Data Cleaning

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.1     ✔ stringr   1.5.2
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── 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
```

```
# Load the precinct-level dataset
votes_precinct <- 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.
```

```
head(votes_precinct)
```

```
# 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>
```

```
<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>, REFVOTE <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>, …
```

```r
votes_precinct <- votes_precinct |>
    mutate(
        # Remove spaces at the beginning/end of text entries
        across(where(is.character), str_trim),
        # Convert all text to uppercase
        across(where(is.character), str_to_upper),
        # Remove any double spaces within text
        across(where(is.character), str_squish),
        # Replace masked values *** with NA
        across(where(is.character), ~ na_if(.x, "***")))

#Convert key columns to the correct data type
votes_precinct <- votes_precinct |>
    mutate(COUNTY = as.integer(COUNTY),
           FIPS = as.character(FIPS),
           CDDIST = as.integer(CDDIST),
           SDDIST = as.integer(SDDIST),
           BEDIST = as.integer(BEDIST),
           TOTREG = as.numeric(TOTREG))

# Each precinct should have a unique SVPREC_KEY. If duplicates exist, only
keep the first record.
sum(duplicated(votes_precinct$SVPREC_KEY))
```

```
[1] 0
```

```r
votes_precinct <- votes_precinct |> distinct(SVPREC_KEY, .keep_all = TRUE)

write_csv(votes_precinct, "data/g24_sov_by_g24_svprec_clean.csv")
votes_precinct
```

```
# A tibble: 51,123 × 76
   COUNTY FIPS  SVPREC  ADDIST SVPREC_KEY ELECTION GEO_TYPE CDDIST SDDIST
BEDIST
    <int> <chr> <chr>    <dbl> <chr>      <chr>    <chr>     <int>  <int>
<int>
 1      1 06001 200100      14 060012001… G24      SVPREC       12      7
2
 2      1 06001 200100A     14 060012001… G24      SVPREC       12      7
2
 3      1 06001 200200      14 060012002… G24      SVPREC       12      7
2
 4      1 06001 200200A     14 060012002… G24      SVPREC       12      7
2
 5      1 06001 201400      14 060012014… G24      SVPREC       12      7
2
 6      1 06001 201400A     14 060012014… G24      SVPREC       12      7
2
 7      1 06001 202200      14 060012022… G24      SVPREC       12      7
2
 8      1 06001 202200A     14 060012022… G24      SVPREC       12      7
2
 9      1 06001 202500      14 060012025… G24      SVPREC       12      7
2
10      1 06001 202500A     14 060012025… G24      SVPREC       12      7
2
# ℹ 51,113 more rows
# 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>, REFVOTE <dbl>, DCLVOTE <dbl>,
#   MSCVOTE <dbl>, PRCVOTE <dbl>, ABSVOTE <dbl>, ASSAIP01 <chr>,
#   ASSDEM01 <chr>, ASSDEM02 <chr>, ASSREP01 <chr>, ASSREP02 <chr>, …
```

Note: According to the Statewide Database disclaimers, some precincts have fewer than ten voters and are masked for privacy, so certain vote count fields are recorded as 0 or ***. These were replaced with NA where appropriate. A few 0s are normal and expected in uncontested races or small precincts.

```r
View(votes_precinct)
```

Section 2:

```r
library(sf)
```

```
Linking to GEOS 3.13.1, GDAL 3.11.0, PROJ 9.6.0; sf_use_s2() is TRUE
```

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

```
Reading layer `srprec_state_g24_v01_shp' from data source
  `C:\Users\jiaaa\Desktop\gerrymandering-
Selina568\data\shapefiles\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:
C:\Users\jiaaa\Desktop\gerrymandering-
Selina568\data\shapefiles\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 24145 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
sr_shp <- sr_shp |>
    st_transform(3310) |>
    st_set_precision(1) |>
    st_make_valid() |>
    st_collection_extract("POLYGON")
```

```r
cd_ab604 <- st_read("data/shapefiles/AB604.shp") |>
    st_transform(3310)
```

```
Reading layer `AB604' from data source
  `C:\Users\jiaaa\Desktop\gerrymandering-Selina568\data\shapefiles\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_votes <- 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.
```

```
sr_data <- left_join(sr_shp, sr_votes, by = c("SRPREC_KEY" = "SRPREC_KEY"))
```

```
reallocated_DEM <- st_interpolate_aw(sr_data["DEMREG"], cd_ab604, extensive =
TRUE)
```

```
Warning in st_interpolate_aw.sf(sr_data["DEMREG"], cd_ab604, extensive =
TRUE):
st_interpolate_aw assumes attributes are constant or uniform over areas of x
```

```
reallocated_REP <- st_interpolate_aw(sr_data["REPREG"], cd_ab604, extensive =
TRUE)
```

```
Warning in st_interpolate_aw.sf(sr_data["REPREG"], cd_ab604, extensive =
TRUE):
st_interpolate_aw assumes attributes are constant or uniform over areas of x
```

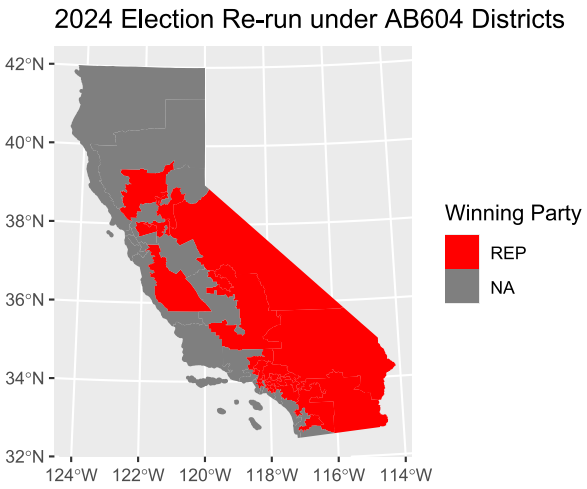
```
cd_ab604$DEM <- reallocated_DEM$DEMREG
cd_ab604$REP <- reallocated_REP$REPREG

cd_ab604 <- cd_ab604 |> mutate(TOTAL = DEM + REP, WINNER = if_else(DEM > REP,
"DEM", "REP"))

head(cd_ab604)
```

```
Simple feature collection with 6 features and 19 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 123581.1 ymin: -590224.2 xmax: 540036.5 ymax: -236299.5
Projected CRS: NAD83 / California Albers
  DISTRICT  A_POP DEVIATION   CVAP HSP_CVAP IND_CVAP BLK_CVAP ASN_CVAP
WHT_CVAP
1       42 760067         1 547320   134603     4347    36075    69836
295693
2       40 760066         0 543973   152392     4037    28337    63206
291303
3       49 760067         1 525988    96790     4217    16308    67875
336391
4       47 760065        -1 514402    78502     2812    13187   130254
284408
5       23 760066         0 514103   190014     6566    46719    22557
242849
6       48 760066         0 518620   166118     6486    27573    43349
268028
  CVAP_PCT HSP_CVAP_P IND_CVAP_P BLK_CVAP_P ASN_CVAP_P WHT_CVAP_P
1 0.720094   0.245931   0.007942   0.065912   0.127596   0.540256
2 0.715692   0.280146   0.007421   0.052093   0.116193   0.535510
3 0.692028   0.184016   0.008017   0.031005   0.129043   0.639541
4 0.676787   0.152608   0.005467   0.025636   0.253214   0.552891
5 0.676393   0.369603   0.012772   0.090875   0.043876   0.472374
6 0.682335   0.320308   0.012506   0.053166   0.083585   0.516810
                       geometry DEM REP TOTAL WINNER
1 MULTIPOLYGON (((194609.4 -4...  NA  NA    NA   <NA>
2 MULTIPOLYGON (((216125.6 -4...   0   0     0    REP
3 MULTIPOLYGON (((218794.2 -4...  NA  NA    NA   <NA>
4 MULTIPOLYGON (((200995.2 -4...  NA  NA    NA   <NA>
5 MULTIPOLYGON (((273340.5 -4...   0   0     0    REP
6 MULTIPOLYGON (((311779.4 -4...   0   0     0    REP
```

```r
ggplot(cd_ab604) +
    geom_sf(aes(fill = WINNER), color = NA) +
    scale_fill_manual(values = c("DEM" = "blue", "REP" = "red")) +
    labs(title = "2024 Election Re-run under AB604 Districts", fill = "Winning
Party")
```


2024 Election Re-run under AB604 Districts
42°N
40°N
38°N
36°N
34°N
32°N
124°W
122°W
120°W
118°W
116°W
114°W
Winning Party
REP
NA