

Geospatial Mapping

Libraries/Packages

```
#Function to combine the state and county fips codes
combineFips <- function(state_fips, county_fips) {
  state_fips <- as.numeric(state_fips)
  county_fips <- as.numeric(county_fips)

  state_fips_str <- sprintf("%02d", state_fips)
  county_fips_str <- sprintf("%03d", county_fips)

  full_fips_code <- paste0(state_fips_str, county_fips_str)
  #print(full_fips_code)

  return(full_fips_code)
}
```

Fipscodes dataset can be found here:

<https://www.census.gov/geographies/reference-files/2021/demo/popest/2021-fips.html>

```
gov_fipscodes <- readr::read_csv('data/all-geocodes-v2021.csv', skip = 4)

New names:
Rows: 43833 Columns: 11
-- Column specification
----- Delimiter: ","
(7): Summary Level, State Code (FIPS), County Code (FIPS), County Subdiv... lgl
(4): ...8, ...9, ...10, ...11
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.
* `` -> `...8`
```

```

* `` -> `...9`
* `` -> `...10`
* `` -> `...11`


gov_fipscodes_clean <- gov_fipscodes |>
  rename(
    county_fips = 'County Code (FIPS)',
    state_fips = 'State Code (FIPS)',
    area_name = 'Area Name (including legal/statistical area description)'
  ) |>
  filter(
    county_fips != "000",
    state_fips != "00"
  ) |>
  mutate(
    state_fips = as.numeric(state_fips),
    county_fips = as.numeric(county_fips)
  ) |>
  select(state_fips, county_fips, area_name)

complete_codes = c(1, 2, 4, 201, 203)

#Fips codes can change over time ...
#So even if R packages are up to date, the data may not be


comp_mapping <- comp_internet |>
  select(
    "county_fips" = "GTC0",
    "Status" = "HUFINAL",
    "state_fips" = "GESTFIPS",
    "CBSAcode" = "GTCBSA",
    "UseDesktop" = "HEDESKTP",
    "UseLaptop" = "HELAPTOP",
    "UseTablet" = "HETABLET",
    "UseSmartphone" = "HEMPHONE",
    "UseWearable" = "HEWEARAB",
    "UseInternet" = "HEINHOME",
    "UseInternetSchool" = "HEINSchl",
    "UseDataplan" = "HEMOBDAT",
    "UseInternetTrainEdu" = "PEEDTRAI"
  ) |>
  drop_na(state_fips, county_fips) |>

```

```

filter(
  any(Status == complete_codes),
  county_fips != "0",
) |>
mutate(
  fips = combineFips(state_fips, county_fips),
  state = fips_info(state_fips)$full
) |>
group_by(state_fips, county_fips, fips, state, UseDesktop) |>
count() |>
filter(UseDesktop == 1 | UseDesktop == 2) |>
pivot_wider(
  names_from = UseDesktop,
  values_from = n
) |>
rename( Yes = "1", No = "2" ) |>
mutate(proportion = Yes/(Yes + No))

fips_info(01041)

```

	full	abbr	county	fips
1	Alabama	AL	Crenshaw County	01041

```

#Show distribution of country code
comp_internet |>
  group_by(GTCO) |>
  summarise( count = n())

```

```

# A tibble: 100 x 2
  GTCO count
  <dbl> <int>
1     0 76179
2     1 4441
3     3 4385
4     5 1690
5     7  281
6     9  797
7    11 1262
8    13 2380
9    15  611

```

```

10     17 1362
# i 90 more rows

#Show how many nonzero country codes there were
comp_internet |>
  filter(GTC0 != 0) |>
  summarise( count = n())

# A tibble: 1 x 1
count
<int>
1 51196

#Read the books on mapping + spatial viz
county_map <- map_data("county") |>
  rename("state" = "region") |>
  mutate(
    "state" = str_to_title(state),
    "subregion" = str_to_title(subregion)
  )

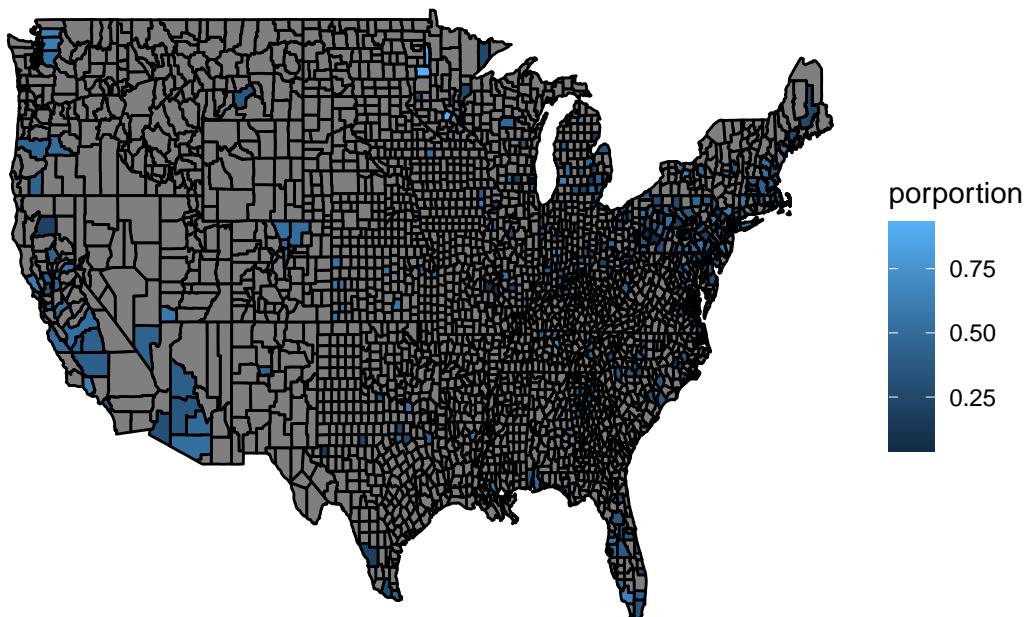
fipsCoded <- merge(comp_mapping, gov_fipscodes_clean, by = c("state_fips", "county_fips"))
  mutate(
    subregion = word(area_name, start = 1, end = 1)
  ) |>
  rename(
    "county" = "area_name"
  ) |>
  mutate(
    "fips" = combineFips(state_fips, county_fips)
  )

#Unfortunately mapping by county isn't particularly useful due to lack of data
internet_map <- fipsCoded |>
  full_join(county_map, by = join_by(subregion == subregion, state == state))

Warning in full_join(fipsCoded, county_map, by = join_by(subregion == subregion, : Detected a
i Row 1 of `x` matches multiple rows in `y`.
i Row 22063 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.

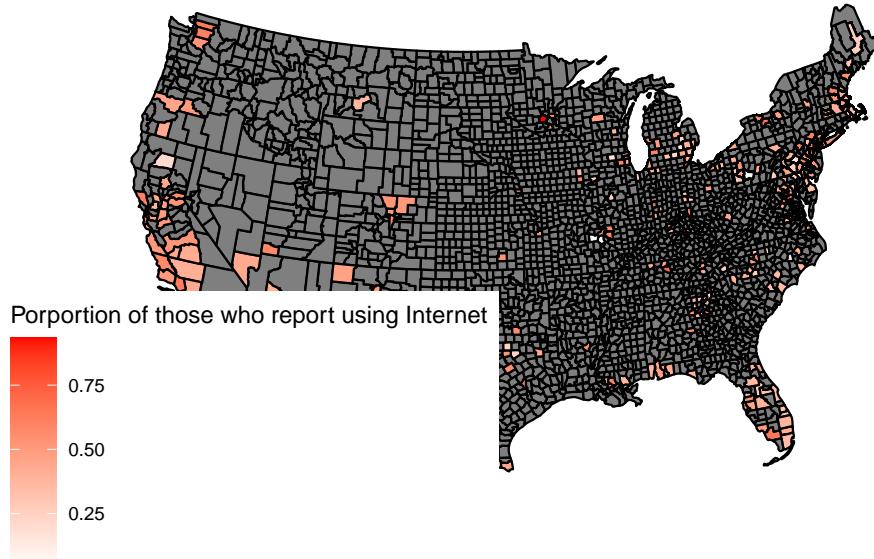
```

```
internet_map |>  
  ggplot(aes(x = long, y = lat, group = group, fill = porportion)) +  
  geom_polygon(color = "black") +  
  theme_void()
```



```
fips_trimmed <- fipsCoded |>  
  select(fips, state, county, porportion)  
  
#Map for porportion of people that report using a desktop by county  
plot_usmap(data = fips_trimmed,  
            values = "porportion",  
            regions = "counties") +  
  scale_fill_continuous(low = "white", high = "red", name = "Porportion of those who repor  
  labs(title = "County Map")
```

County Map



```
#Investigate different map packages
#Plotting two variables on one map is difficult
#Consider mapping two colors for reading and math scores
#geom_maps (maps package)
#Dot Size + color
#Shape + color
#Think about the story I want to tell
#Mapping differences between math & reading scores
#Think about direction of project: modeling/visualizing, deliverables, etc.

#If I am dividing my data, note if there are differences in distributions/analysis
#Potential mapping of district to county?
#Stuff like the "FIPS" issues may be included in a process section, make sure to keep note
#Comparing different subjects, use the same scale so it's easier to compare standard deviations
#Standard deviations are good for relative comparisons and can't be generalized to absolute values
#See if there's another location identifier I could use
#Look into another long scale survey/question internet/education online/enrichment
#Investigate the large number of not specified county codes, possibly information not provided
#Start bivariate
```

```

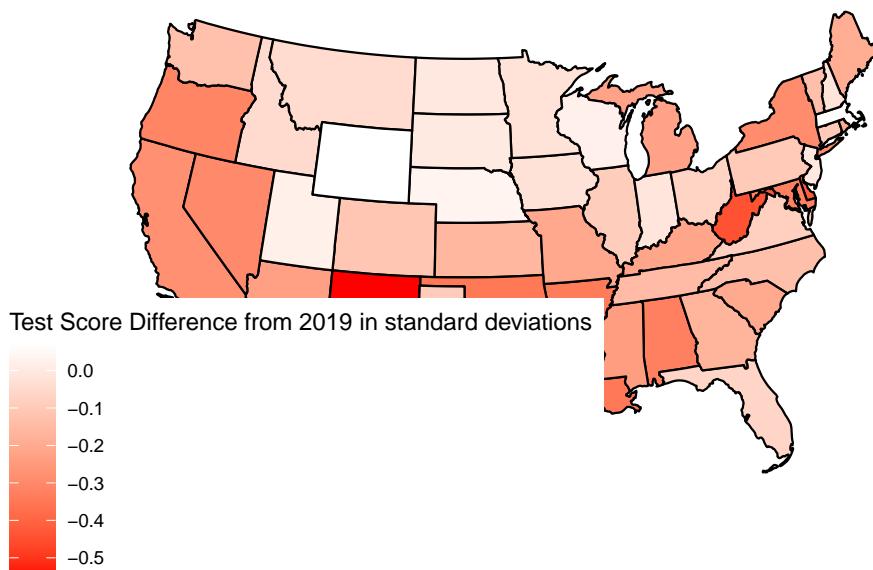
state_grouped_all <- test_state_ys |>
  filter(subgroup == "all") |>
  group_by(sedafipsname, subject) |>
  summarise( mean_ol = mean(ys_mn_2022_ol), mean_eb = mean(ys_mn_2022_eb
)) |>
  rename (
    state = sedafipsname
  )

`summarise()` has grouped output by 'sedafipsname'. You can override using the
`.groups` argument.

plot_usmap(data = state_grouped_all |> filter(subject == "mth"), values = "mean_ol", region
scale_fill_continuous(low = "red", high = "white", name = "Test Score Difference from 20
labs(title = "State Map")

```

State Map



```

# Neither the maps package or the USMaps package support mapping by district
# I will try mapping districts to counties
# https://www.census.gov/programs-surveys/saipe/guidance-geographies/districts-counties.html
# If this doesn't work out, will likely have to use the sf package with shapefiles

```

```

# One thing

# US Census Data on District by County
district_by_county <- read_excel("data/sdlist-21.xls", skip = 2)

# US Census Small Area Income and Poverty Estimates (SAIPE) by District
saipe_district <- read_excel("data/ussd22.xls", skip = 2)

```

Something to consider, sometimes states can have duplicate county names

<https://www.census.gov/programs-surveys/saipe/guidance-geographies/same-name/2022.html>

It is really important that I use DistrictID for each unique **SCHOOL** district.

```

census_clean <- comp_internet |>
  select(
    "county_fips" = "GTC0",
    "state_fips" = "GESTFIPS",
    "Status" = "HUFINAL",
    "CBSAcode" = "GTCBSA",
    "UseDesktop" = "HEDESKTP",
    "UseLaptop" = "HELAPTOP",
    "UseTablet" = "HETABLET",
    "UseSmartphone" = "HEMPHONE",
    "UseWearable" = "HEWEARAB",
    "UseInternet" = "HEINHOME",
    "UseInternetSchool" = "HEINSCHL",
    "UseDataplan" = "HEMOBDAT",
    "UseInternetTrainEdu" = "PEEDTRAI"
  )

valid_county <- census_clean |>
  filter(
    county_fips != 0
  ) |>
  mutate(
    state = fips_info(state_fips)$full,
  ) |>
  group_by(state, Status) |>
  count()

```

```

valid_county_sum <- valid_county |>
  group_by(Status) |>
  summarise(n = sum(n)) |>
  arrange(-n)

invalid_county <- census_clean |>
  filter(
    county_fips == 0
  ) |>
  mutate(
    state = fips_info(state_fips)$full,
  ) |>
  group_by(state, Status) |>
  count()

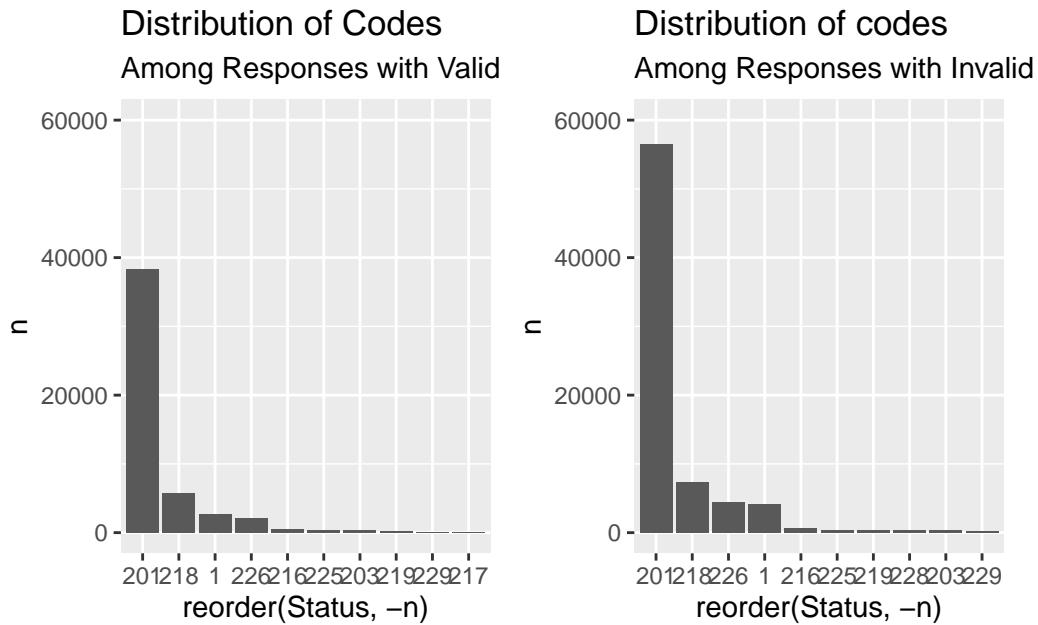
invalid_county_sum <- invalid_county |>
  group_by(Status) |>
  summarise(n = sum(n)) |>
  arrange(-n)

valid_county_status_plot <- ggplot(valid_county_sum |> slice(1:10)) +
  geom_col(aes(x = reorder(Status, -n), y = n) ) +
  #geom_col(data = invalid_county_sum) +
  labs(
    title = "Distribution of Codes",
    subtitle = "Among Responses with Valid counties"
  ) +
  scale_y_continuous( limits = c(0, 60000) )

invalid_county_status_plot <- ggplot(invalid_county_sum |> slice(1:10)) +
  geom_col(aes(x = reorder(Status, -n), y = n)) +
  labs(
    title = "Distribution of codes",
    subtitle = "Among Responses with Invalid counties"
  ) +
  scale_y_continuous( limits = c(0, 60000) )

valid_county_status_plot + invalid_county_status_plot

```



There does not appear to be a significant difference in the completion code distribution between respondents in valid counties vs invalid counties

Code Mappings for “Status”

- 1: Fully Complete CATI (computer-assisted telephone interviewing)
- 2: Partially Complete CATI
- 201: CAPI (computer-assisted personal interviewing)

There appears to be no correlation between the interview completion

```
#Mapping by state since county data is very incomplete
comp_state_mapping <- comp_internet |>
  select(
    "county_fips" = "GTC0",
    "Status" = "HUFINAL",
    "state_fips" = "GESTFIPS",
    "CBSAcode" = "GTCBSA",
    "UseDesktop" = "HEDESKTP",
    "UseLaptop" = "HELAPTOP",
    "UseTablet" = "HETABLET",
    "UseSmartphone" = "HEMPHONE",
```

```

"UseWearable" = "HEWEARAB",
"UseInternet" = "HEINHOME",
"UseInternetSchool" = "HEINSCHL",
"UseDataplan" = "HEMOBDAT",
"UseInternetTrainEdu" = "PEEDTRAI"
) |>
drop_na(state_fips, county_fips) |>
filter(
  any(Status == complete_codes)
) |>
mutate(
  state = fips_info(state_fips)$full
) |>
group_by(state, UseDesktop) |>
count() |>
filter(UseDesktop == 1 | UseDesktop == 2) |>
pivot_wider(
  names_from = UseDesktop,
  values_from = n
) |>
rename( Yes = "1", No = "2" ) |>
mutate(proportion = Yes/(Yes + No))

state_map <- map_data("state") |>
mutate(
  "region" = str_to_title(region),
)

internet_map <- comp_state_mapping |>
left_join(state_map, by = join_by(state == region))

state_centroids <- state_map|>
group_by(region) |>
summarise( long = mean (long), lat = mean(lat))

score_map <- state_grouped_all |>
left_join(state_centroids, by = join_by(state == region))

ggplot(internet_map, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group, fill = proportion), color = "black") +

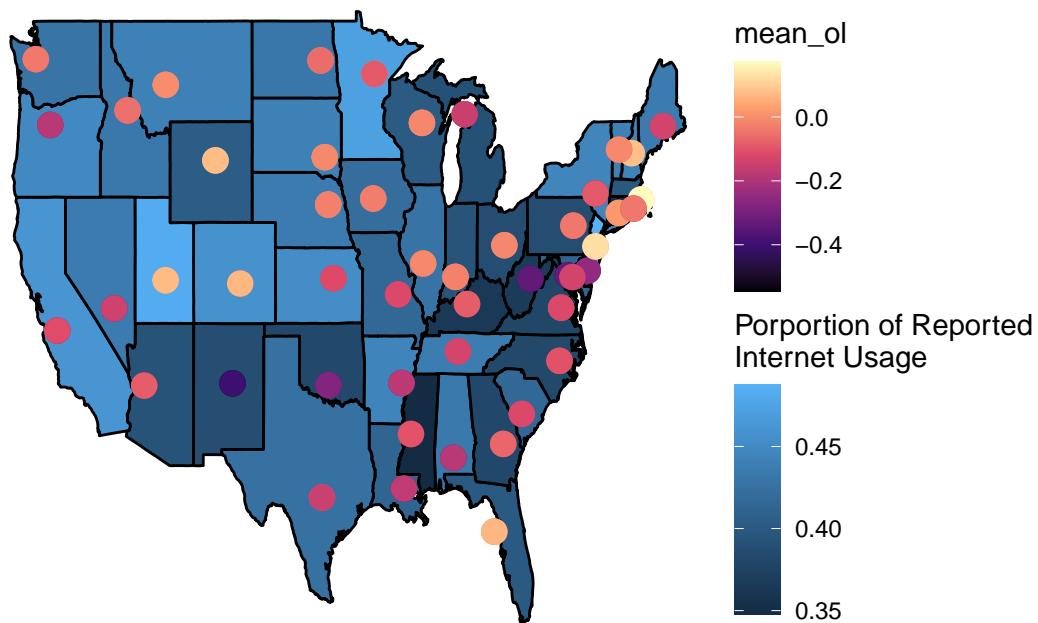
```

```

geom_point(data = score_map, size = 4, aes(color = mean_ol)) +
scale_color_viridis_c(option = "A") +
theme_void() +
labs(
  fill = "Porportion of Reported\nInternet Usage"
)

```

Warning: Removed 4 rows containing missing values (`geom_point()`).



Unfortunately, it's really hard to make meaningful conclusions or analysis about state metrics

```

#Since county codes did not cover a significant geographic portion of the US, I want to se

#Show distribution of country codes
comp_internet |>
  group_by(GTC0) |>
  summarise( count = n())

# A tibble: 100 x 2

```

```

GTC0 count
<dbl> <int>
1      0 76179
2      1 4441
3      3 4385
4      5 1690
5      7 281
6      9 797
7     11 1262
8     13 2380
9     15  611
10    17 1362
# i 90 more rows

#Show how many nonzero country codes there were
comp_internet |>
  filter(GTC0 != 0) |>
  summarise( count = n())

# A tibble: 1 x 1
count
<int>
1 51196

#Distribution of CBSA codes
comp_internet |>
  group_by(GTCBSA) |>
  count() |>
  arrange(-n)

# A tibble: 261 x 2
# Groups:   GTCBSA [261]
  GTCBSA      n
  <dbl> <int>
1      0 33615
2  35620  5025
3 47900  3815
4 31080  3464
5 16980  2516
6 37980  2283

```

```

7 14460 2189
8 19100 2010
9 26420 1888
10 33100 1525
# i 251 more rows

#Show how many nonzero CBSA codes
comp_internet |>
  filter(GTCBSA != 0) |>
  count()

# A tibble: 1 x 1
  n
  <int>
1 93760

```

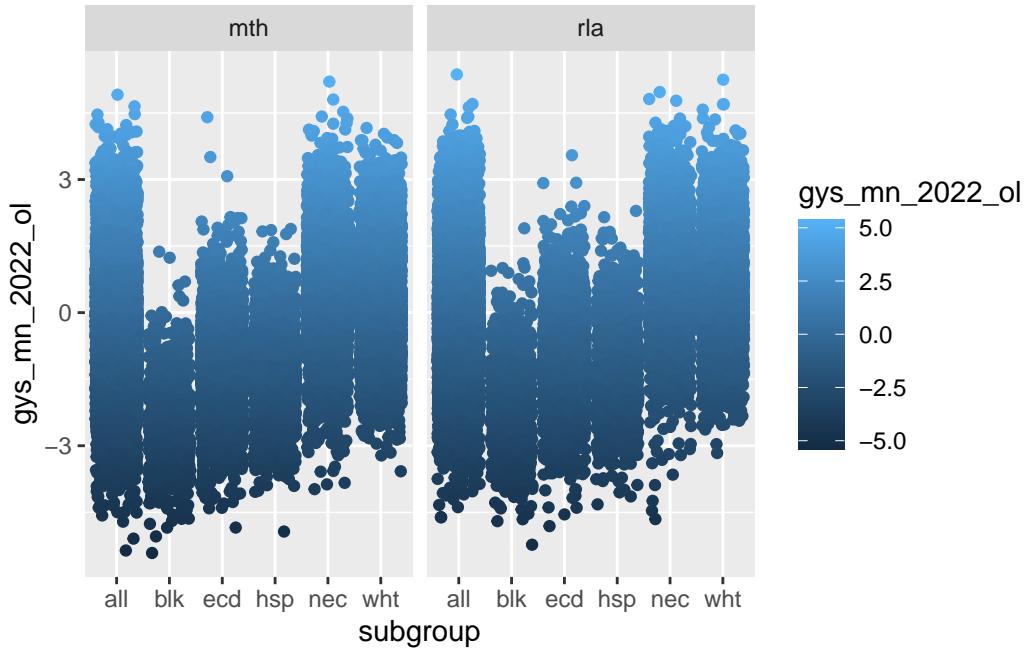
In conclusion, while there are fewer invalid/incomplete CBSA codes, a vast majority of other datasets tend to use districts or counties. As such, I will not be moving forward with CBSA codes.

The Educational Opportunity Project Standardizes their data with the Stanford Education Data Archive (SEDA). They created their own identifier for each district, SEDA Administrative District ID, which details can be found here: <https://edopportunity.org/methods/>

```

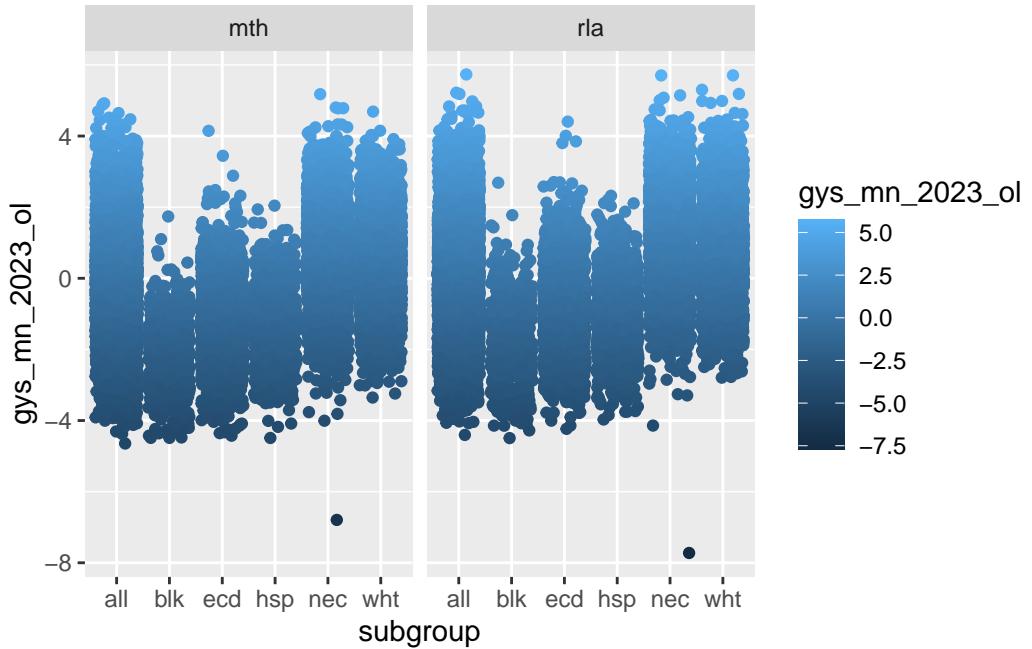
ggplot(test_admindist_gys) +
  geom_jitter(
    aes(
      x = subgroup,
      y = gys_mn_2022_ol,
      color = gys_mn_2022_ol
    ),
    )
  ) +
  facet_grid(
    cols = vars(subject)
  )

```



```
ggplot(test_admindist_gys) +
  geom_jitter(
    aes(
      x = subgroup,
      y = gys_mn_2023_ol,
      color = gys_mn_2023_ol
    ),
  ) +
  facet_grid(
    cols = vars(subject)
  )
```

Warning: Removed 9374 rows containing missing values (`geom_point()`).



```

ggplot(test_admindist_gys |> filter (subject == "mth")) +
  geom_jitter(
    aes(
      x = subgroup,
      y = gys_mn_2022_ol,
      color = gys_mn_2022_ol
    ),
    ) +
  scale_y_continuous(
    limits = c(-6, 6)
  ) +
  
```

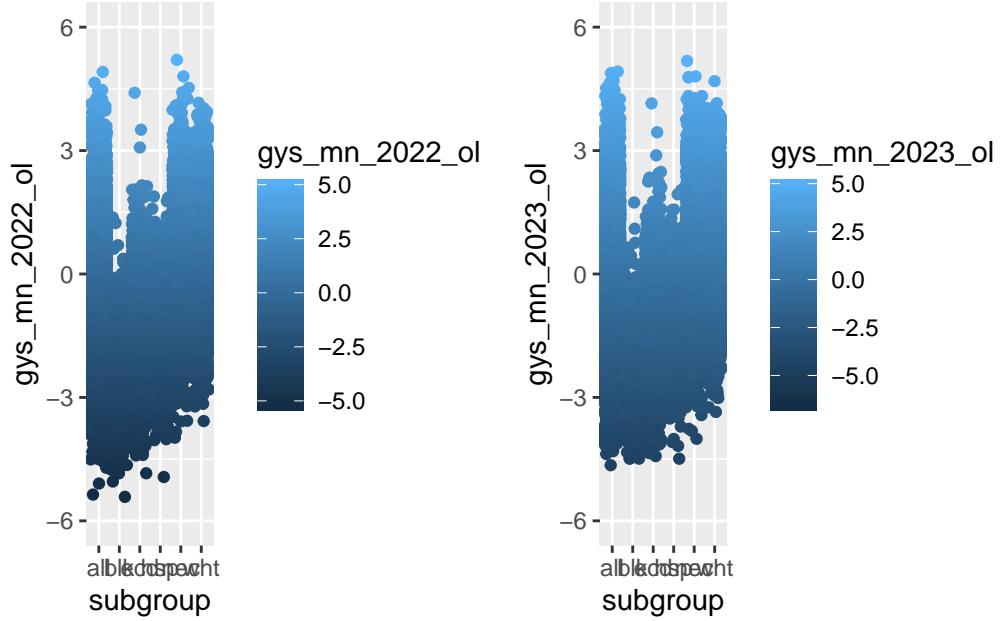


```

ggplot(test_admindist_gys |> filter (subject == "mth")) +
  geom_jitter(
    aes(
      x = subgroup,
      y = gys_mn_2023_ol,
      color = gys_mn_2023_ol
    ),
    ) +
  scale_y_continuous(
    limits = c(-6, 6)
  )
  
```

```
)
```

```
Warning: Removed 4959 rows containing missing values (`geom_point()`).
```



```
mean(na.omit(test_admindist_gys$gys_mn_2022_ol))
```

```
[1] -0.2498951
```

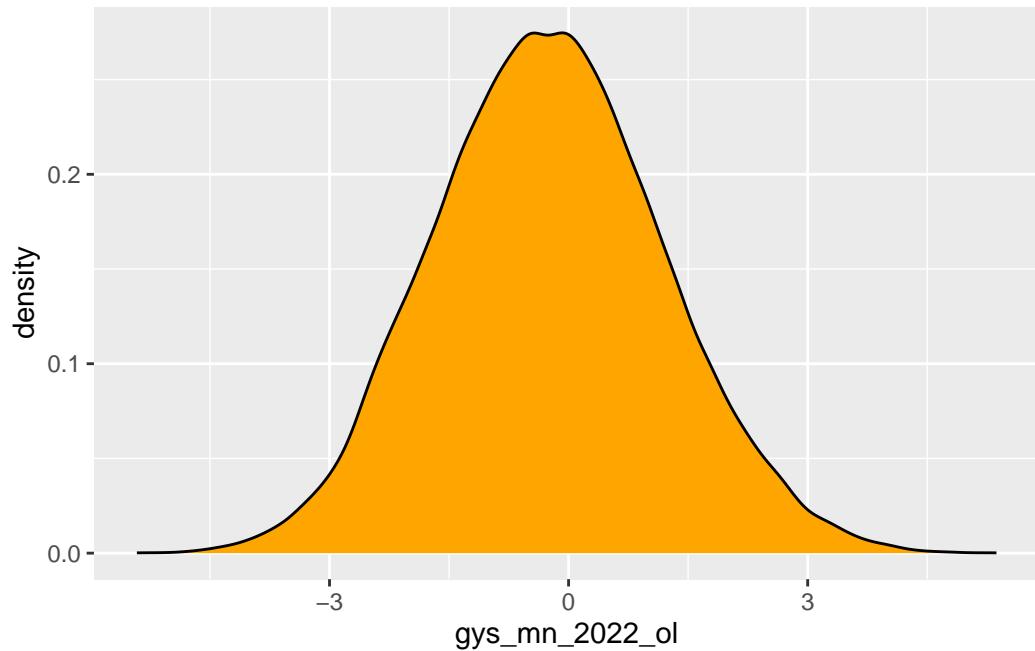
```
median(na.omit(test_admindist_gys$gys_mn_2022_ol))
```

```
[1] -0.2672775
```

```
median(na.omit(test_admindist_gys$gys_mn_2023_ol))
```

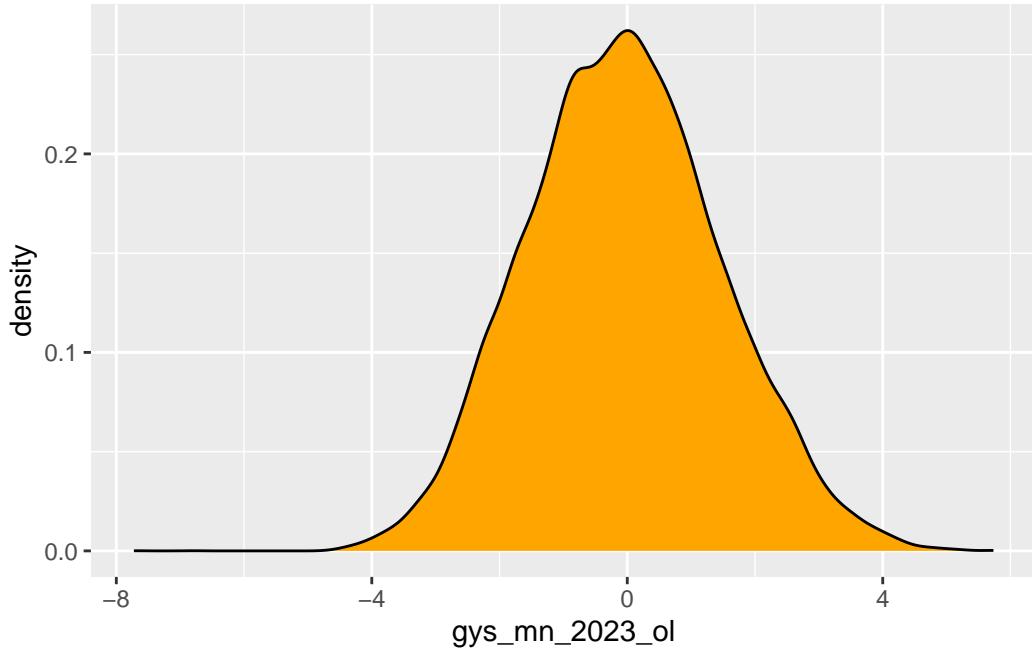
```
[1] -0.0795574
```

```
ggplot(test_admindist_gys) +  
  geom_density(  
    aes(  
      x = gys_mn_2022_ol,  
      ),  
    fill = "orange"  
)
```



```
ggplot(test_admindist_gys) +  
  geom_density(  
    aes(  
      x = gys_mn_2023_ol,  
      ),  
    fill = "orange"  
)
```

Warning: Removed 9374 rows containing non-finite values (`stat_density()`).



More Resources to Consider

Additional Census Data:

- Household income in the past year
[https://data.census.gov/table/ACSST1Y2021.S2503?t=Financial%20Characteristics&g=010XX00US\\$0500000&y=2021&tp=true](https://data.census.gov/table/ACSST1Y2021.S2503?t=Financial%20Characteristics&g=010XX00US$0500000&y=2021&tp=true)
- Age + Race Demographics
[https://data.census.gov/table/ACSDP1Y2021.DP05?g=010XX00US\\$0500000&y=2021&tp=true](https://data.census.gov/table/ACSDP1Y2021.DP05?g=010XX00US$0500000&y=2021&tp=true)
- Characteristics of Foreign Born Population
[https://data.census.gov/table/ACSST1Y2021.S0501?g=010XX00US\\$0500000&y=2021&tp=true](https://data.census.gov/table/ACSST1Y2021.S0501?g=010XX00US$0500000&y=2021&tp=true)

School Budget/School Opening/State Funding

<https://about.burbio.com/#sign-up>

Absenteeism

<https://www.future-ed.org/tracking-state-trends-in-chronic-absenteeism/>

Additional Metrics

- Literacy/Active Reading in Households

- Education level of parents/family
 - Bachelor's %
- General COVID statistics (cases, deaths, etc.)
- Recovery Strategies
 - Look into outliers in the trends and what they are doing
- Inner-district disparities among schools

Referenced Reports/Resources

- <https://www.nytimes.com/interactive/2024/01/31/us/pandemic-learning-loss-recovery.html>
- <https://www.nytimes.com/interactive/2024/02/01/upshot/learning-loss-school-districts.html>
- <https://www.nytimes.com/2023/12/09/opinion/education-learning-loss.html>
- <https://www.nytimes.com/2023/11/17/us/chronic-absenteeism-pandemic-recovery.html>
- <https://educationrecoveryscorecard.org/>
- <https://edopportunity.org/>

Todo:

- See if I can map school district to congressional district
- Try to map congressional districts
- How to handle changing districts
 - Look into other reports
 - Census website
 - Maybe just pick a specific year
 - Take aggregate measurement over time periods
 - Possibly look at the data across years and determine if there's enough variability
- Main Visualization:
 - Scatterplot (based on <https://educationrecoveryscorecard.org/>)
 - * Y: Reading/Math Scores

- * X: Some metric
- * Filters: other metrics (location, income, race, etc.)
- Scatterplot
 - * Y: Reading Score
 - * X: Math score
 - * Filters: metrics
- By next week: Try to finalize datasets and make some versions of the scatterplot
-