Gerrymandering Metrics

```r
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```r
vote_prec_clean <- read_csv("data/vote_prec_clean.csv")
```

```
Rows: 48337 Columns: 76
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (49): fips, svprec, svprec_key, election, 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
glimpse(vote_prec_clean)
```

```
Rows: 48,337
Columns: 76
$ county      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,…
$ fips        <chr> "06001", "06001", "06001", "06001", "06001", "06001",
"0600…
$ svprec     <chr> "200100", "200100A", "200200", "200200A", "201400",
"201400…
$ addist     <dbl> 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14,
14,…
$ svprec_key <chr> "06001200100", "06001200100A", "06001200200",
"06001200200A…
$ election   <chr> "g24", "g24", "g24", "g24", "g24", "g24", "g24", "g24",
"g2…
$ geo_type   <chr> "svprec", "svprec", "svprec", "svprec", "svprec",
"svprec",…
$ cddist     <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,
12,…
$ sddist     <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7,…
$ bedist     <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,…
$ totreg     <dbl> 3535, 0, 2442, 0, 3773, 0, 541, 0, 1105, 0, 948, 0, 2721,
0…
$ demreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ repreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ aipreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ grnreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ libreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ nlpreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ refreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ dclreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ mscreg     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ totvote    <dbl> 256, 2804, 262, 1816, 283, 2782, 89, 343, 394, 297, 837,
29…
$ demvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ repvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ aipvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ grnvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ libvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ nlpvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ refvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ dclvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ mscvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ prcvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ absvote    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,…
$ assaip01   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ assdem01   <chr> "94", "444", "117", "348", "107", "588", "45", "105",
"181"…
$ assdem02   <chr> "110", "2023", "91", "1243", "128", "1841", "24", "172",
"1…
$ assrep01   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ assrep02   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ cngdem01   <chr> "102", "1668", "108", "1063", "139", "1688", "35", "192",
"…
$ cngdem02   <chr> "102", "771", "99", "513", "98", "739", "38", "93", "143",
…
$ cngind01   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ cngrep01   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ cngrep02   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ prsaip01   <chr> "3", "10", "1", "6", "3", "13", "2", "2", "3", "4", "5",
"2…
$ prsdem01   <chr> "181", "2562", "207", "1647", "231", "2522", "73", "297",
"…
$ prsgrn01   <chr> "9", "48", "13", "41", "9", "52", "0", "13", "6", "13",
"23…
$ prslib01   <chr> "1", "10", "2", "3", "4", "13", "0", "0", "3", "2", "3",
"0…
$ prspaf01   <chr> "5", "17", "7", "12", "8", "32", "2", "4", "3", "7", "16",
…
$ prsrep01   <chr> "51", "108", "26", "83", "17", "111", "11", "23", "55",
"24…
$ pr_2_n     <chr> "58", "493", "45", "342", "39", "399", "17", "45", "52",
"3…
$ pr_2_y     <chr> "169", "2156", "196", "1385", "226", "2231", "66", "278",
"…
$ pr_32_n    <chr> "78", "636", "55", "439", "74", "536", "14", "60", "81",
"6…
$ pr_32_y    <chr> "148", "1966", "187", "1261", "190", "2070", "68", "255",
"…
$ pr_33_n    <chr> "136", "1774", "105", "1092", "124", "1509", "30", "127",
"…
$ pr_33_y    <chr> "86", "784", "126", "584", "133", "1053", "49", "177",
"231…
$ pr_34_n    <chr> "123", "1485", "121", "1027", "144", "1515", "46", "186",
"…
$ pr_34_y    <chr> "98", "980", "105", "601", "96", "941", "33", "107",
"174",…
$ pr_35_n    <chr> "54", "581", "45", "419", "58", "563", "20", "57", "61",
"5…
$ pr_35_y    <chr> "171", "2003", "188", "1261", "196", "1988", "58", "248",
"…
$ pr_36_n    <chr> "106", "1356", "142", "888", "146", "1487", "49", "197",
"2…
$ pr_36_y    <chr> "118", "1223", "99", "786", "119", "1084", "31", "112",
"14…
$ pr_3_n     <chr> "51", "133", "25", "116", "33", "152", "10", "26", "38",
"2…
$ pr_3_y     <chr> "183", "2553", "220", "1646", "240", "2508", "74", "295",
"…
$ pr_4_n     <chr> "52", "381", "37", "271", "37", "330", "14", "41", "40",
"2…
$ pr_4_y     <chr> "181", "2294", "209", "1472", "231", "2316", "68", "279",
"…
$ pr_5_n     <chr> "94", "961", "66", "605", "63", "742", "19", "76", "72",
"6…
$ pr_5_y     <chr> "132", "1660", "168", "1096", "197", "1862", "61", "240",
"…
$ pr_6_n     <chr> "75", "607", "59", "407", "57", "532", "17", "53", "85",
"5…
$ pr_6_y     <chr> "143", "1958", "180", "1274", "196", "2029", "62", "257",
"…
$ sendem01   <chr> "107", "1719", "101", "1102", "136", "1578", "37", "153",
"…
$ sendem02   <chr> "103", "809", "114", "516", "105", "908", "34", "133",
"174…
$ senrep01   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ senrep02   <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",…
$ uspdem01   <chr> "172", "2461", "199", "1572", "217", "2444", "67", "285",
"…
$ usprep01   <chr> "53", "155", "34", "111", "32", "153", "11", "23", "53",
"3…
$ ussdem01   <chr> "173", "2487", "207", "1593", "222", "2478", "67", "288",
"…
$ ussrep01   <chr> "55", "155", "29", "109", "33", "151", "12", "23", "51",
"2…
```

2024 Election Results and the 2024 District Map

```
vote_prec_use <- vote_prec_clean |>
  filter(!str_detect(svprec, "TOT")) |>
  filter(cddist > 0)


vote_prec_use <- vote_prec_use |>
  mutate(
    cngdem01 = as.numeric(cngdem01),
    cngdem02 = as.numeric(cngdem02),
    cngrep01 = as.numeric(cngrep01),
    cngrep02 = as.numeric(cngrep02),
    cngind01 = as.numeric(cngind01)
  )
```

```
Warning: There were 5 warnings in `mutate()`.
The first warning was:
i In argument: `cngdem01 = as.numeric(cngdem01)`.
Caused by warning:
! NAs introduced by coercion
i Run `dplyr::last_dplyr_warnings()` to see the 4 remaining warnings.
```

```
district_results <- vote_prec_use |>
  group_by(cddist) |>
  summarize(
    dem_total   = sum(cngdem01 + cngdem02, na.rm = TRUE),
    rep_total   = sum(cngrep01 + cngrep02, na.rm = TRUE),
    ind_total   = sum(cngind01, na.rm = TRUE),
    total_votes = sum(totvote, na.rm = TRUE),
    .groups = "drop"
  ) |>
  mutate(
    two_party_total = dem_total + rep_total
  )

district_results
```

```
# A tibble: 52 × 6
   cddist dem_total rep_total ind_total total_votes two_party_total
    <dbl>     <dbl>     <dbl>     <dbl>       <dbl>           <dbl>
 1      1     96097    173271         0      280075          269368
 2      2    265967    102467         0      386639          368434
 3      3    164310    209904         0      393009          374214
 4      4    213615    105012         0      333223          318627
 5      5    121038    190015         0      327223          311053
 6      6    164133    120760         0      304139          284893
 7      7    188770     93100         0      300562          281870
 8      8    185667     64141         0      265502          249808
 9      9    130093    121006         0      262106          251099
10     10    241926    121826         0      387335          363752
# ℹ 42 more rows
```

```
district_results |>
  filter(dem_total == 0 | rep_total == 0)
```

```
# A tibble: 5 × 6
  cddist dem_total rep_total ind_total total_votes two_party_total
   <dbl>     <dbl>     <dbl>     <dbl>       <dbl>           <dbl>
1     12    282974         0         0      324446          282974
2     16    298322         0         0      340185          298322
3     20         0    282634         0      326235          282634
4     34    189408         0         0      218653          189408
5     37    160361         0     44450      238930          160361
```

```
district_two_party <- district_results |>
  filter(dem_total > 0, rep_total > 0)

district_two_party <- district_two_party |>
  mutate(
    dem_share = dem_total / two_party_total
  )

mean_dem_share   <- mean(district_two_party$dem_share)
median_dem_share <- median(district_two_party$dem_share)
mean_median_score <- mean_dem_share - median_dem_share

mean_dem_share
```

```
[1] 0.5971725
```

```
median_dem_share
```

```
[1] 0.6113796
```

```
mean_median_score
```

```
[1] -0.01420717
```

```
wasted_votes <- function(votes_A, votes_B) {
  total  <- votes_A + votes_B
  needed <- floor(total / 2) + 1

  ifelse(
    votes_A > votes_B,
    votes_A - needed,
    votes_A
  )
}

district_two_party <- district_two_party |>
  mutate(
    wasted_dem = wasted_votes(dem_total, rep_total),
    wasted_rep = wasted_votes(rep_total, dem_total)
  )

total_wasted_dem <- sum(district_two_party$wasted_dem)
total_wasted_rep <- sum(district_two_party$wasted_rep)
total_two_party_votes <- sum(district_two_party$two_party_total)

efficiency_gap <- (total_wasted_rep - total_wasted_dem) /
total_two_party_votes

total_wasted_dem
```

```
[1] 2477655
```

```
total_wasted_rep
```

```
[1] 4157887
```

```
efficiency_gap
```

```
[1] 0.1266078
```

2024 Election Results and the proposed 2025 District Map

```r
library(sf)
```

```
Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```r
sr_sov <- 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
names(sr_sov)
```

```
 [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
dem_cols <- c("CNGDEM01", "CNGDEM02")
rep_cols <- c("CNGREP01", "CNGREP02")

sr_votes <- sr_sov |>
  mutate(
    across(
      all_of(c(dem_cols, rep_cols)),
      ~ as.numeric(gsub(",", "", .x))
    )
  ) |>
  mutate(
    dem_total = rowSums(across(all_of(dem_cols)), na.rm = TRUE),
    rep_total = rowSums(across(all_of(rep_cols)), na.rm = TRUE)
  ) |>
  select(SRPREC, dem_total, rep_total)
```

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

```r
sr_shp <- st_read("data/shapefiles/srprec_state_g24_v01_shp/
srprec_state_g24_v01_shp.shp")
```

```
Reading layer `srprec_state_g24_v01_shp' from data source
  `/Users/hibahalam/Desktop/stat133/gerrymandering-hibahalam/data/shapefiles/
srprec_state_g24_v01_shp/srprec_state_g24_v01_shp.shp'
  using driver `ESRI Shapefile'
```

```
Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
GDAL
Message 1:
/Users/hibahalam/Desktop/stat133/gerrymandering-hibahalam/data/shapefiles/
srprec_state_g24_v01_shp/srprec_state_g24_v01_shp.shp
contains polygon(s) with rings with invalid winding order. Autocorrecting
them,
but that shapefile should be corrected using ogr2ogr for example.
```

```
Simple feature collection with 24224 features and 6 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
```

```
Bounding box:  xmin: -124.482 ymin: 32.52883 xmax: -114.1312 ymax: 42.0095
Geodetic CRS:  NAD83
```

```r
names(sr_shp)
```

```
[1] "SRPREC"     "COUNTY"     "ELECTION"   "PREC_TYPE"  "FIPS_CODE"
[6] "SRPREC_KEY" "geometry"
```

```r
sr_shp <- sr_shp |>
  st_transform(3310) |>
  st_set_precision(1) |>
  st_make_valid() |>
  st_collection_extract("POLYGON")
```

```r
sr_geo <- sr_shp |>
  left_join(sr_votes, by = "SRPREC")
```

```
Warning in sf_column %in% names(g): Detected an unexpected many-to-many
relationship between `x` and `y`.
i Row 11 of `x` matches multiple rows in `y`.
i Row 376 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
ab604 <- st_read("data/shapefiles/AB604/AB604.shp")
```

```
Reading layer `AB604' from data source
  `/Users/hibahalam/Desktop/stat133/gerrymandering-hibahalam/data/shapefiles/
AB604/AB604.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 52 features and 15 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -13857270 ymin: 3832931 xmax: -12705030 ymax: 5162404
Projected CRS: WGS 84 / Pseudo-Mercator
```

```r
names(ab604)
```

```
 [1] "DISTRICT"   "A_POP"      "DEVIATION"  "CVAP"       "HSP_CVAP"
 [6] "IND_CVAP"   "BLK_CVAP"   "ASN_CVAP"   "WHT_CVAP"   "CVAP_PCT"
```

```
[11] "HSP_CVAP_P" "IND_CVAP_P" "BLK_CVAP_P" "ASN_CVAP_P" "WHT_CVAP_P"
[16] "geometry"
```

```r
ab604 <- ab604 |>
  rename(
    new_cddist = DISTRICT
  ) |>
  st_transform(3310) |>
  st_make_valid()
```

```r
sr_ab_int <- st_intersection(sr_geo, ab604)
```

```
Warning: attribute variables are assumed to be spatially constant throughout
all geometries
```

```r
sr_ab_int <- sr_ab_int |>
  mutate(
    area_part = st_area(geometry)
  )

sr_areas <- sr_ab_int |>
  st_drop_geometry() |>
  group_by(SRPREC) |>
  summarize(
    area_sr = sum(as.numeric(area_part), na.rm = TRUE),
    .groups = "drop"
  )

sr_ab_int <- sr_ab_int |>
  left_join(sr_areas, by = "SRPREC") |>
  mutate(
    area_part_num = as.numeric(area_part),
    area_fraction = if_else(area_sr > 0,
                            area_part_num / area_sr,
                            0)
  )
```

```r
sr_ab_votes <- sr_ab_int |>
  st_drop_geometry() |>
  mutate(
    dem_alloc = dem_total * area_fraction,
    rep_alloc = rep_total * area_fraction
  )
```

```r
ab604_results <- sr_ab_votes |>
  group_by(new_cddist) |>
  summarize(
    dem_total_new = sum(dem_alloc, na.rm = TRUE),
    rep_total_new = sum(rep_alloc, na.rm = TRUE),
    .groups = "drop"
  ) |>
  mutate(
    two_party_total_new = dem_total_new + rep_total_new,
    dem_share_new       = dem_total_new / two_party_total_new
  )

ab604_results
```

```
# A tibble: 52 × 5
   new_cddist dem_total_new rep_total_new two_party_total_new dem_share_new
   <chr>              <dbl>         <dbl>               <dbl>         <dbl>
 1 01               172906.       135493.             308399.         0.561
 2 02               220393.       140519.             360912.         0.611
 3 03               177004.       154019.             331023.         0.535
 4 04               183427.       122407.             305834.         0.600
 5 05               111084.       192629.             303713.         0.366
 6 06               166713.       148287.             315000.         0.529
 7 07               188341.       143708.             332049.         0.567
 8 08               191836.        88974.             280810.         0.683
 9 09               145746.       103143.             248890.         0.586
10 10               235405.       120320.             355725.         0.662
# ℹ 42 more rows
```

```r
write_csv(ab604_results, "data/new_results_area_weighted.csv")
```

```r
ab604_two_party <- ab604_results |>
  filter(dem_total_new > 0, rep_total_new > 0)

ab604_two_party <- ab604_two_party |>
  mutate(
    dem_share_new = dem_total_new / (dem_total_new + rep_total_new)
  )

mean_dem_share_new   <- mean(ab604_two_party$dem_share_new)
median_dem_share_new <- median(ab604_two_party$dem_share_new)
mean_median_score_new <- mean_dem_share_new - median_dem_share_new

mean_dem_share_new
```

```
[1] 0.6155237
```

```
median_dem_share_new
```

```
[1] 0.5884477
```

```
mean_median_score_new
```

```
[1] 0.02707599
```

```
ab604_two_party <- ab604_two_party |>
  mutate(
    wasted_dem_new = wasted_votes(dem_total_new, rep_total_new),
    wasted_rep_new = wasted_votes(rep_total_new, dem_total_new)
  )

total_wasted_dem_new <- sum(ab604_two_party$wasted_dem_new)
total_wasted_rep_new <- sum(ab604_two_party$wasted_rep_new)
total_two_party_votes_new <- sum(ab604_two_party$dem_total_new +
ab604_two_party$rep_total_new)

efficiency_gap_new <- (total_wasted_rep_new - total_wasted_dem_new) /
total_two_party_votes_new

total_wasted_dem_new
```

```
[1] 2358345
```

```
total_wasted_rep_new
```

```
[1] 4917607
```

```
efficiency_gap_new
```

```
[1] 0.1758706
```