PPOL 6819 | Assignment 02

Ethan Sager

2026-01-25

Exercise 02:

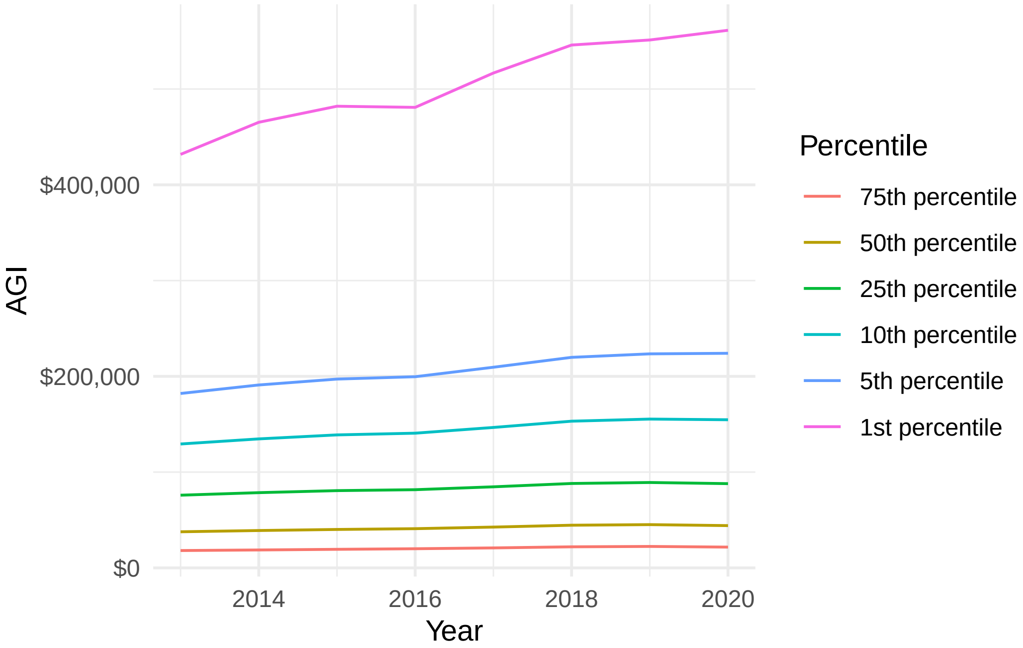
```r
library(tidyverse)
library(here)

here::i_am("analysis/assignment02_es1772.qmd")

# Define years
yrs <- 13:20
# Append each year as {yrs} in url and file name
urls <- str_glue("https://www.irs.gov/pub/irs-soi/{yrs}instateshares.csv")
destinations <- here("data", str_glue("soi20{yrs}.csv"))
# Use walk2 to download files
walk2(urls, destinations, download.file)
```

Exercise 03:

```r
# We already have the destinations vector
# from above so we can use that here
# but need relational path for rendering
# List files to read
files <- list.files(here("data"), full.names = TRUE) %>%
    # We set the names so we can use that for the source
    set_names(str_extract(basename(.), "20\\d{2}"))

# Read and combine files into single data frame
soi <- map(files, read_csv, col_types = cols(statefips = "n")) %>%
    bind_rows(.id = "year") %>%
    mutate(year = as.integer(year)) %>%
    select(
        year,
        statefips,
        state,
        state_name,
        total_agi,
        starts_with("agi")
    )

# Break out into national and state level
# National
soi_national <- soi %>%
    filter(statefips == 0)

# State
soi_state <- soi %>%
    # Drop national level
    filter(statefips != 0) %>%
    # Drop other areas which is only defined in 14,16,17
    filter(statefips != 57)
```

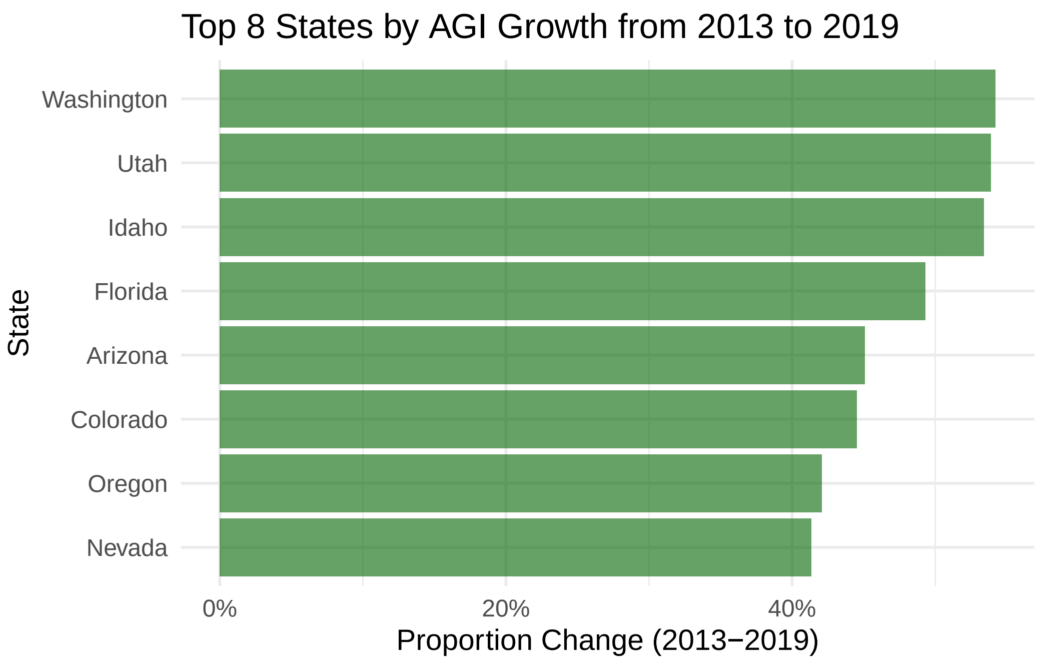
Exercise 04:

```
# Generate a the ggplot for national level
plot_data <- soi_national %>%
    select(year, starts_with("agi")) %>%
    pivot_longer(
        -year,
        names_to = "percentile",
        names_pattern = "agi_(.*)",
        values_to = "agi"
    ) %>%
    mutate(
        percentile = factor(
            percentile,
            levels = c("75", "50", "25", "10", "05", "01"),
            labels = c(
                "75th percentile",
                "50th percentile",
                "25th percentile",
                "10th percentile",
                "5th percentile",
                "1st percentile"
            )
        )
    )

# Graph the data
ggplot(plot_data, aes(x = year, y = agi, color = percentile)) +
    geom_line() +
    scale_y_continuous(labels = scales::dollar_format()) +
    labs(
        x = "Year",
        y = "AGI",
        color = "Percentile"
    ) +
    theme_minimal()
```



Exercise 05:

```r
state_plot <- soi_state %>%
    # Filter to years 2013 and 2019
    filter(year %in% c(2013, 2019)) %>%
    # Arrange by year
    arrange(year) %>%
    # Group by state
    group_by(state_name) %>%
    mutate(
        # Calculate growth from previous year
        growth = (total_agi / lag(total_agi)) - 1
    ) %>%
    ungroup() %>%
    # Filter to 2019 only
    filter(year == 2019) %>%
    # Pull out the 8 max growth states
    slice_max(growth, n = 8) %>%
    # Turn state into factor ordered by growth
    mutate(state_name = fct_reorder(state_name, growth))

ggplot(state_plot, aes(x = growth, y = state_name)) +
    geom_col(fill = "darkgreen", alpha = 0.6) +
    scale_x_continuous(labels = scales::percent_format()) +
    labs(
        x = "Proportion Change (2013-2019)",
        y = "State",
        title = "Top 8 States by AGI Growth from 2013 to 2019"
    ) +
    theme_minimal()
```



Exercise 05:

```
# Note the trick in the wording i.e august 2023 is
# after national lasagne day so skip it

seq(ymd("2024-04-29"), ymd("2100-04-29"), by = "1 years") %>%
    tibble(
        date = .,
        weekday = wday(., label = TRUE, abbr = FALSE),
        year = year(.)
    ) %>%
    filter(weekday == "Monday")
```

```
# A tibble: 11 x 3
   date       weekday  year
   <date>     <ord>   <dbl>
 1 2024-04-29 Monday   2024
 2 2030-04-29 Monday   2030
 3 2041-04-29 Monday   2041
 4 2047-04-29 Monday   2047
 5 2052-04-29 Monday   2052
 6 2058-04-29 Monday   2058
 7 2069-04-29 Monday   2069
 8 2075-04-29 Monday   2075
 9 2080-04-29 Monday   2080
10 2086-04-29 Monday   2086
11 2097-04-29 Monday   2097
```

```r
# Now lets find the 14 yrs with 53 mondays

seq(ymd("2024-01-01"), ymd("2099-12-31"), by = "1 days") %>%
    tibble(
        year = year(.),
        weekday = wday(., label = TRUE, abbr = FALSE)
    ) %>%
    mutate(is_monday = weekday == "Monday") %>%
    group_by(year) %>%
    summarise(monday_count = sum(is_monday)) %>%
    filter(monday_count == 53) %>%
    ungroup() %>%
    select(year)
```

```
# A tibble: 14 x 1
    year
   <dbl>
 1  2024
 2  2029
 3  2035
 4  2040
 5  2046
 6  2052
 7  2057
 8  2063
 9  2068
10  2074
11  2080
12  2085
13  2091
14  2096
```