Problem Set 4

Takuma Kazama

2026-02-07

Due 02/07 at 5:00PM Central.

"This submission is my work alone and complies with the 30538 integrity policy." Add your initials to indicate your agreement: **TK**

Github Classroom Assignment Setup and Submission Instructions

1. **Accepting and Setting up the PS4 Assignment Repository**
   - Each student must individually accept the repository for the problem set from Github Classroom ("ps4") – https://classroom.github.com/a/hWhtcHqH
     - You will be prompted to select your cnetid from the list in order to link your Github account to your cnetid.
     - If you can't find your cnetid in the link above, click "continue to next step" and accept the assignment, then add your name, cnetid, and Github account to this Google Sheet and we will manually link it: https://rb.gy/9u7fb6
   - If you authenticated and linked your Github account to your device, you should be able to clone your PS4 assignment repository locally.
   - Contents of PS4 assignment repository:
     - `ps4_template.qmd`: this is the Quarto file with the template for the problem set. You will write your answers to the problem set here.
2. **Submission Process**:
   - Knit your completed solution `ps4.qmd` as a pdf `ps4.pdf`.
     - Your submission does not need runnable code. Instead, you will tell us either what code you ran or what output you got.
   - To submit, push `ps4.qmd` and `ps4.pdf` to your PS4 assignment repository. Confirm on Github.com that your work was successfully pushed.

Grading

- You will be graded on what was last pushed to your PS4 assignment repository before the assignment deadline
- Problem sets will be graded for completion as: {missing (0%); - (incomplete, 50%); + (excellent, 100%)}
  - The percent values assigned to each problem denote how long we estimate the problem will take as a share of total time spent on the problem set, not the points they are associated with.
- In order for your submission to be considered complete, you need to push both your `ps4.qmd` and `ps4.pdf` to your repository. Submissions that do not include both files will automatically receive 50% credit.

```
RendererRegistry.enable('png')
```

Step 1: Develop initial scraper and crawler

The head of the dataframe is below.

```
                                               title              date  \
0  Houston Transplant Doctor Indicted For Making ...  February 5, 2026   
1  MultiCare Health System to Pay Millions to Set...  February 4, 2026   
2  Brooklyn Banker Pleads Guilty to Laundering Pr...  February 3, 2026   
3  Delafield Man Sentenced to 18 Months' Imprison...  February 3, 2026   
4  Former NFL Player Convicted for $197M Medicare...  February 3, 2026   

                    categories  \
0  [Criminal and Civil Actions]   
1  [Criminal and Civil Actions]   
2                   [COVID-19]   
3  [Criminal and Civil Actions]   
4  [Criminal and Civil Actions]   

                                                link  
0  https://oig.hhs.gov/fraud/enforcement/houston-...  
1  https://oig.hhs.gov/fraud/enforcement/multicar...  
2  https://oig.hhs.gov/fraud/enforcement/brooklyn...  
3  https://oig.hhs.gov/fraud/enforcement/delafiel...  
4  https://oig.hhs.gov/fraud/enforcement/former-n...  
```

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code

The pseudo code is as follows:

FUNCTION dynamic_scraper(start_year, start_month):

```
IF start_year < 2013:
    PRINT warning message
    RETURN empty table with columns
```

```
SET start_date = first day of (start_year, start_month)

INITIALIZE empty list rows
SET page = 1

LOOP forever:
    BUILD page_url using base url and page number
    REQUEST page_url
    PARSE HTML response

    FIND all action blocks on the page

    FOR each action block:
        EXTRACT title
        EXTRACT relative link
        BUILD full link (base domain + relative link)

        EXTRACT date text
        PARSE date text into date object d

        IF d is earlier than start_date:
            RETURN table made from rows
            (stop everything immediately)

        EXTRACT all category labels

        ADD {title, date, categories, full link} to rows

    WAIT 1 second (polite scraping)

    CHECK if a "Next" pagination link exists
    IF no "Next" link exists:
        BREAK out of loop

    INCREMENT page number

RETURN table made from rows
```

ENDFUNCTION

- b. Create Dynamic Scraper

```
1772
```

```
title: Former Nurse Aide Indicted In Death Of Clarksville Patient Arrested In
Georgia
date: January 3, 2024
categories: ['State Enforcement Agencies']
link:
https://oig.hhs.gov/fraud/enforcement/former-nurse-aide-indicted-in-death-of-clarksville-pat
```

The numver of the actions scraped is 1772, and the details of the eraliest entry is seen above.

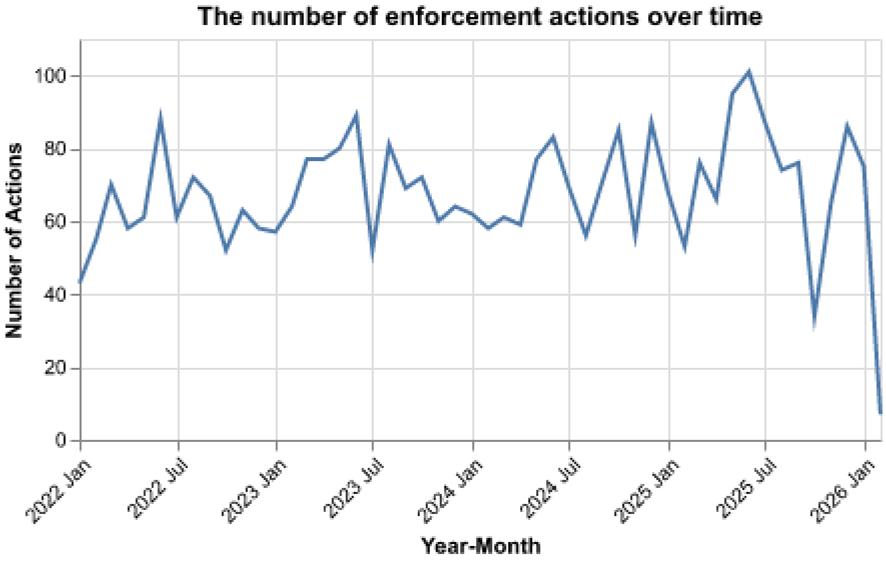
- c. Test Your Code

```
3377
title: Integrated Pain Management Medical Group Agreed to Pay $10,000 for
Allegedly Violating the Civil Monetary Penalties Law by Employing Excluded
Individuals
date: January 4, 2022
categories: ['Fraud Self-Disclosures']
link:
https://oig.hhs.gov/fraud/enforcement/integrated-pain-management-medical-group-agreed-to-pay
```

The numver of the actions scraped is 3377, and the details of the eraliest entry is seen above.

Step 3: Plot data based on scraped data

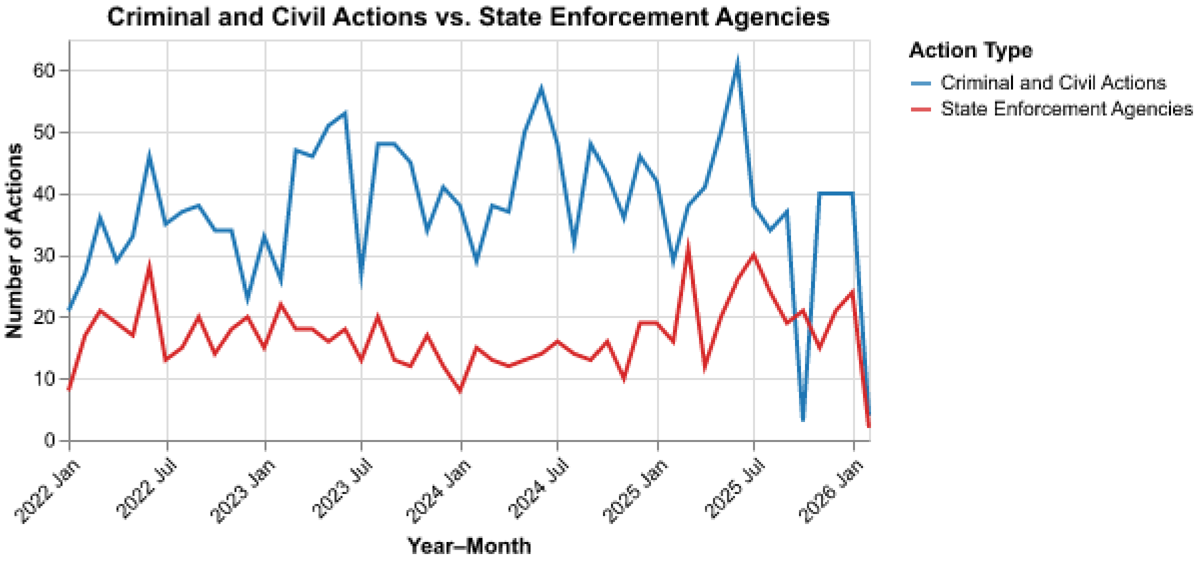
1. Plot the number of enforcement actions over time



The plot is as above. In most of the months, the number lies within the range between 40 and 80.

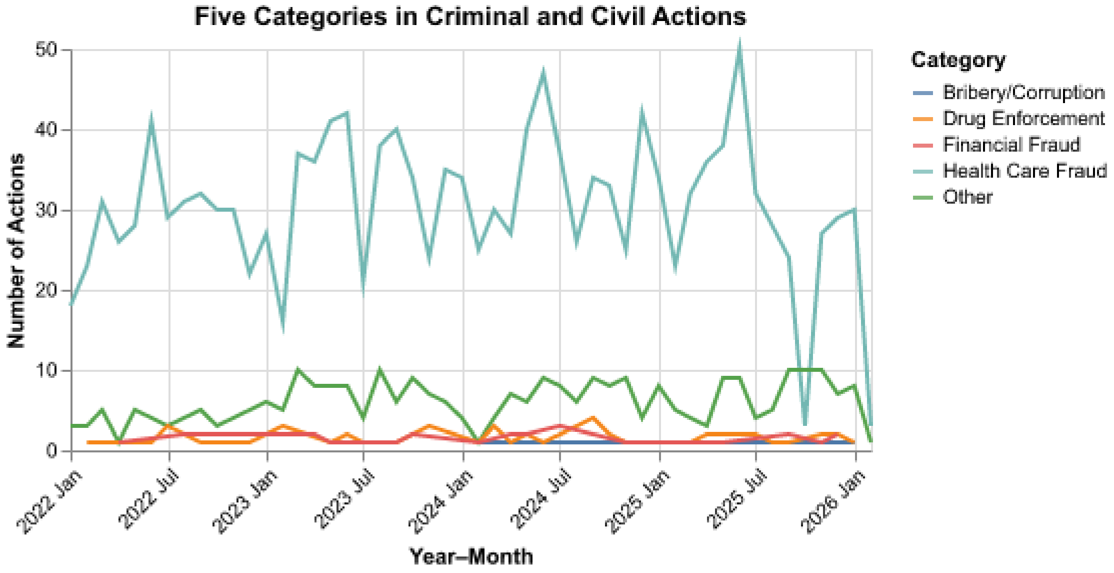
2. Plot the number of enforcement actions categorized:

- based on “Criminal and Civil Actions” vs. “State Enforcement Agencies”



The plot is above. It is noticeable that the number of State Enfrocement Agencies is substantially lower that that of Criminal and Civil Actions almost all the time druing the period.

- based on five topics



The plot is seen above. It is clear that mose of the cases in Criminal and Civil Actions are associated with Health Care Fraud.