Data Cleaning

```r
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
```

```r
sov_2024 <- read_csv("data/g24_sov_by_g24_svprec.csv") #Write into R Studio
```

```
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.
```

```r
glimpse(sov_2024) #Take a look at it
```

```
Rows: 51,123
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…
```

```
sum(duplicated(sov_2024$SVPREC_KEY)) #Check for uniqueness of each precinct
```

```
[1] 0
```

```
sum(is.na(sov_2024$CDDIST)) #Check for precincts that dont have congressional
District number
```

```
[1] 0
```

```
sov_2024 <- sov_2024 %>% #Cleaning for columns of interest
    mutate(
    COUNTY = as.numeric(COUNTY),
    FIPS   = as.numeric(FIPS),
    CDDIST = as.numeric(CDDIST)
    ) %>%
    mutate(
    CNGDEM01 = as.numeric(CNGDEM01),
    CNGREP01 = as.numeric(CNGREP01)
    )
```

```
Warning: There were 2 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 1 remaining warning.
```

```
glimpse(sov_2024)
```

```
Rows: 51,123
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       <dbl> 6001, 6001, 6001, 6001, 6001, 6001, 6001, 6001, 6001,
6001,…
```

```
$ 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   <dbl> 102, 1668, 108, 1063, 139, 1688, 35, 192, 172, 150, 366,
15…
$ 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   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA,
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…
```

```
# keep only the columns we need and create total votes + Democratic share

precinct_votes <- sov_2024 %>%
    select(
    COUNTY,
    SVPREC_KEY,
    CDDIST,
    dem_votes = CNGDEM01,
    rep_votes = CNGREP01
) %>%
    mutate(
    total_votes = dem_votes + rep_votes,
    dem_share   = dem_votes / total_votes
    )

# quick check of the new precinct-level dataset

glimpse(precinct_votes)
```

```
Rows: 51,123
Columns: 7
$ COUNTY      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1…
$ SVPREC_KEY  <chr> "06001200100", "06001200100A", "06001200200",
"06001200200…
$ CDDIST      <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,
12…
$ dem_votes   <dbl> 102, 1668, 108, 1063, 139, 1688, 35, 192, 172, 150, 366,
1…
$ rep_votes   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA,
…
$ total_votes <dbl> 102, 1668, 108, 1063, 139, 1688, 35, 192, 172, 150, 366,
1…
$ dem_share   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, NA,
…
```

```
# Question 1: compute basic turnout summaries across precincts

# remove rows where total_votes is NA using base indexing
```

```
precinct_turnout <- precinct_votes[!is.na(precinct_votes$total_votes), ]

# number of precincts with valid turnout

n_precincts <- nrow(precinct_turnout)

# total votes across all precincts

total_votes_all <- sum(precinct_turnout$total_votes, na.rm = TRUE)

# average precinct turnout

avg_votes <- mean(precinct_turnout$total_votes, na.rm = TRUE)

# smallest and largest precinct turnout

min_votes <- min(precinct_turnout$total_votes, na.rm = TRUE)
max_votes <- max(precinct_turnout$total_votes, na.rm = TRUE)

n_precincts
```

```
[1] 46439
```

```
total_votes_all
```

```
[1] 101589643
```

```
avg_votes
```

```
[1] 2187.593
```

```
min_votes
```

```
[1] 0
```

```
max_votes
```

```
[1] 3324096
```

```
# Question 2: summarize Democratic vote share at the precinct level

# keep rows with non-missing dem_share using base indexing

precinct_share <- precinct_votes[!is.na(precinct_votes$dem_share), ]

# center and spread of precinct-level Democratic vote share

mean_dem_share_precinct   <- mean(precinct_share$dem_share, na.rm = TRUE)
median_dem_share_precinct <- median(precinct_share$dem_share, na.rm = TRUE)
min_dem_share_precinct    <- min(precinct_share$dem_share, na.rm = TRUE)
max_dem_share_precinct    <- max(precinct_share$dem_share, na.rm = TRUE)

mean_dem_share_precinct
```

```
[1] 0.5384042
```

```
median_dem_share_precinct
```

```
[1] 0.5457199
```

```
min_dem_share_precinct
```

```
[1] 0
```

```
max_dem_share_precinct
```

```
[1] 1
```

```
# classify precincts as Strong D, Strong R, or Mixed based on dem_share

precinct_strength <- precinct_share |>
    mutate(
        category = if_else(
        dem_share >= 0.6,
        "Strong D",
        if_else(
        dem_share <= 0.4,
        "Strong R",
        "Mixed"
        )
    )
) |>
group_by(category) |>
    summarize(
    n = n()
)

precinct_strength
```

```
# A tibble: 3 × 2
  category     n
  <chr>    <int>
1 Mixed    11513
2 Strong D 13660
3 Strong R  8855
```

```
# Question 3: aggregate precinct votes up to Congressional districts

# drop rows with missing CDDIST using base indexing

sov_non_missing_cd <- sov_2024[!is.na(sov_2024$CDDIST), ]

# sum Democratic and Republican votes within each district

district_votes <- sov_non_missing_cd |>
    select(
    CDDIST,
    dem_votes = CNGDEM01,
    rep_votes = CNGREP01
    ) |>
        mutate(
        total_votes = dem_votes + rep_votes
        ) |>
            group_by(CDDIST) |>
                summarize(
                dem_votes = sum(dem_votes, na.rm = TRUE),
                rep_votes = sum(rep_votes, na.rm = TRUE),
                total_votes = sum(total_votes, na.rm = TRUE)
                ) |>
                    mutate(
                    dem_share = dem_votes / total_votes,
                        winner = if_else(
                        dem_votes > rep_votes,
                        "DEM",
                        "REP"
                        )
                    )

district_votes
```

```
# A tibble: 53 × 6
   CDDIST dem_votes rep_votes total_votes dem_share winner
    <dbl>     <dbl>     <dbl>       <dbl>     <dbl> <chr>
 1      0  52843210  34247705    87090915     0.607 DEM
 2      1    110472    208150      318622     0.347 REP
 3      2    272384    106407      378791     0.719 DEM
 4      3    187960    233895      421855     0.446 REP
 5      4    227321    114644      341965     0.665 DEM
 6      5    134467    214223      348690     0.386 REP
 7      6    165386    121625      287011     0.576 DEM
 8      7    197361     98273      295634     0.668 DEM
 9      8    201756     70932      272688     0.740 DEM
10      9    130093    121006      251099     0.518 DEM
# ℹ 43 more rows
```

```
# count how many districts are won by each party

district_wins <- district_votes |>
group_by(winner) |>
summarize(
n_districts = n()
)

district_wins
```

```
# A tibble: 2 × 2
  winner n_districts
  <chr>        <int>
1 DEM             44
2 REP              9
```

```
# summarize Democratic vote share at the district level

mean_dem_share_district <- mean(district_votes$dem_share, na.rm = TRUE)
median_dem_share_district <- median(district_votes$dem_share, na.rm = TRUE)
min_dem_share_district <- min(district_votes$dem_share, na.rm = TRUE)
max_dem_share_district <- max(district_votes$dem_share, na.rm = TRUE)

mean_dem_share_district
```

```
[1] 0.615584
```

```
median_dem_share_district
```

```
[1] 0.6272145
```

```
min_dem_share_district
```

```
[1] 0
```

```
max_dem_share_district
```

```
[1] 1
```

```
# save precinct- and district-level datasets for use in other .qmd files

write_csv(precinct_votes, "data/precinct_votes_clean.csv")
write_csv(district_votes, "data/district_votes_2024.csv")
```

###PART 5

library(sf) library(tidyverse)

g24_sr_precinct <- st_read(“data/shapefiles/srprec_state_g24_v01_shp.shp”) g24_proposed_congressional_map <- st_read(“data/shapefiles/AB604.shp”)

g24_sov_precinct <- read_csv(“data/g24_sov_by_g24_svprec.csv”)

sr_shp <- g24_sr_precinct |> st_transform(3310) |> st_set_precision(1) |> st_make_valid() |> st_collection_extract(“POLYGON”)

proposed_map <- g24_proposed_congressional_map |> st_transform(3310) |> st_set_precision(1) |> st_make_valid() |> st_collection_extract(“POLYGON”)

overlapping <- st_intersection(sr_shp |> select(SRPREC), proposed_map |> select(new_dist = DISTRICT)) |> mutate(overlap_area = st_area(geometry)) |> group_by(SRPREC) |> mutate(prop_of_precinct = as.numeric(overlap_area / sum(overlap_area)))

votes_precinct <- g24_sov_precinct |> select(SVPREC, CNGDEM01, CNGREP01) |> mutate(CNGDEM01 = as.numeric(CNGDEM01), CNGREP01 = as.numeric(CNGREP01)) |> rename(SRPREC = SVPREC, d_votes = CNGDEM01, r_votes = CNGREP01)

```
joined_precinct <- overlapping |> left_join(votes_precinct, by = “SRPREC”) |> mutate( d_votes_based_district = d_votes * prop_of_precinct, r_votes_based_district = r_votes * prop_of_precinct )
```

```
final <- joined_precinct |> group_by(new_dist) |> summarize( d_votes_total_dist = sum(d_votes_based_district, na.rm = TRUE), r_votes_total_dist = sum(r_votes_based_district, na.rm = TRUE) ) |> mutate( winner = if_else(d_votes_total_dist > r_votes_total_dist, “D”, “R”), total = d_votes_total_dist + r_votes_total_dist, d_prop = d_votes_total_dist / total, r_prop = r_votes_total_dist / total )
```

```
write_csv(final, “data/final.csv”)
```