Data Cleaning

```
#library and packages
library(readr)
library(dplyr)
```

```


Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(stringr)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ forcats   1.0.1     ✔ purrr     1.1.0
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
```

```
── 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
```

```
raw_data <- read_csv("data/g24_sov_by_g24_svprec.csv")
```

```
Rows: 51123 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.
```

```
vote_cols <- c(
  "TOTREG", "TOTVOTE",
  "CNGDEM01", "CNGDEM02",
  "CNGREP01", "CNGREP02")
#make clean df
cleaned_df <- raw_data |>
  mutate(
    across(where(is.character), str_trim)
  ) |>
    mutate(across(all_of(vote_cols), ~{
    temp <- ifelse(. %in% c("***", "NA", "N/A", ".", "-", ""), NA, .)
    numeric_val <- suppressWarnings(as.numeric(temp))
  numeric_val[is.na(numeric_val)] <- 0
    as.integer(numeric_val)
  }))
#clean dataset
g24_house <-cleaned_df |>
    mutate(
        county = COUNTY, fips = FIPS, svprec = SVPREC, svprec_key =
SVPREC_KEY, cddist = CDDIST, totreg = TOTREG, totvote = TOTVOTE, house_dem =
CNGDEM01 + CNGDEM02, house_rep = CNGREP01 + CNGREP02)|>
    select(county,fips,svprec,svprec_key, cddist, totreg, totvote,
house_dem,house_rep)|>
    filter(!is.na(svprec_key),!is.na(totvote), totvote >=0)

#save
write_csv(g24_house, "data/g24_house_clean.csv")

library(tidyverse)
library(sf)
```

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

```r
# 1. Load cleaned SR-level vote data
sr_raw <- read_csv("/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/
new_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_vote_cols <- c(
  "TOTREG", "TOTVOTE",
  "CNGDEM01", "CNGDEM02",
  "CNGREP01", "CNGREP02"
)

sr_clean <- sr_raw |>
  # trim spaces from all character columns
  mutate(across(where(is.character), str_trim)) |>
  # turn vote columns into integers, cleaning "***" etc
  mutate(
    across(all_of(sr_vote_cols), ~{
      temp <- ifelse(. %in% c("***", "NA", "N/A", ".", "-", ""), NA, .)
      numeric_val <- suppressWarnings(as.numeric(temp))
      numeric_val[is.na(numeric_val)] <- 0
      as.integer(numeric_val)
    })
  )

  sr_votes <- sr_clean |>
  mutate(
    house_dem = CNGDEM01 + CNGDEM02,
    house_rep = CNGREP01 + CNGREP02
  ) |>
  select(SRPREC, house_dem, house_rep)
```

```r
#Load SR precinct geometries
sr_shp <- st_read("/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/
new_data/srprec_state_g24_v01_shp") |>
  st_transform(3310) |>
  st_make_valid()
```

```
Reading layer `srprec_state_g24_v01_shp' from data source
  `/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/new_data/
srprec_state_g24_v01_shp'
  using driver `ESRI Shapefile'
```

```
Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
GDAL
Message 1:
/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/new_data/
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
```

```
# 3. Load proposed 2025 district map (AB 604)
ab604 <- st_read("/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/
new_data/AB604/AB604.shp")|>
  st_transform(3310) |>
  st_make_valid()
```

```
Reading layer `AB604' from data source
  `/Users/alaiawittfeld/gerrymandering-alaiaslw-1/data/new_data/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
```

```
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 344 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.
~~~

~~~
ab604_new <- ab604 %>%
  mutate(new_cddist = DISTRICT)

overlap <- st_intersection(sr_geo, ab604_new)
~~~

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

~~~
overlap <- overlap %>%
  mutate(piece_area = st_area(geometry)) %>%
  group_by(SRPREC) %>%
  mutate(
    sr_area = sum(piece_area),
    weight = as.numeric(piece_area / sr_area),
    dem_alloc = house_dem * weight,
    rep_alloc = house_rep * weight
  ) %>%
  ungroup()
new_district_results <- overlap %>%
  st_drop_geometry() %>%
  group_by(new_cddist) %>%
  summarise(
    dem_votes = sum(dem_alloc, na.rm = TRUE),
    rep_votes = sum(rep_alloc, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(new_cddist)
write_csv(new_district_results, "data/new_district_results.csv")
~~~

```
```