PS4

Nandini Krishnan

2026-02-06

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: NK

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.

```
import pandas as pd
import altair as alt
import time

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

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

Step 1: Develop initial scraper and crawler

I used AI to generate the code below. the prompt is as follows:

okay with inspection i can find that the info i need is in the tag

with class=“usa-card card–list pep-card–minimal mobile:grid-col-12” this is the structure:

AG Murrill’s Medicaid Fraud Control Unit Secures Guilty Plea For Alexandria Man For Dragging A Victim Across A Hallway By His Wrist

January 16, 2026

State Enforcement Agencies

Title of enforcement action: AG Murrill’s Medicaid Fraud Control Unit Secures Guilty Plea For Alexandria Man For Dragging A Victim Across A Hallway By His Wrist Date: January 16, 2026 Category: State Enforcement Agencies Link: “/fraud/enforcement/ag-murrills-medicaid-fraud-control-unit-secures-guilty-plea-for-alexandria-man-for-dragging-a-victim-across-a-hallway-by-his-wrist/” can you write a python code to scrape this using beautiful soup

```
import requests
from bs4 import BeautifulSoup

# URL for HHS OIG Enforcement Actions (first page)
url =
 ↪  "https://oig.hhs.gov/fraud/enforcement/?text=enforcement&action-details-date=all#results"

# Get the page
response = requests.get(url)
response.raise_for_status()  # fail loudly if request breaks

# Parse HTML
```

```python
soup = BeautifulSoup(response.text, "html.parser")

# Print the first ~2000 characters so you can see the structure
print(soup.prettify()[:50])
```

```
<!DOCTYPE html>
<html class="no-js" lang="en">
 <h
```

```python
import requests
import pandas as pd
from bs4 import BeautifulSoup

url =
 ↪  "https://oig.hhs.gov/fraud/enforcement/?text=enforcement&action-details-date=all#results"
response = requests.get(url)
response.raise_for_status()

soup = BeautifulSoup(response.text, "html.parser")

rows = []

cards = soup.find_all(
    "li",
    class_="usa-card card--list pep-card--minimal mobile:grid-col-12"
)

for card in cards:
    title_tag = card.select_one("h2.usa-card__heading a")

    title = title_tag.get_text(strip=True)
    link = title_tag["href"]

    date = card.select_one(
        "span.text-base-dark.padding-right-105"
    ).get_text(strip=True)

    category = card.select_one(
        "li.usa-tag"
    ).get_text(strip=True)

    rows.append({
        "title": title,
        "date": pd.to_datetime(date),
        "category": category,
        "link": link
    })

df = pd.DataFrame(rows)

print(df)
```

```
                                                title       date  \
0   Attorney General Hanaway Obtains Medicaid Frau... 2026-02-03
1   AG's Office Secures Indictments Against Peabod... 2026-02-02
2   Attorney General Jeff Jackson Announces Health... 2026-01-30
3   Attorney General Labrador Announces Sentencing... 2026-01-29
4   Attorney General Hanaway Obtains Medicaid Frau... 2026-01-29
5   Attorney General James Uthmeier Announces Arre... 2026-01-28
6   Attorney General Jeff Jackson Announces $8.8 M... 2026-01-27
7   Attorney General Alan Wilson Announces Nurse A... 2026-01-27
8   Two Women Charged with Health Care Fraud for A... 2026-01-26
9   TennCare Fraud Investigation Leads to Arrests ... 2026-01-26
10  Attorney General Hanaway Secures Medicaid Frau... 2026-01-23
11  Former Saginaw Physician to Stand Trial on 23 ... 2026-01-22
12  Attorney General Nessel Charges Former CEO of ... 2026-01-20
13  AG Murrill's Medicaid Fraud Control Unit Secur... 2026-01-16
14  Eye Care Clinics Agree To Pay Combined $520K O... 2026-01-15
15  Attorney General Jeff Jackson Announces $15.6 ... 2026-01-15
16  Former Memphis Caregiver Arrested, Charged wit... 2026-01-15
17  Attorney General Ellison Announces Over $3 Mil... 2026-01-14
18  Attorney General James Announces Conviction an... 2026-01-14
19  Attorney General Alan Wilson Announces Chesnee... 2026-01-14

                      category  \
0   State Enforcement Agencies
1   State Enforcement Agencies
2   State Enforcement Agencies
3   State Enforcement Agencies
4   State Enforcement Agencies
5   State Enforcement Agencies
6   State Enforcement Agencies
7   State Enforcement Agencies
8   State Enforcement Agencies
9   State Enforcement Agencies
```

```
10  State Enforcement Agencies
11  State Enforcement Agencies
12  State Enforcement Agencies
13  State Enforcement Agencies
14  State Enforcement Agencies
15  State Enforcement Agencies
16  State Enforcement Agencies
17  State Enforcement Agencies
18  State Enforcement Agencies
19  State Enforcement Agencies

                                                 link
0   /fraud/enforcement/attorney-general-hanaway-ob...
1   /fraud/enforcement/ags-office-secures-indictme...
2   /fraud/enforcement/attorney-general-jeff-jacks...
3   /fraud/enforcement/attorney-general-labrador-a...
4   /fraud/enforcement/attorney-general-hanaway-ob...
5   /fraud/enforcement/attorney-general-james-uthm...
6   /fraud/enforcement/attorney-general-jeff-jacks...
7   /fraud/enforcement/attorney-general-alan-wilso...
8   /fraud/enforcement/two-women-charged-with-heal...
9   /fraud/enforcement/tenncare-fraud-investigatio...
10  /fraud/enforcement/attorney-general-hanaway-se...
11  /fraud/enforcement/former-saginaw-physician-to...
12  /fraud/enforcement/attorney-general-nessel-cha...
13  /fraud/enforcement/ag-murrills-medicaid-fraud-...
14  /fraud/enforcement/eye-care-clinics-agree-to-p...
15  /fraud/enforcement/attorney-general-jeff-jacks...
16  /fraud/enforcement/former-memphis-caregiver-ar...
17  /fraud/enforcement/attorney-general-ellison-an...
18  /fraud/enforcement/attorney-general-james-anno...
19  /fraud/enforcement/attorney-general-alan-wilso...
```

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code

1. Inputs

start_year

start_month

run_scraper (TRUE/FALSE indicator)

2. Validation

If start_year < 2013, print warning and exit.

3. Initialize

Base URL

Empty list to store results

Page counter starting at 1

A flag like keep_scraping = True

4. Loop

Use a while loop

You stop only when: There are no more enforcement cards or all actions on the page are earlier than your start date

5. Inside the loop

Request page ?page=N

Parse enforcement cards

– For each card:

Extract title, link, date, category

Convert date to datetime

Keep only rows   start_year/start_month

–If no rows on this page meet the condition, stop looping

–Otherwise:

Append rows

page += 1

sleep(1) before next request

6. Finalize

Convert results list to DataFrame

Save as enforcement_actions___.csv

- b. Create Dynamic Scraper I used AI here especially to figure out the page change. I am still struggling at the part about why i cant write a for loop to the request function when i know the pattern for the page change url. ChatGPT was very much againts it and i dont understand the logic.

Additionally i think the better way to do this may be would be to use the filter in the website manually, get the filtered url and then scrape, otherwise it is running for too long

```python
def scrape_enforcement_custom_safe(start_date_str, end_date_str=None, run_scraper=False):
    import requests, pandas as pd, time
    from bs4 import BeautifulSoup
    from datetime import datetime

    if not run_scraper:
        print("Scraper indicator is OFF. Skipping scrape.")
        return None

    if end_date_str is None:
        end_date_str = datetime.today().strftime("%m/%d/%Y")

    base_url = "https://oig.hhs.gov/fraud/enforcement/"
    page = 1
    rows = []
    seen_links = set()  # track all scraped links to detect last page

    while True:
        print(f"Scraping page {page} ...")
        params = {
            "action-details-date": "custom",
            "from_date-action-details-date": start_date_str,
            "to_date-action-details-date": end_date_str,
            "page": page
        }

        response = requests.get(base_url, params=params)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, "html.parser")

        cards = soup.find_all(
            "li", class_="usa-card card--list pep-card--minimal mobile:grid-col-12"
        )
        if not cards:
            break

        new_links_this_page = []

        for card in cards:
            try:
                title_tag = card.select_one("h2.usa-card__heading a")
                title = title_tag.get_text(strip=True)
                link = title_tag["href"]

                date_text = card.select_one("span.text-base-dark.padding-right-105").get_text(strip=True)
                date = pd.to_datetime(date_text)

                category = card.select_one("li.usa-tag").get_text(strip=True)

                if link not in seen_links:
                    rows.append({
                        "title": title,
                        "date": date,
                        "category": category,
                        "link": link
                    })
                    new_links_this_page.append(link)
                    seen_links.add(link)
            except Exception:
                continue

        # If no new links found on this page, stop scraping
        if not new_links_this_page:
            print("No new links found; reached last page.")
            break

        page += 1
        time.sleep(1)

    result = pd.DataFrame(rows).sort_values("date", ascending=False)
    filename = f"enforcement_actions_{start_date_str.replace('/', '-')}_to_{end_date_str.replace('/', '-')}.csv"
    result.to_csv(filename, index=False)
    print(f"Scraping complete: {len(result)} records saved to {filename}")
```

```
    return result



# Example: scrape actions from Jan 1, 2024 to today
result=scrape_enforcement_custom_safe(start_date_str="01/01/2024",
 ↪  run_scraper=False)

#result.to_csv("/Users/nandinikrishnan/Documents/GitHub/Data_vis/ps4-nandinikrishnan1/data_20
 ↪  index=False)

data_2024=
 ↪  pd.read_csv("/Users/nandinikrishnan/Documents/GitHub/Data_vis/ps4-nandinikrishnan1/data_2

data_2024['date'] = pd.to_datetime(data_2024['date'])

num_actions = len(data_2024)
print(f"Number of enforcement actions: {num_actions}")

# Earliest enforcement action
earliest = data_2024.sort_values("date").iloc[0]  # earliest date is first
 ↪  row after sorting
print("Earliest enforcement action:")
print(f"Date: {earliest['date'].date()}")
print(f"Title: {earliest['title']}")
print(f"Category: {earliest['category']}")
print(f"Link: {earliest['link']}")
```

```
Scraper indicator is OFF. Skipping scrape.
Number of enforcement actions: 1787
Earliest enforcement action:
Date: 2024-01-03
Title: Former Nurse Aide Indicted In Death Of Clarksville Patient Arrested In
Georgia
Category: State Enforcement Agencies
Link:
/fraud/enforcement/former-nurse-aide-indicted-in-death-of-clarksville-patient-arrested-in-geo
```

- c. Test Your Code

```
result_2022=scrape_enforcement_custom_safe(start_date_str="01/01/2022",
 ↪  run_scraper=False)
```

```
Scraper indicator is OFF. Skipping scrape.
```

```
# Save the CSV
#result_2022.to_csv("/Users/nandinikrishnan/Documents/GitHub/Data_vis/ps4-nandinikrishnan1/da
 ↪  index=False)
```

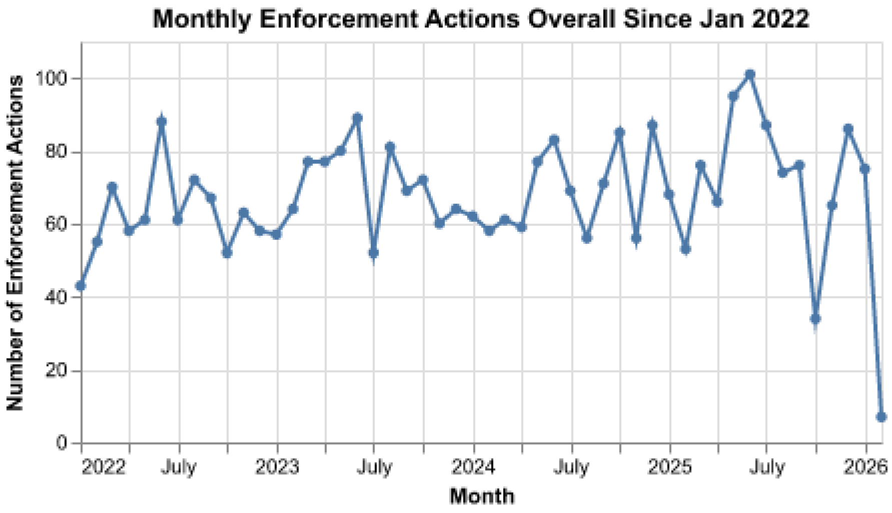
Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time

```
import altair as alt

# Load the CSV from Step 2
df =
 ↪  pd.read_csv("/Users/nandinikrishnan/Documents/GitHub/Data_vis/ps4-nandinikrishnan1/data_

df['date'] = pd.to_datetime(df['date'])

num_actions = len(df)
print(f"Number of enforcement actions: {num_actions}")

# Earliest enforcement action
earliest = df.sort_values("date").iloc[0]  # earliest date is first row after
 ↪  sorting
print("Earliest enforcement action:")
print(f"Date: {earliest['date'].date()}")
print(f"Title: {earliest['title']}")
print(f"Category: {earliest['category']}")
print(f"Link: {earliest['link']}")

# Convert date column to datetime
df['date'] = pd.to_datetime(df['date'])

# Create a 'year-month' column for aggregation
df['year_month'] = df['date'].dt.to_period('M').astype(str)

monthly_counts =
 ↪  df.groupby('year_month').size().reset_index(name='num_actions')
```

```
line_chart_overall = alt.Chart(monthly_counts).mark_line(point=True).encode(
    x=alt.X('year_month:T', title='Month'),
    y=alt.Y('num_actions:Q', title='Number of Enforcement Actions'),
    tooltip=['year_month', 'num_actions']
).properties(
    title='Monthly Enforcement Actions Overall Since Jan 2022',
    width=400,
    height=200
)

line_chart_overall
```

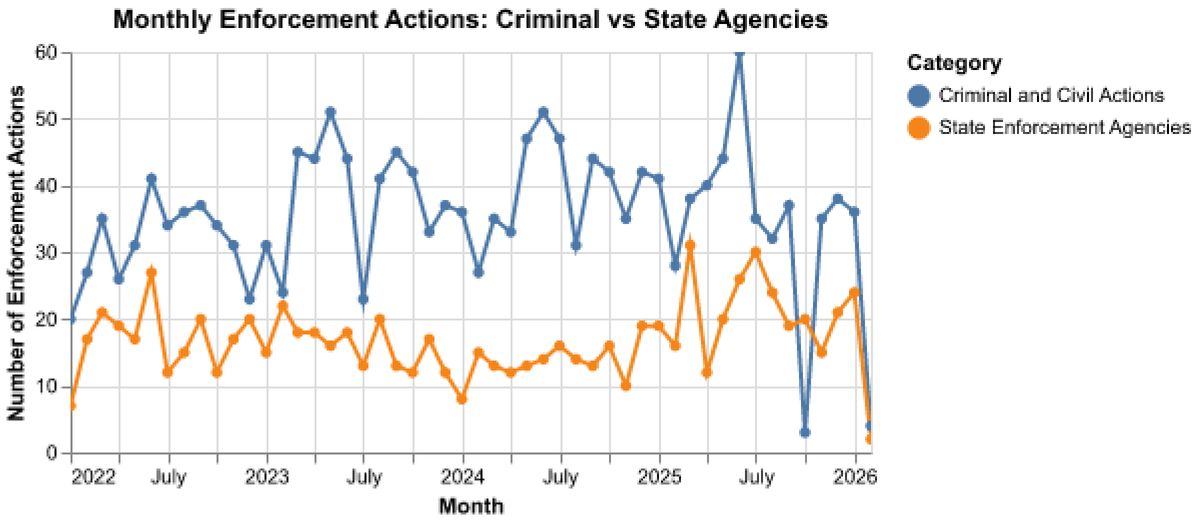
```
Number of enforcement actions: 3377
Earliest enforcement action:
Date: 2022-01-04
Title: Integrated Pain Management Medical Group Agreed to Pay $10,000 for
Allegedly Violating the Civil Monetary Penalties Law by Employing Excluded
Individuals
Category: Fraud Self-Disclosures
Link:
/fraud/enforcement/integrated-pain-management-medical-group-agreed-to-pay-10000-for-allegedly
```



2. Plot the number of enforcement actions categorized:

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

```
# Filter only the two categories
df_filtered = df[df['category'].isin(['Criminal and Civil Actions', 'State
 ↪  Enforcement Agencies'])].copy()

# Aggregate monthly counts
monthly_by_category = (
    df_filtered
    .groupby(['year_month', 'category'])
    .size()
    .reset_index(name='num_actions')
)

# Plot line chart
line_chart_category =
 ↪  alt.Chart(monthly_by_category).mark_line(point=True).encode(
    x=alt.X('year_month:T', title='Month'),
    y=alt.Y('num_actions:Q', title='Number of Enforcement Actions'),
    color=alt.Color('category:N', title='Category'),  # color according to
 ↪  category
    tooltip=['year_month', 'category', 'num_actions']
).properties(
    title='Monthly Enforcement Actions: Criminal vs State Agencies',
    width=400,
    height=200
).interactive()

line_chart_category
```



- based on five topics

```
# Filter only the two categories
df_filtered = df[df['category'].isin(['Criminal and Civil Actions'])].copy()

print(len(df_filtered))
```

```
1776
```

```
import pandas as pd

# df_filtered = your dataset where category == "Criminal and Civil Actions"

# Define keywords for each topic
topic_keywords = {
    'Health Care Fraud': ['health', 'medicaid', 'medicare', 'care',
     ↪  'hospital', 'clinic'],
    'Financial Fraud': ['bank', 'financial', 'loan', 'credit', 'investment',
     ↪  'mortgage', 'tax'],
    'Drug Enforcement': ['drug', 'opioid', 'controlled substance',
     ↪  'narcotic', 'pharmacy'],
    'Bribery/Corruption': ['bribery', 'corruption', 'kickback', 'payoff',
     ↪  'fraudulent payment']
}

# Function to assign topic with Health Care override and proper multiple flag
def assign_topic_health_override(title):
```

```python
    title_lower = title.lower()
    matched_topics = []

    for topic, keywords in topic_keywords.items():
        if any(word in title_lower for word in keywords):
            matched_topics.append(topic)

    # Initialize flags
    flag_other = False
    flag_multiple = False

    if len(matched_topics) == 0:
        # No match
        topic = 'Other'
        flag_other = True
    elif len(matched_topics) == 1:
        # Single match
        topic = matched_topics[0]
    else:
        # Multiple matches
        # Special override: Health Care + Bribery/Corruption
        if 'Health Care Fraud' in matched_topics and 'Bribery/Corruption' in matched_topics:
            topic = 'Health Care Fraud'
            flag_multiple = False  # do NOT flag in this special case
        else:
            topic = 'Other'
            flag_multiple = True  # flag all other multiple matches

    return topic, flag_other, flag_multiple

# Apply to DataFrame
df_filtered[['topic', 'flag_other', 'flag_multiple']] = df_filtered['title'].apply(
    lambda t: pd.Series(assign_topic_health_override(t))
)

# Quick checks
print("Flagged for no match (Other):")
print(df_filtered[df_filtered['flag_other']][['title', 'topic']])

print("\nFlagged for multiple matches (excluding Health Care override):")
print(df_filtered[df_filtered['flag_multiple']][['title', 'topic']])
print("\nTopic counts:")
print(df_filtered['topic'].value_counts())
```

```
Flagged for no match (Other):
                                                  title  topic
0     Houston Transplant Doctor Indicted For Making ...  Other
13    Scranton Heart Institute Agrees To Pay $48,709...  Other
33    Chipley-Based Physical Therapy Practice Pays O...  Other
46    Honduran Man and Kansas Woman Indicted for Fra...  Other
58    Kaiser Permanente Affiliates Pay $556 Million ...  Other
...                                                 ...    ...
3362  Diabetic Shoe Company Agrees to Pay $5.5 Milli...  Other
3365  Bowie, Maryland Nurse Practitioner Pleads Guil...  Other
3368  United States Attorney Announces Flint, MI, Ma...  Other
3372  North Carolina Physician Indicted for Adultera...  Other
3374  Central Medical Systems, LLC, Alan Trent Harle...  Other

[455 rows x 2 columns]

Flagged for multiple matches (excluding Health Care override):
                                                  title  topic
78    New Jersey Doctor Charged in 58-Count Indictme...  Other
119   Medical Biller Charged with Diverting Opioids ...  Other
164   Nurse Practitioner Sentenced to 18 Months and ...  Other
168   Southern Colorado Hospital and Doctors Agree t...  Other
171   New Jersey Pharmacy Owner Pleads Guilty to Hea...  Other
...                                                 ...    ...
3324  West L.A. Compounding Pharmacy Owner Sentenced...  Other
3326  Murrysville Doctor Sentenced for Illegal Drug ...  Other
3339  U.S. Attorney’s Office Files Suit Against Phil...  Other
3346  Former Tennessee Clinic Owner Sentenced for Op...  Other
3361  NC Pharmacy Operator Pleads Guilty to Conspira...  Other

[113 rows x 2 columns]

Topic counts:
topic
Health Care Fraud     932
Other                 568
Drug Enforcement      165
Bribery/Corruption     97
Financial Fraud        14
```

```
Name: count, dtype: int64
```

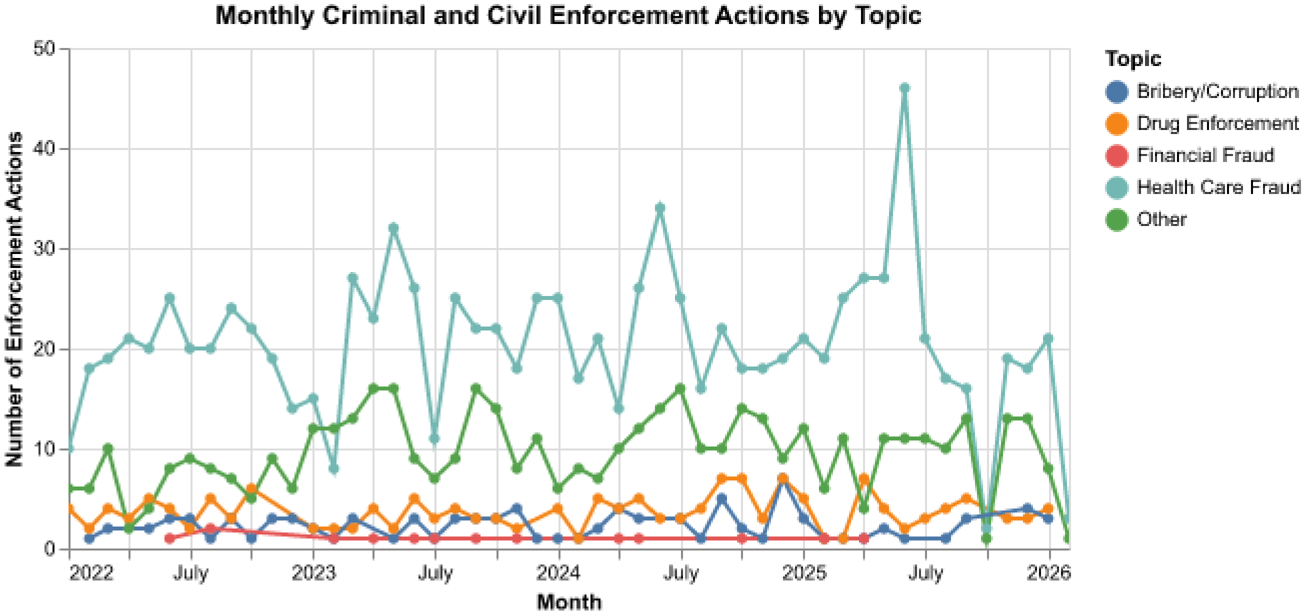
```
# First, keep the original matched topics for reference
def get_matched_topics(title):
    title_lower = title.lower()
    matched = []
    for topic, keywords in topic_keywords.items():
        if any(word in title_lower for word in keywords):
            matched.append(topic)
    return matched

df_filtered['matched_topics'] =
 ↪  df_filtered['title'].apply(get_matched_topics)

# Now assign topic and flags based on matched topics
def assign_topic_final(matched):
    flag_other = False
    flag_multiple = False

    if len(matched) == 0:
        topic = 'Other'
        flag_other = True
    elif len(matched) == 1:
        topic = matched[0]
    else:
        # Multiple matches
        if 'Health Care Fraud' in matched:
            topic = 'Health Care Fraud'
            flag_multiple = False  # resolved by override
        else:
            topic = 'Other'
            flag_multiple = True

    return pd.Series([topic, flag_other, flag_multiple])

# Apply final assignment
df_filtered[['topic', 'flag_other', 'flag_multiple']] =
 ↪  df_filtered['matched_topics'].apply(assign_topic_final)

# Optional: drop the temporary column if no longer needed
df_filtered.drop(columns=['matched_topics'], inplace=True)

# Quick check
```

```
print("Flagged for no match (Other):")
print(df_filtered[df_filtered['flag_other']][['title', 'topic']])

print("\nFlagged for multiple matches (excluding Health Care override):")
print(df_filtered[df_filtered['flag_multiple']][['title', 'topic']])

print("\nTopic counts:")
print(df_filtered['topic'].value_counts())
```

```
Flagged for no match (Other):
                                                  title  topic
0     Houston Transplant Doctor Indicted For Making ...  Other
13    Scranton Heart Institute Agrees To Pay $48,709...  Other
33    Chipley-Based Physical Therapy Practice Pays O...  Other
46    Honduran Man and Kansas Woman Indicted for Fra...  Other
58    Kaiser Permanente Affiliates Pay $556 Million ...  Other
...                                                 ...    ...
3362  Diabetic Shoe Company Agrees to Pay $5.5 Milli...  Other
3365  Bowie, Maryland Nurse Practitioner Pleads Guil...  Other
3368  United States Attorney Announces Flint, MI, Ma...  Other
3372  North Carolina Physician Indicted for Adultera...  Other
3374  Central Medical Systems, LLC, Alan Trent Harle...  Other

[455 rows x 2 columns]

Flagged for multiple matches (excluding Health Care override):
                                                  title  topic
453   Former Iowa Nurse Sentenced to Federal Prison ...  Other
704   U.S. Attorney Announces $202 Million Settlemen...  Other
1071  Pharmaceutical Company QOL Medical and CEO Agr...  Other
1073  QOL Medical and Its CEO Agree To Pay $47 Milli...  Other
1161  Drug Maker Teva Pharmaceuticals Agrees To Pay ...  Other
1206  Former Pharmacy Chief Financial Officer and Fo...  Other
1471  Pharmacy Owner Sentenced For Paying Illegal Ki...  Other
1692  California Pharmaceutical Company to Pay $750,...  Other
1791  Pharmaceutical Company Ultragenyx Agrees to Pa...  Other
2145  Pharmacy Owner Convicted Of Payment Of Illegal...  Other
2183  Dothan Man Sentenced For Federal Tax Evasion, ...  Other
2353  Former Creve Coeur Pharmacy Owner Admits Payin...  Other
2426  Texas Laboratory Agrees To Pay $5.9 Million To...  Other
2527  Dothan Man Pleads Guilty to Federal Tax Evasio...  Other
2560  Hudson Physician Sentenced To Prison And Order...  Other
2684  Opioid Abuse Treatment Facility To Pay $3.15 M...  Other
```

```
2710  Doctor Sentenced For Accepting Illegal Kickbac...  Other
2749  Hudson Physician And Ohio Pharmaceutical Rep P...  Other
2774  Doctor Sentenced for Accepting Illegal Kickbac...  Other
2933  New Jersey Pharmacy Admits Illegal Distributio...  Other
3215  Co-Owner of Defunct Medical Testing Lab Convic...  Other
3305  Doctor Pleads Guilty to Accepting Illegal Kick...  Other

Topic counts:
topic
Health Care Fraud     1023
Other                  477
Drug Enforcement       165
Bribery/Corruption      97
Financial Fraud         14
Name: count, dtype: int64
```

```
import altair as alt
import pandas as pd

# Ensure your DataFrame has a datetime column
df_filtered['date'] = pd.to_datetime(df_filtered['date'])

# Create a year-month column for aggregation
df_filtered['year_month'] =
 ↪  df_filtered['date'].dt.to_period('M').dt.to_timestamp()

# Aggregate counts per month per topic
monthly_counts = (
    df_filtered
    .groupby(['year_month', 'topic'])
    .size()
    .reset_index(name='num_actions')
)

# Altair line chart
line_chart = alt.Chart(monthly_counts).mark_line(point=True).encode(
    x=alt.X('year_month:T', title='Month'),
    y=alt.Y('num_actions:Q', title='Number of Enforcement Actions'),
    color=alt.Color('topic:N', title='Topic'),  # color by topic
    tooltip=['year_month', 'topic', 'num_actions']
).properties(
    width=500,
    height=250,
    title='Monthly Criminal and Civil Enforcement Actions by Topic'
).interactive()

line_chart
```


Monthly Criminal and Civil Enforcement Actions by Topic
Number of Enforcement Actions
July
July
July
July
Month
Topic
Bribery/Corruption
Drug Enforcement
Financial Fraud
Health Care Fraud
Other