Warning: There was 1 warning in `summarise()`.
i In argument: `ewma_adjusted_pct = calculate_ewma(cur_data(), "adjusted_pct",
  0.85)`.
```

```
i In group 1: `state = "Arizona"`.
Caused by warning:
! `cur_data()` was deprecated in dplyr 1.1.0.
i Please use `pick()` instead.
```

```
# A tibble: 8 x 2
  state      ewma_adjusted_pct
  <chr>      <dbl>
1 Arizona    46.93262
2 Georgia    51.00229
3 Michigan   55.61154
```

4 Nevada	44.04701
5 New Mexico	65.79504
6 North Carolina	50.12148
7 Pennsylvania	49.61117
8 Wisconsin	44.71328

Additional Considerations and Data Limitations

This dataset introduces several inconsistencies to the model which will be addressed here. First, the inconsistent number of polls conducted within each state creates uncertainty in the accuracy of the data. Second, the variability of polling sources opens the data to potential bias. FiveThirtyEight uses extensive guidelines when choosing polls to include within their data in order to account for bias; however, this is mostly a subjective science and isn't statistically grounded in my model. Information on 538's polling policy can be found here (<https://fivethirtyeight.com/features/polls-policy-and-faqs/>). Third, this model uses a rudimentary modeling algorithm that adjusts based on sample size and time decay. Other weights such as pollster rating and margin of error are common strategies, but are not considered in this model.

Weighting and averaging data admits a certain level of subjectivity into the data as the methods by which the data is adjusted are largely statistically insignificant. The weighting and averaging methods I chose were subjective choices influenced by common practice but are not scientifically grounded as the best practice.