PS4

Charlie Schraw

2026-02-03

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: CS

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.

```python
import pandas as pd
import altair as alt
import time
from bs4 import BeautifulSoup
import requests
from urllib.parse import urljoin
from datetime import datetime

import warnings
warnings.filterwarnings('ignore')
alt.renderers.enable("png")
```

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

Step 1: Develop initial scraper and crawler

```python
with open(r'C:\Users\Owner\Documents\Data_Vis\ps4-clschraw\hhs.htm', 'r',
 ↪  encoding='utf-8') as file:
    html_content = file.read()

soup = BeautifulSoup(html_content, 'html.parser')
card_group = soup.find('ul', class_='usa-card-group')
cards = card_group.find_all('li', class_='usa-card') if card_group else []

enforcement_actions = []

for card in cards:
    header = card.find('header', class_='usa-card__header')

    if header:
        heading = header.find('h2', class_='usa-card__heading')

        if heading:
            link_tag = heading.find('a')
            if link_tag:
                href = link_tag.get('href')
                # Add base URL to make full link
                full_link = f"https://oig.hhs.gov{href}"
                title = link_tag.get_text(strip=True)

                date_span = header.find('span', class_='text-base-dark')
                date = date_span.get_text(strip=True) if date_span else None

                tag_elements = header.find_all('li', class_='usa-tag')
                tags = [tag.get_text(strip=True) for tag in tag_elements]
                tags_str = ', '.join(tags) if tags else None

                enforcement_actions.append({
                    'title': title,
                    'link': full_link,
                    'date': date,
                    'tags': tags_str
                })

# Create DataFrame
df = pd.DataFrame(enforcement_actions)
df.head()
```

	title	link
0	Brooklyn Banker Pleads Guilty to Laundering Pr...	https://oig.hhs.gov/fraud/enforcement/brooklyn...
1	Delafield Man Sentenced to 18 Months' Imprison...	https://oig.hhs.gov/fraud/enforcement/delafiel...
2	Former NFL Player Convicted for $197M Medicare...	https://oig.hhs.gov/fraud/enforcement/former-n...
3	AG's Office Secures Indictments Against Peabod...	https://oig.hhs.gov/fraud/enforcement/ags-offi...
4	Florida Man Pleads Guilty to Conspiracy to Vio...	https://oig.hhs.gov/fraud/enforcement/florida-...

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code

FUNCTION scrape_enforcement_actions(year, month, run_scraper=False):

```
# Step 1: Input validation
IF year < 2013:
    PRINT "Error: Please use year >= 2013"
    RETURN

# Step 2: Check if we should run
IF run_scraper == False:
    PRINT "Scraper is disabled. Set run_scraper=True to execute."
    RETURN
```

```
# Step 3: Calculate date range
start_date = create date from (year, month, 1)
end_date = today's date

# Step 4: Initialize collection
all_enforcement_actions = empty list
page_number = 1
base_url = "https://oig.hhs.gov/fraud/enforcement/"

# Step 5: Loop through pages until no more results
WHILE True:
    # Build URL with filters for date and page number
    url = base_url + "?date_from=YYYY-MM-DD&page=" + page_number

    # Fetch the page HTML
    response = GET request to url

    IF response fails:
        PRINT "Error fetching page"
        BREAK

    # Parse HTML
    soup = parse HTML from response

    # Find card group
    card_group = find ul with class 'usa-card-group'

    IF card_group is None OR card_group has no cards:
        PRINT "No more results found"
        BREAK  # Exit the while loop - no more pages

    # Extract cards from this page
    cards = find all li with class 'usa-card' in card_group

    IF cards is empty:
        BREAK  # No cards found, we're done

    # Step 6: Extract data from each card
    FOR each card in cards:
        header = find header in card
        IF header exists:
            heading = find h2 in header
            IF heading exists:
                link_tag = find a tag in heading
                IF link_tag exists:
                    href = get href attribute
                    full_link = base_url + href
                    title = get text from link_tag

                    date_span = find span with date
                    date = get text from date_span

                    # Check if date is within our range
                    IF date >= start_date AND date <= end_date:
                        tag_elements = find all li tags
                        tags = extract text from tags

                        ADD {title, link, date, tags} to
                        all_enforcement_actions
                    ELSE IF date < start_date:
                        # We've gone past our date range, stop completely
                        SET flag to exit outer loop
                        BREAK from card loop

    # Step 7: Check if we should continue to next page
    IF we hit a date before start_date:
        BREAK  # Stop pagination

    # Step 8: Increment page and wait
    page_number = page_number + 1
    SLEEP for 1 second  # Be polite to the server

# Step 9: Convert to DataFrame and save
df = create DataFrame from all_enforcement_actions
filename = f"enforcement_actions_{year}_{month}.csv"
SAVE df to filename

PRINT f"Scraped {len(df)} enforcement actions"
PRINT f"Saved to {filename}"

RETURN df
```

- b. Create Dynamic Scraper

```
def scrape_enforcement_actions(year, month, run_scraper=False):
    # Step 1: Input validation
    if year < 2013:
        print("Error: Please use year >= 2013. Only enforcement actions after
         ↪  2013 are listed.")
        return None

    # Step 2: Check if we should run
    if not run_scraper:
        print("Scraper is disabled. Set run_scraper=True to execute.")
        return None

    # Step 3: Set up date range
    start_date = datetime(year, month, 1)
    end_date = datetime.now()
    start_date_str = start_date.strftime('%Y-%m-%d')

    print(f"Scraping enforcement actions from {start_date_str} to today...")

    # Step 4: Initialize collection
    all_enforcement_actions = []
    page_number = 1
    base_url = "https://oig.hhs.gov"
    search_url = f"{base_url}/fraud/enforcement/"
    earliest_date = None  # Track the earliest date we encounter

    # Step 5: Loop through pages
    while True:
        # Build URL with date filter and pagination
        params_url =
 ↪  f"{search_url}?date_from={start_date_str}&page={page_number}"

        print(f"Fetching page {page_number}...")

        try:
            # Fetch the page
            response = requests.get(params_url)
            response.raise_for_status()  # Raise error for bad status codes

        except requests.RequestException as e:
            print(f"Error fetching page {page_number}: {e}")
            break
        # Parse HTML
        soup = BeautifulSoup(response.content, 'html.parser')

        # Find card group
        card_group = soup.find('ul', class_='usa-card-group')

        if not card_group:
            print(f"No card group found on page {page_number}. Stopping.")
            break

        # Extract cards from this page
        cards = card_group.find_all('li', class_='usa-card')

        if not cards:
            print(f"No cards found on page {page_number}. Stopping.")
            break

        print(f"Found {len(cards)} cards on page {page_number}")

        # Flag to track if we've gone past our date range
        past_date_range = False

        # Step 6: Extract data from each card
        for card in cards:
            header = card.find('header', class_='usa-card__header')

            if header:
                heading = header.find('h2', class_='usa-card__heading')

                if heading:
                    link_tag = heading.find('a')
                    if link_tag:
                        href = link_tag.get('href')
                        full_link = f"{base_url}{href}"
                        title = link_tag.get_text(strip=True)

                        # Get date
                        date_span = header.find('span',
↪   class_='text-base-dark')
                        date_str = date_span.get_text(strip=True) if
↪   date_span else None

                        if date_str:
                            try:
                                # Parse date (format: "February 3, 2026")
                                card_date = datetime.strptime(date_str, '%B
↪   %d, %Y')

                                # Update earliest date
                                if earliest_date is None or card_date <
                                 ↪  earliest_date:
                                    earliest_date = card_date

                                # Check if date is within our range
                                if card_date >= start_date and card_date <=
                                 ↪  end_date:
                                    # Get tags
                                    tag_elements = header.find_all('li',
↪   class_='usa-tag')
                                    tags = [tag.get_text(strip=True) for tag
↪   in tag_elements]
                                    tags_str = ', '.join(tags) if tags else
↪   None

                                    # Add to collection
                                    all_enforcement_actions.append({
                                        'title': title,
                                        'link': full_link,
                                        'date': date_str,
                                        'tags': tags_str
                                    })

                                elif card_date < start_date:
                                    # We've gone past our date range
                                    past_date_range = True
                                    break

                            except ValueError:
                                # If date parsing fails, skip this card
                                print(f"Could not parse date: {date_str}")
                                continue

        # Step 7: Check if we should continue to next page
        if past_date_range:
            break
        # Step 8: Increment page and wait
        page_number += 1
        time.sleep(1)  # Be polite to the server

    # Step 9: Convert to DataFrame and save
    if not all_enforcement_actions:
        print("No enforcement actions found in the specified date range.")
        return None

    df = pd.DataFrame(all_enforcement_actions)
    filename = f"enforcement_actions_{year}_{month}.csv"
    df.to_csv(filename, index=False)

    print(f"\nScraped {len(df)} enforcement actions")
    if earliest_date:
        print(f"Earliest date encountered: {earliest_date.strftime('%B %d,
         ↪  %Y')}")
    print(f"Saved to {filename}")

    return df
```

```
df = scrape_enforcement_actions(2024, 1, run_scraper=True)
```

```
Scraping enforcement actions from 2024-01-01 to today...
Fetching page 1...
Found 20 cards on page 1
Fetching page 2...
Found 20 cards on page 2
Fetching page 3...
Found 20 cards on page 3
Fetching page 4...
Found 20 cards on page 4
Fetching page 5...
Found 20 cards on page 5
Fetching page 6...
Found 20 cards on page 6
Fetching page 7...
Found 20 cards on page 7
Fetching page 8...
Found 20 cards on page 8
Fetching page 9...
```

```
Found 20 cards on page 9
Fetching page 10...
Found 20 cards on page 10
Fetching page 11...
Found 20 cards on page 11
Fetching page 12...
Found 20 cards on page 12
Fetching page 13...
Found 20 cards on page 13
Fetching page 14...
Found 20 cards on page 14
Fetching page 15...
Found 20 cards on page 15
Fetching page 16...
Found 20 cards on page 16
Fetching page 17...
Found 20 cards on page 17
Fetching page 18...
Found 20 cards on page 18
Fetching page 19...
Found 20 cards on page 19
Fetching page 20...
Found 20 cards on page 20
Fetching page 21...
Found 20 cards on page 21
Fetching page 22...
Found 20 cards on page 22
Fetching page 23...
Found 20 cards on page 23
Fetching page 24...
Found 20 cards on page 24
Fetching page 25...
Found 20 cards on page 25
Fetching page 26...
Found 20 cards on page 26
Fetching page 27...
Found 20 cards on page 27
Fetching page 28...
Found 20 cards on page 28
Fetching page 29...
Found 20 cards on page 29
Fetching page 30...
Found 20 cards on page 30
```

```
Fetching page 31...
Found 20 cards on page 31
Fetching page 32...
Found 20 cards on page 32
Fetching page 33...
Found 20 cards on page 33
Fetching page 34...
Found 20 cards on page 34
Fetching page 35...
Found 20 cards on page 35
Fetching page 36...
Found 20 cards on page 36
Fetching page 37...
Found 20 cards on page 37
Fetching page 38...
Found 20 cards on page 38
Fetching page 39...
Found 20 cards on page 39
Fetching page 40...
Found 20 cards on page 40
Fetching page 41...
Found 20 cards on page 41
Fetching page 42...
Found 20 cards on page 42
Fetching page 43...
Found 20 cards on page 43
Fetching page 44...
Found 20 cards on page 44
Fetching page 45...
Found 20 cards on page 45
Fetching page 46...
Found 20 cards on page 46
Fetching page 47...
Found 20 cards on page 47
Fetching page 48...
Found 20 cards on page 48
Fetching page 49...
Found 20 cards on page 49
Fetching page 50...
Found 20 cards on page 50
Fetching page 51...
Found 20 cards on page 51
Fetching page 52...
```

```
Found 20 cards on page 52
Fetching page 53...
Found 20 cards on page 53
Fetching page 54...
Found 20 cards on page 54
Fetching page 55...
Found 20 cards on page 55
Fetching page 56...
Found 20 cards on page 56
Fetching page 57...
Found 20 cards on page 57
Fetching page 58...
Found 20 cards on page 58
Fetching page 59...
Found 20 cards on page 59
Fetching page 60...
Found 20 cards on page 60
Fetching page 61...
Found 20 cards on page 61
Fetching page 62...
Found 20 cards on page 62
Fetching page 63...
Found 20 cards on page 63
Fetching page 64...
Found 20 cards on page 64
Fetching page 65...
Found 20 cards on page 65
Fetching page 66...
Found 20 cards on page 66
Fetching page 67...
Found 20 cards on page 67
Fetching page 68...
Found 20 cards on page 68
Fetching page 69...
Found 20 cards on page 69
Fetching page 70...
Found 20 cards on page 70
Fetching page 71...
Found 20 cards on page 71
Fetching page 72...
Found 20 cards on page 72
Fetching page 73...
Found 20 cards on page 73
```

```
Fetching page 74...
Found 20 cards on page 74
Fetching page 75...
Found 20 cards on page 75
Fetching page 76...
Found 20 cards on page 76
Fetching page 77...
Found 20 cards on page 77
Fetching page 78...
Found 20 cards on page 78
Fetching page 79...
Found 20 cards on page 79
Fetching page 80...
Found 20 cards on page 80
Fetching page 81...
Found 20 cards on page 81
Fetching page 82...
Found 20 cards on page 82
Fetching page 83...
Found 20 cards on page 83
Fetching page 84...
Found 20 cards on page 84
Fetching page 85...
Found 20 cards on page 85
Fetching page 86...
Found 20 cards on page 86
Fetching page 87...
Found 20 cards on page 87
Fetching page 88...
Found 20 cards on page 88
Fetching page 89...
Found 20 cards on page 89

Scraped 1770 enforcement actions
Earliest date encountered: December 22, 2023
Saved to enforcement_actions_2024_1.csv
```

It took 1770 enforcement actions and the earliest date it encountered was December 22, 2023, but it did not account for it.

- c. Test Your Code

Enforcements since January 2022

```
df = scrape_enforcement_actions(2022, 1, run_scraper=True)
```

```
Scraping enforcement actions from 2022-01-01 to today...
Fetching page 1...
Found 20 cards on page 1
Fetching page 2...
Found 20 cards on page 2
Fetching page 3...
Found 20 cards on page 3
Fetching page 4...
Found 20 cards on page 4
Fetching page 5...
Found 20 cards on page 5
Fetching page 6...
Found 20 cards on page 6
Fetching page 7...
Found 20 cards on page 7
Fetching page 8...
Found 20 cards on page 8
Fetching page 9...
Found 20 cards on page 9
Fetching page 10...
Found 20 cards on page 10
Fetching page 11...
Found 20 cards on page 11
Fetching page 12...
Found 20 cards on page 12
Fetching page 13...
Found 20 cards on page 13
Fetching page 14...
Found 20 cards on page 14
Fetching page 15...
Found 20 cards on page 15
Fetching page 16...
Found 20 cards on page 16
Fetching page 17...
Found 20 cards on page 17
Fetching page 18...
Found 20 cards on page 18
Fetching page 19...
Found 20 cards on page 19
Fetching page 20...
Found 20 cards on page 20
```

```
Fetching page 21...
Found 20 cards on page 21
Fetching page 22...
Found 20 cards on page 22
Fetching page 23...
Found 20 cards on page 23
Fetching page 24...
Found 20 cards on page 24
Fetching page 25...
Found 20 cards on page 25
Fetching page 26...
Found 20 cards on page 26
Fetching page 27...
Found 20 cards on page 27
Fetching page 28...
Found 20 cards on page 28
Fetching page 29...
Found 20 cards on page 29
Fetching page 30...
Found 20 cards on page 30
Fetching page 31...
Found 20 cards on page 31
Fetching page 32...
Found 20 cards on page 32
Fetching page 33...
Found 20 cards on page 33
Fetching page 34...
Found 20 cards on page 34
Fetching page 35...
Found 20 cards on page 35
Fetching page 36...
Found 20 cards on page 36
Fetching page 37...
Found 20 cards on page 37
Fetching page 38...
Found 20 cards on page 38
Fetching page 39...
Found 20 cards on page 39
Fetching page 40...
Found 20 cards on page 40
Fetching page 41...
Found 20 cards on page 41
Fetching page 42...
```

```
Found 20 cards on page 42
Fetching page 43...
Found 20 cards on page 43
Fetching page 44...
Found 20 cards on page 44
Fetching page 45...
Found 20 cards on page 45
Fetching page 46...
Found 20 cards on page 46
Fetching page 47...
Found 20 cards on page 47
Fetching page 48...
Found 20 cards on page 48
Fetching page 49...
Found 20 cards on page 49
Fetching page 50...
Found 20 cards on page 50
Fetching page 51...
Found 20 cards on page 51
Fetching page 52...
Found 20 cards on page 52
Fetching page 53...
Found 20 cards on page 53
Fetching page 54...
Found 20 cards on page 54
Fetching page 55...
Found 20 cards on page 55
Fetching page 56...
Found 20 cards on page 56
Fetching page 57...
Found 20 cards on page 57
Fetching page 58...
Found 20 cards on page 58
Fetching page 59...
Found 20 cards on page 59
Fetching page 60...
Found 20 cards on page 60
Fetching page 61...
Found 20 cards on page 61
Fetching page 62...
Found 20 cards on page 62
Fetching page 63...
Found 20 cards on page 63
```

```
Fetching page 64...
Found 20 cards on page 64
Fetching page 65...
Found 20 cards on page 65
Fetching page 66...
Found 20 cards on page 66
Fetching page 67...
Found 20 cards on page 67
Fetching page 68...
Found 20 cards on page 68
Fetching page 69...
Found 20 cards on page 69
Fetching page 70...
Found 20 cards on page 70
Fetching page 71...
Found 20 cards on page 71
Fetching page 72...
Found 20 cards on page 72
Fetching page 73...
Found 20 cards on page 73
Fetching page 74...
Found 20 cards on page 74
Fetching page 75...
Found 20 cards on page 75
Fetching page 76...
Found 20 cards on page 76
Fetching page 77...
Found 20 cards on page 77
Fetching page 78...
Found 20 cards on page 78
Fetching page 79...
Found 20 cards on page 79
Fetching page 80...
Found 20 cards on page 80
Fetching page 81...
Found 20 cards on page 81
Fetching page 82...
Found 20 cards on page 82
Fetching page 83...
Found 20 cards on page 83
Fetching page 84...
Found 20 cards on page 84
Fetching page 85...
```

```
Found 20 cards on page 85
Fetching page 86...
Found 20 cards on page 86
Fetching page 87...
Found 20 cards on page 87
Fetching page 88...
Found 20 cards on page 88
Fetching page 89...
Found 20 cards on page 89
Fetching page 90...
Found 20 cards on page 90
Fetching page 91...
Found 20 cards on page 91
Fetching page 92...
Found 20 cards on page 92
Fetching page 93...
Found 20 cards on page 93
Fetching page 94...
Found 20 cards on page 94
Fetching page 95...
Found 20 cards on page 95
Fetching page 96...
Found 20 cards on page 96
Fetching page 97...
Found 20 cards on page 97
Fetching page 98...
Found 20 cards on page 98
Fetching page 99...
Found 20 cards on page 99
Fetching page 100...
Found 20 cards on page 100
Fetching page 101...
Found 20 cards on page 101
Fetching page 102...
Found 20 cards on page 102
Fetching page 103...
Found 20 cards on page 103
Fetching page 104...
Found 20 cards on page 104
Fetching page 105...
Found 20 cards on page 105
Fetching page 106...
Found 20 cards on page 106
```

```
Fetching page 107...
Found 20 cards on page 107
Fetching page 108...
Found 20 cards on page 108
Fetching page 109...
Found 20 cards on page 109
Fetching page 110...
Found 20 cards on page 110
Fetching page 111...
Found 20 cards on page 111
Fetching page 112...
Found 20 cards on page 112
Fetching page 113...
Found 20 cards on page 113
Fetching page 114...
Found 20 cards on page 114
Fetching page 115...
Found 20 cards on page 115
Fetching page 116...
Found 20 cards on page 116
Fetching page 117...
Found 20 cards on page 117
Fetching page 118...
Found 20 cards on page 118
Fetching page 119...
Found 20 cards on page 119
Fetching page 120...
Found 20 cards on page 120
Fetching page 121...
Found 20 cards on page 121
Fetching page 122...
Found 20 cards on page 122
Fetching page 123...
Found 20 cards on page 123
Fetching page 124...
Found 20 cards on page 124
Fetching page 125...
Found 20 cards on page 125
Fetching page 126...
Found 20 cards on page 126
Fetching page 127...
Found 20 cards on page 127
Fetching page 128...
```

```
Found 20 cards on page 128
Fetching page 129...
Found 20 cards on page 129
Fetching page 130...
Found 20 cards on page 130
Fetching page 131...
Found 20 cards on page 131
Fetching page 132...
Found 20 cards on page 132
Fetching page 133...
Found 20 cards on page 133
Fetching page 134...
Found 20 cards on page 134
Fetching page 135...
Found 20 cards on page 135
Fetching page 136...
Found 20 cards on page 136
Fetching page 137...
Found 20 cards on page 137
Fetching page 138...
Found 20 cards on page 138
Fetching page 139...
Found 20 cards on page 139
Fetching page 140...
Found 20 cards on page 140
Fetching page 141...
Found 20 cards on page 141
Fetching page 142...
Found 20 cards on page 142
Fetching page 143...
Found 20 cards on page 143
Fetching page 144...
Found 20 cards on page 144
Fetching page 145...
Found 20 cards on page 145
Fetching page 146...
Found 20 cards on page 146
Fetching page 147...
Found 20 cards on page 147
Fetching page 148...
Found 20 cards on page 148
Fetching page 149...
Found 20 cards on page 149
```

```
Fetching page 150...
Found 20 cards on page 150
Fetching page 151...
Found 20 cards on page 151
Fetching page 152...
Found 20 cards on page 152
Fetching page 153...
Found 20 cards on page 153
Fetching page 154...
Found 20 cards on page 154
Fetching page 155...
Found 20 cards on page 155
Fetching page 156...
Found 20 cards on page 156
Fetching page 157...
Found 20 cards on page 157
Fetching page 158...
Found 20 cards on page 158
Fetching page 159...
Found 20 cards on page 159
Fetching page 160...
Found 20 cards on page 160
Fetching page 161...
Found 20 cards on page 161
Fetching page 162...
Found 20 cards on page 162
Fetching page 163...
Found 20 cards on page 163
Fetching page 164...
Found 20 cards on page 164
Fetching page 165...
Found 20 cards on page 165
Fetching page 166...
Found 20 cards on page 166
Fetching page 167...
Found 20 cards on page 167
Fetching page 168...
Found 20 cards on page 168
Fetching page 169...
Found 20 cards on page 169

Scraped 3360 enforcement actions
Earliest date encountered: December 30, 2021
```

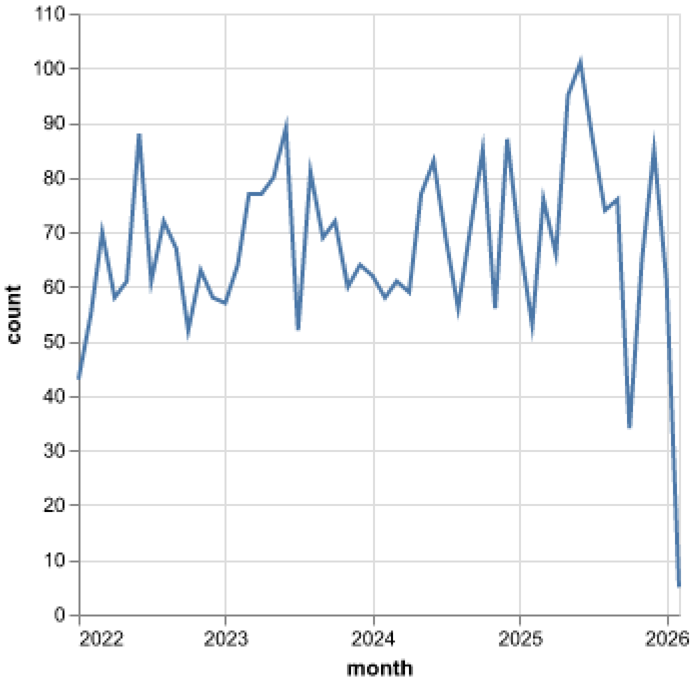
```
Saved to enforcement_actions_2022_1.csv
```

It took 3360 enforcement actions and the earliest date it encountered was December 30, 2021, but it did not account for it.

Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time

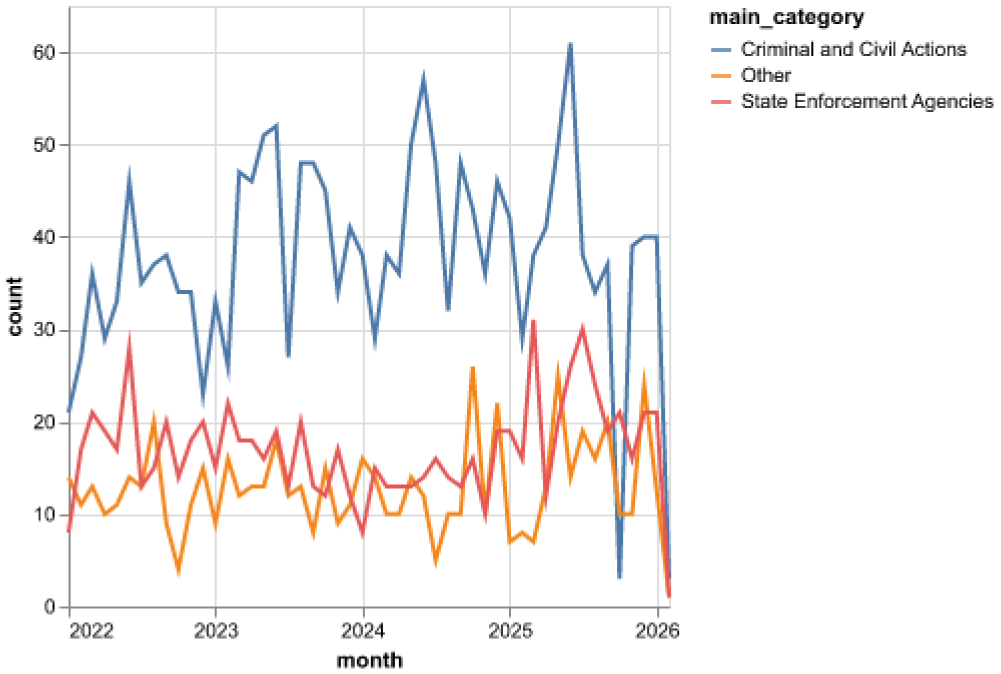
```
df =
↪   pd.read_csv(r'C:\Users\Owner\Documents\Data_Vis\ps4-clschraw\enforcement_actions_2022_1.c
df['date'] = pd.to_datetime(df['date'])
df = df[df['date'] >= '2022-01-01']
df['month'] = df['date'].dt.to_period('M').dt.to_timestamp()

monthly = df.groupby('month').size().reset_index(name='count')

chart = alt.Chart(monthly).mark_line().encode(
    x=alt.X('month:T'),
    y=alt.Y('count:Q')
)
chart.show()
```



2. Plot the number of enforcement actions categorized:

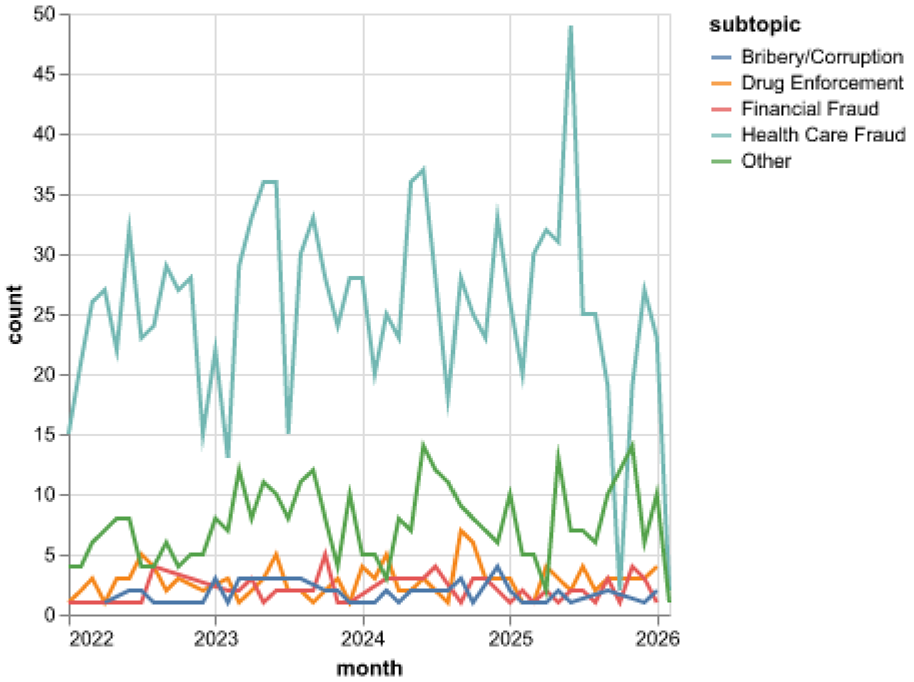
- based on “Criminal and Civil Actions” vs. “State Enforcement Agencies”

```
def classify_main_category(tags):
    tags = str(tags).lower()
    if 'state enforcement agencies' in tags:
        return 'State Enforcement Agencies'
    elif 'criminal and civil actions' in tags:
        return 'Criminal and Civil Actions'
    return 'Other'

df['main_category'] = df['tags'].apply(classify_main_category)

monthly_main = df.groupby(['month',
 ↪  'main_category']).size().reset_index(name='count')

chart = alt.Chart(monthly_main).mark_line().encode(
    x=alt.X('month:T'),
    y=alt.Y('count:Q'),
    color=alt.Color('main_category:N')
)

chart.show()
```



- based on five topics

```
df['title_lower'] = df['title'].str.lower()

# Filter for Criminal and Civil Actions only
df = df[df['tags'].str.contains('Criminal and Civil Actions', na=False)]

# Classify into subtopics
def classify_subtopic(title):
    # Health Care Fraud
    if any(word in title for word in ['health care', 'healthcare', 'medical',
    ↪   'medicare', 'medicaid', 'hospital', 'doctor', 'physician', 'clinic',
    ↪   'patient']):
        return 'Health Care Fraud'

    # Drug Enforcement
    elif any(word in title for word in ['drug', 'opioid', 'fentanyl',
    ↪   'controlled substance', 'prescription', 'pharmacy', 'narcotics']):
        return 'Drug Enforcement'

    # Bribery/Corruption
    elif any(word in title for word in ['bribery', 'bribe', 'corruption',
     ↪  'kickback', 'extortion']):
        return 'Bribery/Corruption'

    # Financial Fraud
    elif any(word in title for word in ['bank', 'financial', 'securities',
     ↪  'wire fraud', 'money laundering', 'fraud scheme', 'investment',
     ↪  'loan', 'mortgage', 'tax', 'embezzle']):
        return 'Financial Fraud'

    # Other
    else:
        return 'Other'

df['subtopic'] = df['title_lower'].apply(classify_subtopic)

# Aggregate by month and subtopic
monthly_subtopic = df.groupby(['month',
 ↪  'subtopic']).size().reset_index(name='count')

# Create chart
chart = alt.Chart(monthly_subtopic).mark_line().encode(
    x=alt.X('month:T'),
    y=alt.Y('count:Q'),
    color=alt.Color('subtopic:N')
)

chart.show()
```


count
50
45
40
35
30
25
20
15
10
5
0
2022
2023
2024
2025
2026
month
subtopic
Bribery/Corruption
Drug Enforcement
Financial Fraud
Health Care Fraud
Other