

# Data Cleaning

## Introduction

In this document, I clean the **2024 General Election Statement of Vote (SOV)** data at the SV (split-voting) precinct level and prepare additional datasets for re-running the 2024 election under a new congressional district map.

The main goals are to create:

- A cleaned SV-precinct SOV file for exploratory analysis and gerrymandering metrics.
- A cleaned SR-precinct SOV + geometry dataset for **Approach A (area-weighted interpolation)** used to simulate the 2024 election under the **AB 604** proposed map.

Original files used here:

- `data/g24 Sov_by_g24_svpref.csv` – SV precinct SOV (2024 General).
- `data/shapefiles/state_g24 Sov_data_by_g24_srprec.csv` – SR precinct SOV (2024 General).
- Shapefiles for SR precincts: `data/shapefiles/srprec_state_g24_v01_shp/srprec_state_g24_v01_shp.shp` (+ .dbf, .shx, .prj, .cpg).
- Shapefiles for proposed AB 604 congressional districts: `data/shapefiles/AB604/AB604.shp` (+ .dbf, .shx, .prj).

Cleaned output files from this document:

- `data/clean Sov.csv` – cleaned SV precinct SOV.
- `data/newmap-district-votes.csv` – district-level vote totals under AB 604 (via Approach A).

## Setup

```
library(tidyverse)
library(readxl)
library(janitor)
library(sf)
```

## Part 2: Cleaning SV Precinct SOV Data

### Read and Inspect Raw Data

```
# Read in the raw precinct-level statement of vote (SV precincts)
sov_raw <- read_csv("data/g24 Sov_by_g24_svpref.csv",
                      show_col_types = FALSE)
```

```
# Quick look at the structure  
glimpse(sov_raw)
```

```

$ DEMVOTE      <dbl> 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, ...
$ AIPVOTE      <dbl> 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, ...
$ LIBVOTE      <dbl> 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, ...
$ REFVOTE      <dbl> 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, ...
$ MSCVOTE      <dbl> 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, ...
$ ABSVOTE      <dbl> 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", ...
$ 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", ...
$ ASSREP02     <chr> "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", ...
$ CNGREP01     <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ CNGREP02     <chr> "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", ...
$ SENREP02 <chr> "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...

```

```

# Peek at the first few rows
head(sov_raw)

```

```

# A tibble: 6 × 76
  COUNTY FIPS SVPREC ADDIST SVPREC_KEY ELECTION GEO_TYPE CDDIST SDDIST
  BEDIST
    <dbl> <chr> <chr>     <dbl> <chr>      <chr>   <chr>   <dbl> <dbl>
1       1 06001 200100     14 06001200100 g24    svprec    12     7
2
2       1 06001 200100A    14 0600120010... g24    svprec    12     7
2
3       1 06001 200200     14 06001200200 g24    svprec    12     7
2
4       1 06001 200200A    14 0600120020... g24    svprec    12     7
2
5       1 06001 201400     14 06001201400 g24    svprec    12     7
2
6       1 06001 201400A    14 0600120140... g24    svprec    12     7
2
# i 66 more variables: TOTREG <dbl>, DEMREG <dbl>, REPREG <dbl>, AIPREG <dbl>,
# GRNREG <dbl>, LIBREG <dbl>, NLPREG <dbl>, REFREG <dbl>, DCLREG <dbl>,
# MSCREG <dbl>, TOTVOTE <dbl>, DEMVOTE <dbl>, REPVOTE <dbl>, AIPVOTE <dbl>,
# GRNVOTE <dbl>, LIBVOTE <dbl>, NLPVOTE <dbl>, REFPOTE <dbl>, DCLVOTE <dbl>,
# MSCVOTE <dbl>, PRCVOTE <dbl>, ABSVOTE <dbl>, ASSAIP01 <chr>,
# ASSDEM01 <chr>, ASSDEM02 <chr>, ASSREP01 <chr>, ASSREP02 <chr>,
# CNGDEM01 <chr>, CNGDEM02 <chr>, CNGIND01 <chr>, CNGREP01 <chr>, ...

```

## Clean Column Names

```
# Make all column names lower_snake_case and easier to work with
sov <- sov_raw |>
  clean_names()
```

```
# Check the cleaned names
names(sov)
```

```
[1] "county"      "fips"        "svprec"       "addist"       "svprec_key"
[6] "election"    "geo_type"     "cddist"       "sddist"       "bedist"
[11] "totreg"      "demreg"      "repreg"      "aipreg"      "grnreg"
[16] "libreg"      "nlpreg"      "refreg"      "dclreg"      "mscereg"
[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"
```

## Fix Column Types (Vote Columns)

The SOV file sometimes stores vote totals as character because of non-numeric characters (commas, blanks, etc.). Here I identify vote columns and coerce them to numeric using `parse_number()`.

```
# Identify columns that look like vote counts for major contests
vote_cols <- names(sov)[grepl("cng|sen|usp|yes|nos", names(sov))]
```

```
vote_cols
```

```
[1] "cngdem01" "cngdem02" "cngind01" "cngrep01" "cngrep02" "sendem01"
[7] "sendem02" "senrep01" "senrep02" "uspdem01" "usprep01"
```

```
# Convert vote columns to numeric, stripping out any stray characters
sov <- sov |>
  mutate(across(all_of(vote_cols), ~ parse_number(.x)))
```

```
# Sanity check: summary of a few vote columns
sov |>
```

```
select(all_of(vote_cols)) |>
  summary()
```

	cngdem01	cngdem02	cngind01	cngrep01
Min.	: 0	Min. : 0.00	Min. : 0.0	Min. : 0.0
1st Qu.	: 0	1st Qu.: 0.00	1st Qu.: 0.0	1st Qu.: 0.0
Median	: 48	Median : 0.00	Median : 0.0	Median : 45.0
Mean	: 1327	Mean : 46.84	Mean : 6.7	Mean : 860.2
3rd Qu.	: 260	3rd Qu.: 0.00	3rd Qu.: 0.0	3rd Qu.: 169.5
Max.	: 2273160	Max. : 106751.00	Max. : 44450.0	Max. : 1050936.0
NA's	: 4684	NA's : 4684	NA's : 4684	NA's : 4684
	cngrep02	sendem01	sendem02	senrep01
Min.	: 0.00	Min. : 0.0	Min. : 0.00	Min. : 0.0
1st Qu.	: 0.00	1st Qu.: 0.0	1st Qu.: 0.00	1st Qu.: 0.0
Median	: 0.00	Median : 0.0	Median : 0.00	Median : 0.0
Mean	: 28.31	Mean : 647.5	Mean : 67.98	Mean : 409.9
3rd Qu.	: 0.00	3rd Qu.: 47.0	3rd Qu.: 0.00	3rd Qu.: 39.0
Max.	: 110314.00	Max. : 884250.0	Max. : 181838.00	Max. : 476150.0
NA's	: 4684	NA's : 4684	NA's : 4684	NA's : 4684
	senrep02	uspdem01	usprep01	
Min.	: 0.00	Min. : 0	Min. : 0.0	
1st Qu.	: 0.00	1st Qu.: 0	1st Qu.: 0.0	
Median	: 0.00	Median : 47	Median : 53.0	
Mean	: 13.31	Mean : 1329	Mean : 930.2	
3rd Qu.	: 0.00	3rd Qu.: 256	3rd Qu.: 188.0	
Max.	: 15989.00	Max. : 2323607	Max. : 1233740.0	
NA's	: 4684	NA's : 4684	NA's : 4684	

## Check Key Columns and Basic Integrity

Here I run a few basic checks:

- svprec\_key should uniquely identify each row.
- cddist (congressional district) should be in a reasonable range.
- Registration totals (if present) should be non-negative.

```
# Check uniqueness of the split-precinct key
any(duplicated(sov$svprec_key)) # Should be FALSE
```

```
[1] FALSE
```

```
sum(duplicated(sov$svprec_key))
```

```
[1] 0
```

```
# Check congressional district values (if present)
if ("cddist" %in% names(sov)) {
  summary(sov$cddist)
  unique(sort(sov$cddist))
}
```

```
[1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24
[26] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49
[51] 50 51 52
```

```
# If a registration column exists (e.g., totreg), check basic properties
if ("totreg" %in% names(sov)) {
  summary(sov$totreg)
  sum(sov$totreg < 0, na.rm = TRUE)
}
```

```
[1] 0
```

Optionally, we can check that candidate vote totals do not exceed total registration in obviously impossible ways (this is a rough sanity check, not a hard rule, since registration and turnout differ):

```
if (all(c("cngdem01", "cngrep01", "totreg") %in% names(sov))) {
  sov |>
    mutate(
      total_cng_votes = cngdem01 + cngrep01
    ) |>
    summarise(
      max_votes = max(total_cng_votes, na.rm = TRUE),
      max_reg   = max(totreg, na.rm = TRUE)
    )
}
```

```
# A tibble: 1 × 2
  max_votes max_reg
  <dbl>     <dbl>
1 3324096  5745214
```

## Add Convenience Variables and Save Cleaned SV Precinct Data

At this point:

- Column names are standardized.

- Vote columns have been coerced to numeric.
- Basic integrity checks have been performed.

I also create a couple of convenience variables for later use:

- `total_votes` – two-party total (Dem + Rep) for the main congressional race.
- `margin` – Democratic minus Republican votes in that race.

```
sov <- sov |>
  mutate(
    total_votes = cngdem01 + cngrep01,
    margin = cngdem01 - cngrep01
  )

# Save cleaned SV-precinct SOV
write_csv(sov, "data/clean-sov.csv")
```

The file `data/clean-sov.csv` will be the cleaned dataset used in:

- `exploratory-data-analysis.qmd`
- `gerrymandering-metrics.qmd`
- As a reference for totals in later parts of the project.

## Part 5: Re-running the 2024 Election (Approach A: Area-Weighted Interpolation)

In this section, I prepare the data needed to simulate the 2024 election under the proposed AB 604 congressional map using **area-weighted interpolation**.

The basic idea:

1. Work at the **SR precinct** level (geographic precincts).
2. Attach SOV vote totals to SR precinct geometries.
3. Overlay SR precincts with the proposed AB 604 district polygons.
4. For each overlapping piece, compute the fraction of the SR precinct area that lies in each new district.
5. Allocate votes proportionally to that area share.
6. Sum allocated votes within each new district.

### Read SR Precinct Vote Data

```
# Read SR precinct-level SOV data (2024 General)
sr_votes_raw <- read_csv("data/shapefiles/
state_g24 Sov_data_by_g24_srprec.csv",
                           show_col_types = FALSE)

glimpse(sr_votes_raw)
```



```

$ AIPVOTE      <dbl> 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, ...
$ LIBVOTE       <dbl> 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, ...
$ REFVOTE       <dbl> 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, ...
$ MSCVOTE       <dbl> 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, ...
$ ABSVOTE       <dbl> 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", ...
$ ASSDEM01      <chr> "538", "465", "695", "150", "271", "560", "343", "411",
"47...
$ ASSDEM02      <chr> "2133", "1334", "1969", "196", "253", "358", "1702",
"1020"...
$ ASSREP01      <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ ASSREP02      <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ CNGDEM01      <chr> "1770", "1171", "1827", "227", "322", "519", "1456",
"992", ...
$ CNGDEM02      <chr> "873", "612", "837", "131", "227", "447", "573", "417",
"44...
$ CNGIND01      <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ CNGREP01      <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ CNGREP02      <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0", ...
$ PRSAIP01      <chr> "13", "7", "16", "4", "7", "7", "9", "6", "6",
"0", "0",
"8...
$ PRSDEM01      <chr> "2743", "1854", "2753", "370", "558", "940", "2135",
"1492"...
$ PRSGRN01      <chr> "57", "54", "61", "13", "19", "30", "30", "39",
"51", "2",
"...
$ PRSLIB01      <chr> "11", "5", "17", "0", "5", "3", "4", "5", "6",
"0", "0",
"5...
$ PRSPAF01      <chr> "22", "19", "40", "6", "10", "22", "10", "21",
"26", "0",
"...

```

```

$ PRSREP01 <chr> "159", "109", "128", "34", "79", "113", "96", "102", "50",
...
$ PR_2_N <chr> "551", "387", "438", "62", "90", "103", "382", "211",
"187"...
$ PR_2_Y <chr> "2325", "1581", "2457", "344", "539", "938", "1800",
"1350"...
$ PR_32_N <chr> "714", "494", "610", "74", "148", "204", "536", "339",
"290"...
$ PR_32_Y <chr> "2114", "1448", "2260", "323", "468", "811", "1606",
"1207"...
$ PR_33_N <chr> "1910", "1197", "1633", "157", "212", "350", "1438",
"830"...
$ PR_33_Y <chr> "870", "710", "1186", "226", "389", "630", "671", "678",
"9"...
$ PR_34_N <chr> "1608", "1148", "1659", "232", "307", "459", "1340",
"935"...
$ PR_34_Y <chr> "1078", "706", "1037", "140", "280", "513", "696", "509",
"..."...
$ PR_35_N <chr> "635", "464", "621", "77", "111", "118", "600", "359",
"371"...
$ PR_35_Y <chr> "2174", "1449", "2184", "306", "502", "906", "1481",
"1132"...
$ PR_36_N <chr> "1462", "1030", "1633", "246", "381", "591", "1295",
"984"...
$ PR_36_Y <chr> "1341", "885", "1203", "143", "239", "426", "843", "541",
"..."...
$ PR_3_N <chr> "184", "141", "185", "36", "58", "96", "111", "127", "78",
...
$ PR_3_Y <chr> "2736", "1866", "2748", "369", "590", "963", "2082",
"1470"...
$ PR_4_N <chr> "433", "308", "367", "55", "64", "89", "351", "170",
"189"...
$ PR_4_Y <chr> "2475", "1681", "2547", "347", "572", "962", "1827",
"1410"...
$ PR_5_N <chr> "1055", "671", "805", "95", "134", "178", "684", "384",
"37"...
$ PR_5_Y <chr> "1792", "1264", "2059", "301", "494", "848", "1478",
"1159"...
$ PR_6_N <chr> "682", "466", "589", "70", "143", "237", "491", "313",
"241"...
$ PR_6_Y <chr> "2101", "1454", "2225", "319", "466", "757", "1631",
"1208"...
$ SENDEM01 <chr> "1826", "1203", "1714", "190", "239", "418", "1517",
"866"...
$ SENDEM02 <chr> "912", "630", "1013", "167", "295", "521", "547", "590",
"7"...
$ 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", "0", ...
$ USPDEM01 <chr> "2633", "1771", "2661", "352", "525", "902", "2051", "1409"...
$ USPREP01 <chr> "208", "145", "185", "34", "85", "109", "135", "131", "72"...
$ USSDEM01 <chr> "2660", "1800", "2700", "355", "537", "910", "2082", "1433"...
$ USSREP01 <chr> "210", "138", "184", "35", "80", "104", "132", "134", "64"...
```

```
# Clean column names
sr_votes <- sr_votes_raw |>
  clean_names()

names(sr_votes)
```

```
[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"
```

As with the SV data, the SR SOV may store vote totals as character. We apply a similar numeric conversion.

```
# Identify vote columns in the SR SOV
sr_vote_cols <- names(sr_votes)[grepl("^(cng|sen|usp|yes$|no$",
  names(sr_votes))]

sr_vote_cols
```

```
[1] "cngdem01" "cngdem02" "cngind01" "cngrep01" "cngrep02" "sendem01"
[7] "sendem02" "senrep01" "senrep02" "uspdem01" "usprep01"
```

```

# Convert SR vote columns to numeric
sr_votes <- sr_votes |>
  mutate(across(all_of(sr_vote_cols), ~ parse_number(.x)))

# Quick sanity check
sr_votes |>
  select(all_of(sr_vote_cols)) |>
  summary()

```

	cngdem01	cngdem02	cngind01	cngrep01
Min.	: 0	Min. : 0.00	Min. : 0.000	Min. : 0.0
1st Qu.	: 1	1st Qu.: 0.00	1st Qu.: 0.000	1st Qu.: 0.0
Median	: 193	Median : 0.00	Median : 0.000	Median : 121.0
Mean	: 379	Mean : 13.34	Mean : 1.909	Mean : 246.3
3rd Qu.	: 601	3rd Qu.: 0.00	3rd Qu.: 0.000	3rd Qu.: 372.0
Max.	: 5162	Max. : 1814.00	Max. : 1198.000	Max. : 4626.0
NA's	: 1958	NA's : 1958	NA's : 1958	NA's : 1958
	cngrep02	sendem01	sendem02	senrep01
Min.	: 0.000	Min. : 0.0	Min. : 0.00	Min. : 0.0
1st Qu.	: 0.000	1st Qu.: 0.0	1st Qu.: 0.00	1st Qu.: 0.0
Median	: 0.000	Median : 0.0	Median : 0.00	Median : 0.0
Mean	: 8.063	Mean : 184.4	Mean : 19.36	Mean : 116.7
3rd Qu.	: 0.000	3rd Qu.: 192.0	3rd Qu.: 0.00	3rd Qu.: 109.0
Max.	: 3574.000	Max. : 5082.0	Max. : 2049.00	Max. : 2895.0
NA's	: 1958	NA's : 1958	NA's : 1958	NA's : 1958
	senrep02	uspdem01	usprep01	
Min.	: 0.000	Min. : 0.0	Min. : 0.0	
1st Qu.	: 0.000	1st Qu.: 6.0	1st Qu.: 8.0	
Median	: 0.000	Median : 196.0	Median : 147.0	
Mean	: 3.791	Mean : 379.4	Mean : 266.3	
3rd Qu.	: 0.000	3rd Qu.: 584.0	3rd Qu.: 404.0	
Max.	: 818.000	Max. : 5146.0	Max. : 4553.0	
NA's	: 1958	NA's : 1958	NA's : 1958	

## Load SR Precinct Shapefiles and Fix Geometry

```

# Read SR precinct shapefile
sr_shp_raw <- st_read("data/shapefiles/srprec_state_g24_v01_shp /"
  "srprec_state_g24_v01_shp.shp",
  quiet = TRUE) |>
  clean_names()

sr_shp_raw

```

```

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
First 10 features:
  srprec county election prec_type fips_code  srprec_key
1 491650    001        g24    SRPREC      06001 06001491650
2 205200    001        g24    SRPREC      06001 06001205200
3 547720    001        g24    SRPREC      06001 06001547720
4 833300    001        g24    SRPREC      06001 06001833300
5 333410    001        g24    SRPREC      06001 06001333410
6 492110    001        g24    SRPREC      06001 06001492110
7 830410    001        g24    SRPREC      06001 06001830410
8 815140    001        g24    SRPREC      06001 06001815140
9 443720    001        g24    SRPREC      06001 06001443720
10 364600   001        g24    SRPREC     06001 06001364600
                           geometry
1 MULTIPOINT ((-122.0288 3...
2 MULTIPOINT ((-122.2828 3...
3 MULTIPOINT ((-121.8779 3...
4 MULTIPOINT ((-121.963 37...
5 MULTIPOINT ((-122.2673 3...
6 MULTIPOINT ((-122.0243 3...
7 MULTIPOINT ((-122.0914 3...
8 MULTIPOINT ((-122.017 37...
9 MULTIPOINT ((-122.124 37...
10 MULTIPOINT ((-122.1711 3...

```

Apply the geometry fixes recommended in the assignment:

```

sr_shp <- sr_shp_raw |>
  st_transform(3310) |>      # equal-area projection
  st_set_precision(1) |>      # snap to 1 m grid
  st_make_valid() |>          # fix bow-ties/self-intersections
  st_collection_extract("POLYGON")

sr_shp

```

```

Simple feature collection with 24862 features and 6 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:  xmin: -380101 ymin: -605327 xmax: 540038 ymax: 450447
Projected CRS: NAD83 / California Albers
First 10 features:
  srprec county election prec_type fips_code  srprec_key
1 491650    001        g24    SRPREC      06001 06001491650
2 205200    001        g24    SRPREC      06001 06001205200
3 547720    001        g24    SRPREC      06001 06001547720

```

```

4 833300 001      g24    SRPREC    06001 06001833300
5 333410 001      g24    SRPREC    06001 06001333410
6 492110 001      g24    SRPREC    06001 06001492110
7 830410 001      g24    SRPREC    06001 06001830410
8 815140 001      g24    SRPREC    06001 06001815140
9 443720 001      g24    SRPREC    06001 06001443720
10 364600 001     geometry
1  MULTIPOLYGON (((-178716 -37...
2  MULTIPOLYGON (((-200489 -11...
3  MULTIPOLYGON (((-165308 -32...
4  MULTIPOLYGON (((-174823 -53...
5  MULTIPOLYGON (((-199826.3 -...
6  MULTIPOLYGON (((-178651 -35...
7  MULTIPOLYGON (((-184546 -51...
8  MULTIPOLYGON (((-177775 -34...
9  MULTIPOLYGON (((-186959 -31...
10 MULTIPOLYGON (((-190969 -25...

```

## Join SR Votes to SR Geometry

Both the SOV and the shapefile share an SR precinct key (`SRPREC_KEY` in the raw data, which becomes `srprec_key` after `clean_names()`).

```
# Inspect potential join keys
names(sr_shp)
```

```
[1] "srprec"      "county"      "election"    "prec_type"   "fips_code"
[6] "srprec_key"  "geometry"
```

```
names(sr_votes)
```

```
[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"
```

```
# Join SR vote data to SR geometries using srprec_key
sr <- sr_shp |>
  left_join(sr_votes, by = "srprec_key")

# Sanity check: did we lose observations?
nrow(sr_shp)
```

[1] 24862

`nrow(sr)`

[1] 24862

## **Load Proposed AB 604 Congressional Map**

```
# Read AB 604 congressional district shapefile
new_map_raw <- st_read("data/shapefiles/AB604/AB604.shp",
                        quiet = TRUE) |>
  clean_names()

new_map_raw
```

```
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
First 10 features:
  district  a_pop deviation    cvap hsp_cvap ind_cvap blk_cvap asn_cvap
1        42 760067           1 547320  134603     4347   36075   69836
2        40 760066           0 543973  152392     4037   28337   63206
3        49 760067           1 525988   96790     4217   16308   67875
4        47 760065          -1 514402   78502     2812   13187  130254
5        23 760066           0 514103  190014     6566   46719   22557
6        48 760066           0 518620  166118     6486   27573   43349
7        25 760066           0 494546  257693     5998   22647   17043
8        24 760065          -1 532407  150929     5447   11614   30697
9        52 760066           0 490770  254254     2185   43342   73711
10       51 760067           1 573012  132681     4721   44078   53569
```

```

wht_cvap cvap_pct hsp_cvap_p ind_cvap_p blk_cvap_p asn_cvap_p wht_cvap_p
1 295693 0.720094 0.245931 0.007942 0.065912 0.127596 0.540256
2 291303 0.715692 0.280146 0.007421 0.052093 0.116193 0.535510
3 336391 0.692028 0.184016 0.008017 0.031005 0.129043 0.639541
4 284408 0.676787 0.152608 0.005467 0.025636 0.253214 0.552891
5 242849 0.676393 0.369603 0.012772 0.090875 0.043876 0.472374
6 268028 0.682335 0.320308 0.012506 0.053166 0.083585 0.516810
7 187426 0.650662 0.521070 0.012128 0.045794 0.034462 0.378986
8 330619 0.700476 0.283484 0.010231 0.021814 0.057657 0.620989
9 110215 0.645694 0.518072 0.004452 0.088314 0.150195 0.224576
10 330267 0.753897 0.231550 0.008239 0.076923 0.093487 0.576370

geometry
1 MULTIPOLYGON (((-13125196 3...
2 MULTIPOLYGON (((-13099202 3...
3 MULTIPOLYGON (((-13096118 3...
4 MULTIPOLYGON (((-13117436 3...
5 MULTIPOLYGON (((-13029958 3...
6 MULTIPOLYGON (((-12984754 3...
7 MULTIPOLYGON (((-12987691 3...
8 MULTIPOLYGON (((-13294684 3...
9 MULTIPOLYGON (((-12951813 3...
10 MULTIPOLYGON (((-12962898 3...

```

```
names(new_map_raw)
```

```
[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"
```

Reproject to the same CRS as the SR layer (EPSG:3310):

```
new_map <- new_map_raw |>
  st_transform(3310)

st_crs(sr)
```

```
Coordinate Reference System:
  User input: EPSG:3310
  wkt:
PROJCRS["NAD83 / California Albers",
  BASEGEOGCRS["NAD83",
    DATUM["North American Datum 1983",
      ELLIPSOID["GRS 1980",6378137,298.257222101,
        LENGTHUNIT["metre",1]]],
```

```

PRIMEM["Greenwich",0,
    ANGLEUNIT["degree",0.0174532925199433],
    ID["EPSG",4269]],
CONVERSION["California Albers",
    METHOD["Albers Equal Area",
        ID["EPSG",9822]],
    PARAMETER["Latitude of false origin",0,
        ANGLEUNIT["degree",0.0174532925199433],
        ID["EPSG",8821]],
    PARAMETER["Longitude of false origin",-120,
        ANGLEUNIT["degree",0.0174532925199433],
        ID["EPSG",8822]],
    PARAMETER["Latitude of 1st standard parallel",34,
        ANGLEUNIT["degree",0.0174532925199433],
        ID["EPSG",8823]],
    PARAMETER["Latitude of 2nd standard parallel",40.5,
        ANGLEUNIT["degree",0.0174532925199433],
        ID["EPSG",8824]],
    PARAMETER["Easting at false origin",0,
        LENGTHUNIT["metre",1],
        ID["EPSG",8826]],
    PARAMETER["Northing at false origin",-4000000,
        LENGTHUNIT["metre",1],
        ID["EPSG",8827]]],
CS[Cartesian,2],
    AXIS["easting (X)",east,
        ORDER[1],
        LENGTHUNIT["metre",1]],
    AXIS["northing (Y)",north,
        ORDER[2],
        LENGTHUNIT["metre",1]],
USAGE[
    SCOPE["State-wide spatial data management."],
    AREA["United States (USA) - California."],
    BBOX[32.53,-124.45,42.01,-114.12]],
    ID["EPSG",3310]]

```

`st_crs(new_map)`

```

Coordinate Reference System:
User input: EPSG:3310
wkt:
PROJCRS["NAD83 / California Albers",
    BASEGEOGCRS["NAD83",
        DATUM["North American Datum 1983",
            ELLIPSOID["GRS 1980",6378137,298.257222101,

```

```

        LENGTHUNIT["metre",1]],
PRIMEM["Greenwich",0,
       ANGLEUNIT["degree",0.0174532925199433]],
ID["EPSG",4269]],
CONVERSION["California Albers",
METHOD["Albers Equal Area",
       ID["EPSG",9822]],
PARAMETER["Latitude of false origin",0,
          ANGLEUNIT["degree",0.0174532925199433],
          ID["EPSG",8821]],
PARAMETER["Longitude of false origin",-120,
          ANGLEUNIT["degree",0.0174532925199433],
          ID["EPSG",8822]],
PARAMETER["Latitude of 1st standard parallel",34,
          ANGLEUNIT["degree",0.0174532925199433],
          ID["EPSG",8823]],
PARAMETER["Latitude of 2nd standard parallel",40.5,
          ANGLEUNIT["degree",0.0174532925199433],
          ID["EPSG",8824]],
PARAMETER["Easting at false origin",0,
          LENGTHUNIT["metre",1],
          ID["EPSG",8826]],
PARAMETER["Northing at false origin",-4000000,
          LENGTHUNIT["metre",1],
          ID["EPSG",8827]]],
CS[Cartesian,2],
AXIS["easting (X)",east,
      ORDER[1],
      LENGTHUNIT["metre",1]],
AXIS["northing (Y)",north,
      ORDER[2],
      LENGTHUNIT["metre",1]],
USAGE[
SCOPE["State-wide spatial data management."],
AREA["United States (USA) - California."],
BBOX[32.53,-124.45,42.01,-114.12]],
ID["EPSG",3310]]

```

You should see matching CRS for both objects.

## Compute Area-Weighted Intersections

```

# Add SR precinct area column (in projected CRS)
sr_with_area <- sr |>
  mutate(sr_area = st_area(geometry))

# Intersect SR precincts with new districts
sr_new_intersection <- st_intersection(

```

```

    sr_with_area,
  new_map |>
  select(new_cdd = district)
)

sr_new_intersection

```

Simple feature collection with 29757 features and 83 fields

Geometry type: GEOMETRY

Dimension: XY

Bounding box: xmin: -380101 ymin: -605326.4 xmax: 540036.5 ymax: 450447

Projected CRS: NAD83 / California Albers

First 10 features:

	srprec.x	county.x	election.x	prec_type	fips_code	srprec_key	county.y
4309	3850423A	037	g24	SRPREC	06037	060373850423A	19
4342	3850558A	037	g24	SRPREC	06037	060373850558A	19
4621	3850072A	037	g24	SRPREC	06037	060373850072A	19
4675	3850294A	037	g24	SRPREC	06037	060373850294A	19
4680	9001691A	037	g24	SRPREC	06037	060379001691A	19
4684	3850075A	037	g24	SRPREC	06037	060373850075A	19
4687	3850161A	037	g24	SRPREC	06037	060373850161A	19
4722	3850163A	037	g24	SRPREC	06037	060373850163A	19
4748	3850164A	037	g24	SRPREC	06037	060373850164A	19
4765	<NA>	037	g24	SRPREC	06037	<NA>	NA

	fips	srprec.y	election.y	geo_type	addist	cddist	sddist	bedist	totreg
4309	06037	3850423A	g24	srprec	69	42	35	3	0
4342	06037	3850558A	g24	srprec	69	44	33	3	3163
4621	06037	3850072A	g24	srprec	69	42	33	3	6572
4675	06037	3850294A	g24	srprec	69	42	33	3	5306
4680	06037	9001691A	g24	srprec	65	44	35	3	2413
4684	06037	3850075A	g24	srprec	69	44	33	3	3761
4687	06037	3850161A	g24	srprec	69	44	33	3	4031
4722	06037	3850163A	g24	srprec	69	42	33	3	1261
4748	06037	3850164A	g24	srprec	69	42	33	3	278
4765	<NA>	<NA>	<NA>	<NA>	NA	NA	NA	NA	NA

	demreg	repreg	aipreg	grnreg	libreg	nlpreg	refreg	dclreg	mscreg	totvote
4309	0	0	0	0	0	0	0	0	0	0
4342	0	0	0	0	0	0	0	0	0	2138
4621	0	0	0	0	0	0	0	0	0	4551
4675	0	0	0	0	0	0	0	0	0	4315
4680	0	0	0	0	0	0	0	0	0	1078
4684	0	0	0	0	0	0	0	0	0	2976
4687	0	0	0	0	0	0	0	0	0	3157
4722	0	0	0	0	0	0	0	0	0	830
4748	0	0	0	0	0	0	0	0	0	158
4765	NA									

	demvote	repvote	aipvote	grnvote	libvote	nlpvote	refvote	dclvote	mscvote
4309									
4342									
4621									
4675									
4680									
4684									
4687									
4722									
4748									
4765									

4309	0	0	0	0	0	0	0	0	0
4342	0	0	0	0	0	0	0	0	0
4621	0	0	0	0	0	0	0	0	0
4675	0	0	0	0	0	0	0	0	0
4680	0	0	0	0	0	0	0	0	0
4684	0	0	0	0	0	0	0	0	0
4687	0	0	0	0	0	0	0	0	0
4722	0	0	0	0	0	0	0	0	0
4748	0	0	0	0	0	0	0	0	0
4765	NA	NA							
	prcvote	absvote	assaip01	assdem01	assdem02	assrep01	assrep02	cngdem01	
4309	0	0	0	0	0	0	0	0	0
4342	0	0	0	1405	0	588	0	1450	
4621	0	0	0	3382	0	894	0	3416	
4675	0	0	0	2350	0	1695	0	2396	
4680	0	0	0	623	0	344	0	730	
4684	0	0	0	1971	0	837	0	1978	
4687	0	0	0	2190	0	777	0	2214	
4722	0	0	0	579	0	184	0	580	
4748	0	0	0	108	0	40	0	112	
4765	NA	NA	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	NA
	cngdem02	cngind01	cngrep01	cngrep02	prsaip01	prsdem01	prsgrn01	prslib01	
4309	0	0	0	0	0	0	0	0	0
4342	0	0	539	0	14	1455	22	11	
4621	0	0	901	0	35	3489	69	17	
4675	0	0	1695	0	43	2515	42	18	
4680	0	0	250	0	18	656	4	4	
4684	0	0	814	0	33	2058	36	28	
4687	0	0	726	0	39	2270	47	16	
4722	0	0	183	0	12	587	14	10	
4748	0	0	38	0	0	112	1	0	
4765	NA	NA	NA	NA	<NA>	<NA>	<NA>	<NA>	<NA>
	prspaf01	prsrep01	pr_2_n	pr_2_y	pr_32_n	pr_32_y	pr_33_n	pr_33_y	pr_34_n
4309	0	0	0	0	0	0	0	0	0
4342	5	591	754	1228	877	1124	1021	926	1020
4621	37	841	1120	3105	1482	2786	1795	2420	2161
4675	14	1588	2107	1913	2525	1519	2809	1164	1995
4680	5	370	295	677	375	606	450	530	504
4684	9	765	1172	1603	1396	1409	1676	1089	1447
4687	19	704	1153	1783	1326	1603	1660	1221	1631
4722	9	187	226	515	314	440	355	390	392
4748	0	44	35	106	42	97	58	83	77
4765	<NA>	<NA>							
	pr_34_y	pr_35_n	pr_35_y	pr_36_n	pr_36_y	pr_3_n	pr_3_y	pr_4_n	pr_4_y
	pr_5_n								
4309	0	0	0	0	0	0	0	0	0
4342	884	603	1358	779	1205	681	1309	714	1266

967										
4621	1900	997	3195	1848	2376	827	3448	972	3282	
1536										
4675	1866	1604	2342	1204	2732	1376	2643	2014	1999	
2596										
4680	444	264	715	334	649	352	607	279	690	
405										
4684	1237	932	1817	1089	1682	797	1993	1055	1745	
1472										
4687	1165	1018	1872	1186	1734	651	2297	1044	1899	
1494										
4722	328	203	535	302	439	204	547	215	536	
306										
4748	61	36	104	54	86	45	102	34	108	
50										
4765	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	
NA										
	pr_5_y	pr_6_n	pr_6_y	sendem01	sendem02	senrep01	senrep02	uspdem01		
	usprep01									
4309	0	0	0	0	0	0	0	0	0	
0										
4342	970	864	1047	1432	0	575	0	1393		
625										
4621	2661	1450	2704	3438	0	902	0	3397		
927										
4675	1399	2268	1646	2284	0	1723	0	2359		
1754										
4680	548	417	541	472	415	0	0	632		
364										
4684	1289	1274	1464	1969	0	845	0	1980		
863										
4687	1401	1187	1672	2210	0	776	0	2183		
815										
4722	441	286	449	580	0	185	0	571		
198										
4748	91	61	76	109	0	40	0	101		
47										
4765	<NA>	<NA>	<NA>	NA	NA	NA	NA	NA		
NA										
	ussdem01	ussrep01		sr_area	new_cdd					
4309	0	0	72519.50	[m^2]	42					
4342	1388	625	1668673.50	[m^2]	42					
4621	3407	908	26181721.00	[m^2]	42					
4675	2313	1712	3219014.70	[m^2]	42					
4680	631	365	9733217.00	[m^2]	42					
4684	1967	863	1627825.00	[m^2]	42					
4687	2159	817	1456907.00	[m^2]	42					
4722	563	200	323584.50	[m^2]	42					

```

4748      104        44    94076.55 [m^2]      42
4765      <NA>      <NA>  530950325.50 [m^2]      42
                    geometry
4309 POLYGON ((162651 -470774, 1...
4342 POLYGON ((168820 -462679, 1...
4621 POLYGON ((164765 -470771, 1...
4675 MULTIPOLYGON (((174608.1 -4...
4680 MULTIPOLYGON (((162569 -470...
4684 POLYGON ((168031 -463530, 1...
4687 POLYGON ((168049 -464514, 1...
4722 POLYGON ((171348 -468507, 1...
4748 MULTIPOLYGON (((168486.8 -4...
4765 POLYGON ((124260 -549730, 1...

```

Compute the share of each SR precinct's area that lies within each new district:

```

sr_new_intersection <- sr_new_intersection |>
  mutate(
    part_area = st_area(geometry),
    weight = as.numeric(part_area / sr_area)
  )

summary(sr_new_intersection$weight)

```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.0000	0.9994	1.0000	0.8355	1.0000	1.0000

The weight column is the fraction of each SR precinct assigned to each new district.

## Allocate Votes to New Districts (Area-Weighted)

We now multiply each SR precinct's votes by the weight to allocate them to new districts, and then sum within each new district.

Here we demonstrate this for the main congressional race columns `cngdem01` and `cngrep01`. You can expand this to other contests if needed.

```
# Make sure the main congressional vote columns exist
c("cngdem01", "cngrep01") %in% names(sr_new_intersection)
```

```
[1] TRUE TRUE
```

```
# Drop geometry and aggregate votes to new districts
newmap_district_votes <- sr_new_intersection |>
  st_drop_geometry() |>
```

```

group_by(new_cdd) |>
summarise(
  d_votes = sum(cngdem01 * weight, na.rm = TRUE),
  r_votes = sum(cngrep01 * weight, na.rm = TRUE),
  .groups = "drop"
)
newmap_district_votes

```

```

# A tibble: 52 × 3
  new_cdd d_votes r_votes
  <chr>    <dbl>   <dbl>
1 01      189840. 160473.
2 02      252190. 171693.
3 03      197348. 174330.
4 04      190249. 144307.
5 05      117927. 205615.
6 06      170484. 151681.
7 07      194068. 152204.
8 08      221721. 112912.
9 09      169805. 119444.
10 10     248627. 127085.
# i 42 more rows

```

These d\_votes and r\_votes are the **area-weighted estimates of total Democratic and Republican votes for each new AB 604 congressional district.**

## Save New District Vote Totals

```
write_csv(newmap_district_votes, "data/newmap-district-votes.csv")
```