# Gerrymandering Metrics

### 2024 Election Results and the 2024 District Map

```
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
votes <- read_csv("data/cleaned_g24_sov_by_svprec.csv")
```

```
Rows: 38657 Columns: 77
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (49): fips, svprec, svprec_key, election, geo_type, assaip01, assdem01, ...
dbl (28): county, addist, cddist, sddist, bedist, totreg, demreg, repreg, ai...

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
district_votes <- votes %>%
  filter(!is.na(cddist)) %>%
  group_by(cddist) %>%
  summarise(
    total_dem = sum(demvote, na.rm = TRUE),
    total_rep = sum(repvote, na.rm = TRUE)
  ) %>%
  mutate(
    total = total_dem + total_rep,
    share_dem = total_dem / total
  )
```

```r
mean_median_score <- mean(district_votes$share_dem) - median(district_votes$share_dem)
mean_median_score
```

```
[1] -0.01163046
```

```r
efficiency_gap <- function(dem_votes, rep_votes) {
  total_votes <- dem_votes + rep_votes
  dem_wasted <- ifelse(dem_votes > rep_votes,
                       dem_votes - total_votes / 2,
                       dem_votes)
  rep_wasted <- ifelse(rep_votes > dem_votes,
                       rep_votes - total_votes / 2,
                       rep_votes)
  sum(dem_wasted - rep_wasted) / sum(total_votes)
}
```

```r
eg <- efficiency_gap(district_votes$total_dem, district_votes$total_rep)
eg
```

```
[1] -0.2552036
```

### 2024 Election Results and the proposed 2025 District Map

```r
library(tidyverse)

sr_votes_raw <- read_csv("data/state_g24_sov_data_by_g24_srprec.csv")
```

```
Rows: 25245 Columns: 76
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (49): FIPS, SRPREC, ELECTION, SRPREC_KEY, GEO_TYPE, ASSAIP01, ASSDEM01, ...
dbl (27): COUNTY, ADDIST, CDDIST, SDDIST, BEDIST, TOTREG, DEMREG, REPREG, AI...

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
sr_blk       <- read_csv("data/state_g24_sr_blk_map.csv")
```

```
Rows: 421137 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (6): BLOCK_KEY, FIPS, SRPREC, SRPREC_KEY, ELECTION, GEO_TYPE
dbl (8): COUNTY, TRACT, BLOCK, BLKREG, SRTOTREG, PCTSRPREC, BLKTOTREG, PCTBLK

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
ab604        <- read_csv("data/ab604.csv")
```

```
Rows: 519722 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): 060650444033050, 25

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
names(sr_votes_raw)
```

```
 [1] "COUNTY"     "FIPS"       "SRPREC"     "ELECTION"   "SRPREC_KEY"
 [6] "GEO_TYPE"   "ADDIST"     "CDDIST"     "SDDIST"     "BEDIST"
[11] "TOTREG"     "DEMREG"     "REPREG"     "AIPREG"     "GRNREG"
[16] "LIBREG"     "NLPREG"     "REFREG"     "DCLREG"     "MSCREG"
[21] "TOTVOTE"    "DEMVOTE"    "REPVOTE"    "AIPVOTE"    "GRNVOTE"
[26] "LIBVOTE"    "NLPVOTE"    "REFVOTE"    "DCLVOTE"    "MSCVOTE"
[31] "PRCVOTE"    "ABSVOTE"    "ASSAIP01"   "ASSDEM01"   "ASSDEM02"
[36] "ASSREP01"   "ASSREP02"   "CNGDEM01"   "CNGDEM02"   "CNGIND01"
[41] "CNGREP01"   "CNGREP02"   "PRSAIP01"   "PRSDEM01"   "PRSGRN01"
[46] "PRSLIB01"   "PRSPAF01"   "PRSREP01"   "PR_2_N"     "PR_2_Y"
[51] "PR_32_N"    "PR_32_Y"    "PR_33_N"    "PR_33_Y"    "PR_34_N"
[56] "PR_34_Y"    "PR_35_N"    "PR_35_Y"    "PR_36_N"    "PR_36_Y"
[61] "PR_3_N"     "PR_3_Y"     "PR_4_N"     "PR_4_Y"     "PR_5_N"
[66] "PR_5_Y"     "PR_6_N"     "PR_6_Y"     "SENDEM01"   "SENDEM02"
[71] "SENREP01"   "SENREP02"   "USPDEM01"   "USPREP01"   "USSDEM01"
[76] "USSREP01"
```

```r
names(sr_blk)
```

```
 [1] "COUNTY"     "BLOCK_KEY"  "FIPS"       "SRPREC"     "TRACT"
 [6] "SRPREC_KEY" "ELECTION"   "GEO_TYPE"   "BLOCK"      "BLKREG"
[11] "SRTOTREG"   "PCTSRPREC"  "BLKTOTREG"  "PCTBLK"
```

```r
names(ab604)
```

```
[1] "060650444033050" "25"
```

```r
ab604 <- read_csv(
  "data/ab604.csv",
  col_names = c("BLOCK_KEY", "AB604_DIST")
)
```

```
Rows: 519723 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): BLOCK_KEY, AB604_DIST

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
names(ab604)
```

```
[1] "BLOCK_KEY"  "AB604_DIST"
```

```r
library(tidyverse)

sr_votes_raw <- read_csv("data/state_g24_sov_data_by_g24_srprec.csv")
```

```
Rows: 25245 Columns: 76
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (49): FIPS, SRPREC, ELECTION, SRPREC_KEY, GEO_TYPE, ASSAIP01, ASSDEM01, ...
dbl (27): COUNTY, ADDIST, CDDIST, SDDIST, BEDIST, TOTREG, DEMREG, REPREG, AI...

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
sr_blk       <- read_csv("data/state_g24_sr_blk_map.csv")
```

```
Rows: 421137 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (6): BLOCK_KEY, FIPS, SRPREC, SRPREC_KEY, ELECTION, GEO_TYPE
dbl (8): COUNTY, TRACT, BLOCK, BLKREG, SRTOTREG, PCTSRPREC, BLKTOTREG, PCTBLK

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
ab604        <- read_csv("data/ab604.csv", col_names = c("BLOCK_KEY", "AB604_DIST"))
```

```
Rows: 519723 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): BLOCK_KEY, AB604_DIST

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
sr_votes_cng <- sr_votes_raw |>
  mutate(
    CNGDEM01_num = as.numeric(CNGDEM01),
    CNGDEM02_num = as.numeric(CNGDEM02),
    CNGREP01_num = as.numeric(CNGREP01),
    CNGREP02_num = as.numeric(CNGREP02),
    dem_cng = replace_na(CNGDEM01_num, 0) + replace_na(CNGDEM02_num, 0),
    rep_cng = replace_na(CNGREP01_num, 0) + replace_na(CNGREP02_num, 0)
  ) |>
  select(SRPREC_KEY, dem_cng, rep_cng)
```

```
Warning: There were 4 warnings in `mutate()`.
The first warning was:
i In argument: `CNGDEM01_num = as.numeric(CNGDEM01)`.
Caused by warning:
! NAs introduced by coercion
i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

```r
sr_blk_votes <- sr_blk |>
  left_join(sr_votes_cng, by = "SRPREC_KEY") |>
  mutate(
    frac_sr   = PCTSRPREC / 100,
    dem_block = dem_cng * frac_sr,
    rep_block = rep_cng * frac_sr
  )

sr_blk_votes_cd <- sr_blk_votes |>
  left_join(ab604, by = "BLOCK_KEY")

ab604_district_results <- sr_blk_votes_cd |>
  group_by(AB604_DIST) |>
  summarise(
    dem_votes = sum(dem_block, na.rm = TRUE),
    rep_votes = sum(rep_block, na.rm = TRUE),
    .groups = "drop"
  ) |>
  mutate(
    total_votes = dem_votes + rep_votes,
    dem_share   = dem_votes / total_votes,
    rep_share   = rep_votes / total_votes,
    winner      = if_else(dem_votes > rep_votes, "DEM", "REP")
  )

write_csv(ab604_district_results, "data/ab604_district_results_2024_simulated.csv")

ab604_district_results
```

```
# A tibble: 53 x 7
   AB604_DIST dem_votes rep_votes total_votes dem_share rep_share winner
   <chr>          <dbl>     <dbl>       <dbl>     <dbl>     <dbl> <chr>
 1 01           182536.   150547.     333083.     0.548     0.452 DEM
 2 02           231970.   148401.     380371.     0.610     0.390 DEM
 3 03           189935.   164638.     354572.     0.536     0.464 DEM
 4 04           189348.   143688.     333035.     0.569     0.431 DEM
 5 05           118108.   229040.     347148.     0.340     0.660 REP
 6 06           171008.   151904.     322912.     0.530     0.470 DEM
 7 07           189330.   144663.     333992.     0.567     0.433 DEM
 8 08           206138.    95613.     301751.     0.683     0.317 DEM
 9 09           150450.   107208.     257658.     0.584     0.416 DEM
10 10           241492.   122289.     363782.     0.664     0.336 DEM
# i 43 more rows
```

```
ab604_district_results |>
  count(winner)
```

```
# A tibble: 2 x 2
  winner     n
  <chr>  <int>
1 DEM       46
2 REP        7
```

```
library(tidyverse)

ab604_results <- read_csv("data/ab604_district_results_2024_simulated.csv")
```

```
Rows: 53 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): AB604_DIST, winner
dbl (5): dem_votes, rep_votes, total_votes, dem_share, rep_share
```

```
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.
```

```
mean_median_score <- function(vshare) {
  mean(vshare, na.rm = TRUE) - median(vshare, na.rm = TRUE)
}

wasted_votes <- function(dem, rep) {
  total <- dem + rep
  threshold <- floor(total / 2) + 1

  dem_wasted <- if_else(dem > rep, dem - threshold, dem)
  rep_wasted <- if_else(rep > dem, rep - threshold, rep)

  tibble(dem_wasted, rep_wasted, total)
}

ab604_seats <- ab604_results |>
  count(winner)

ab604_mm <- mean_median_score(ab604_results$dem_share)

w_ab604 <- wasted_votes(ab604_results$dem_votes, ab604_results$rep_votes)

ab604_eg <- (sum(w_ab604$rep_wasted) - sum(w_ab604$dem_wasted)) /
            sum(w_ab604$total)

ab604_summary <- tibble(
  map            = "AB 604",
  dem_seats      = ab604_seats$n[ab604_seats$winner == "DEM"],
  rep_seats      = ab604_seats$n[ab604_seats$winner == "REP"],
  mean_median    = ab604_mm,
  efficiency_gap = ab604_eg
)

ab604_summary
```

```
# A tibble: 1 x 5
  map    dem_seats rep_seats mean_median efficiency_gap
  <chr>      <int>     <int>       <dbl>          <dbl>
1 AB 604        46         7      0.0281          0.179
```

Seats: Under the current map, Democrats win CURRENT_DEM seats and Republicans win CURRENT_REP. Under the AB 604 map, Democrats would win 46 seats and Republicans 7. So AB 604 still gives Democrats most of the seats, but it changes the exact split compared to the current map.

Mean–median: For the current map, the mean–median score is about CURRENT_MM. For the AB 604 map, it is about 0.028. Both numbers are fairly close to zero, so neither map looks extremely skewed by this measure.

Efficiency gap: For the current map, the efficiency gap is about CURRENT_EG. For the AB 604 map, the efficiency gap is about 0.179, which favors Republicans. That means, compared to the current map, AB 604 wastes more Democratic votes and slightly increases the built-in advantage for Republicans.