PS4

Ruoyan Cai

2026-02-04

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: RC

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

```
import requests
from bs4 import BeautifulSoup

url = 'https://oig.hhs.gov/fraud/enforcement/'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'lxml')
```

```
cards = soup.find_all('header', class_='usa-card__header')
print(len(cards))
```

```
20
```

```
data_rows = []

for card in cards:
    title = card.find('a').text
    date = card.find('span', class_='text-base-dark
    ↪ padding-right-105').text

    category_items = card.find_all('li')
    categories = [cat.text.strip() for cat in category_items]
    category = ', '.join(categories)

    link = card.find('a').get('href')
    data_rows.append([title, date, category, link])

data_rows = pd.DataFrame(data_rows)
```

```
columns = ['Title', 'Date', 'Category', 'Link']
data_rows.columns = columns
display(data_rows.head())
```

	Title	Date	Category	L
0	Brooklyn Banker Pleads Guilty to Laundering Pr...	February 3, 2026	COVID-19	/
1	Delafield Man Sentenced to 18 Months' Imprison...	February 3, 2026	Criminal and Civil Actions	/
2	Former NFL Player Convicted for $197M Medicare...	February 3, 2026	Criminal and Civil Actions	/
3	AG's Office Secures Indictments Against Peabod...	February 2, 2026	State Enforcement Agencies	/
4	Florida Man Pleads Guilty to Conspiracy to Vio...	January 30, 2026	Criminal and Civil Actions	/

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code function: scrape_enforcement_actions(start_month, start_year)

input start month and year output dataframe that contains all of the enforcement actions acquired

steps: 1. check if year >= 2013, otherwise print a statement reminding the user to restrict to year >= 2013
2. create a empty list to store the data
3. set the page to 1
4. When current date <= today:
a. Construct the URL (including the page parameter)
b. Send the request and retrieve the webpage
c. Parse the webpage using BeautifulSoup
d. Extract all enforcement actions from this page
e. If there is no data on this page, it means the end has been reached, so break the loop.
f. Increment the page number by 1
g. Wait for 1 second
5. transform the data into dataframe
6. save as a CSV document

- b. Create Dynamic Scraper

```
import time
import pandas as pd
```

```
def scrape_enforcement_actions(start_month, start_year):
  start_time = f"{start_year}-{start_month:02d}-01"
  if start_year < 2013:
        print("The start year must be 2013 or later.")
        return None

  data = []
  page = 1

  while True:

    if page == 1:
      url = 'https://oig.hhs.gov/fraud/enforcement/'
    else:
      url = f'https://oig.hhs.gov/fraud/enforcement/?page={page}'

    print(f"Currently scraping page {page}, {url}.")

    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'lxml')
    cards = soup.find_all('header', class_='usa-card__header')

    if not cards:
      print("Stop scraping.")
      break

    for card in cards:
      title = card.find('a').text
      date = card.find('span', class_='text-base-dark
↪  padding-right-105').text.strip()
      date_obj = pd.to_datetime(date)
      date_formatted = date_obj.strftime("%Y-%m-%d")

      if date_formatted < start_time:
        print(f'The date {date} is earlier than the start time
        ↪  {start_time}, stop crawling.')
        return pd.DataFrame(data, columns=['Title', 'Date', 'Category',
        ↪  'Link'])

      category_items = card.find_all('li')
      categories = [cat.text.strip() for cat in category_items]
      category = ', '.join(categories)
```

```
      link = card.find('a').get('href')
      data.append([title, date, category, link])

    page += 1

    time.sleep(1)

  data_df = pd.DataFrame(data)
  data_df.columns = ['Title', 'Date', 'Category', 'Link']
  return data_df
```

```
RUN_SCRAPER_2024 = False

if RUN_SCRAPER_2024:
    df_2024 = scrape_enforcement_actions(start_month=1, start_year=2024)
    df_2024.to_csv('enforcement_actions_2024_01.csv', index=False)
    print(f'There are {len(df_2024)} enforcement actions from Jan. 2024
     ↪  to today.')
    display(df_2024.loc[df_2024['Date'].idxmin()])
else:
    None
```

There are 1769 enforcement actions from Jan. 2024 to today.

The earliest action: Shelby County Man Indicted for Offenses Relating to Child Pornography, Encouraging Illegal Entry into the United States, and Making False Statements to a Federal Agency.
date: April 1, 2024.
category: Criminal and Civil Actions.
link: /fraud/enforcement/shelby-county-man-indicted-for-offenses-relating-to-child-pornography-encouraging-illegal-entry-into-the-united-states-and-making-false-statements-to-a-federal-agency/

- c. Test Your Code

```
RUN_SCRAPER_2022 = False

if RUN_SCRAPER_2022:
    df_2022 = scrape_enforcement_actions(start_month=1, start_year=2022)
    df_2022.to_csv('enforcement_actions_2022_01.csv', index=False)
```

```
    print(f'There are {len(df_2022)} enforcement actions from Jan. 2022
     ↪  to today.')
    display(df_2022.loc[df_2022['Date'].idxmin()])
else:
    None
```

There are 3359 enforcement actions from Jan. 2022 to today.

The earliest enforcenment action: Attorney General Moody Announces Arrest of Medicaid Provider for More Than $77,000 in Medicaid Fraud
date: April 1, 2022
category: State Enforcement Agencies
link: /fraud/enforcement/attorney-general-moody-announces-arrest-of-medicaid-provider-for-more-than-77000-in-medicaid-fraud/

Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time

```
file_path =
 ↪  'D:\\UChi\\26winter\\Python\\ps4-ruoyancai279\\enforcement_actions_2022_01.csv'
df = pd.read_csv(file_path)

df['Year_month'] = pd.to_datetime(df['Date']).dt.strftime("%Y-%m")
df.head()
```

	Title	Date	Category	L
0	Brooklyn Banker Pleads Guilty to Laundering Pr...	February 3, 2026	COVID-19	/
1	Delafield Man Sentenced to 18 Months' Imprison...	February 3, 2026	Criminal and Civil Actions	/
2	Former NFL Player Convicted for $197M Medicare...	February 3, 2026	Criminal and Civil Actions	/
3	AG's Office Secures Indictments Against Peabod...	February 2, 2026	State Enforcement Agencies	/
4	Florida Man Pleads Guilty to Conspiracy to Vio...	January 30, 2026	Criminal and Civil Actions	/

```
df_count = df.groupby("Year_month").size().reset_index(
  name="Count"
)
df_count
```

	Year_month	Count
0	2022-01	43
1	2022-02	55
2	2022-03	70
3	2022-04	58
4	2022-05	61
5	2022-06	88
6	2022-07	61
7	2022-08	72
8	2022-09	67
9	2022-10	52
10	2022-11	63
11	2022-12	58
12	2023-01	57
13	2023-02	64
14	2023-03	77
15	2023-04	77
16	2023-05	80
17	2023-06	89
18	2023-07	52
19	2023-08	81
20	2023-09	69
21	2023-10	72
22	2023-11	60
23	2023-12	64
24	2024-01	62
25	2024-02	58
26	2024-03	61
27	2024-04	59
28	2024-05	77
29	2024-06	83
30	2024-07	69
31	2024-08	56
32	2024-09	71
33	2024-10	85
34	2024-11	56
35	2024-12	87
36	2025-01	68
37	2025-02	53
38	2025-03	76
39	2025-04	66
40	2025-05	95

	Year_month	Count
41	2025-06	101
42	2025-07	87
43	2025-08	74
44	2025-09	76
45	2025-10	34
46	2025-11	65
47	2025-12	85
48	2026-01	61
49	2026-02	4

```
line_chart = alt.Chart(df_count).mark_line(color='purple').encode(
  alt.X("Year_month").title("Time"),
  alt.Y("Count:Q").title("Number of Enforcement Actions")
)
line_chart
```

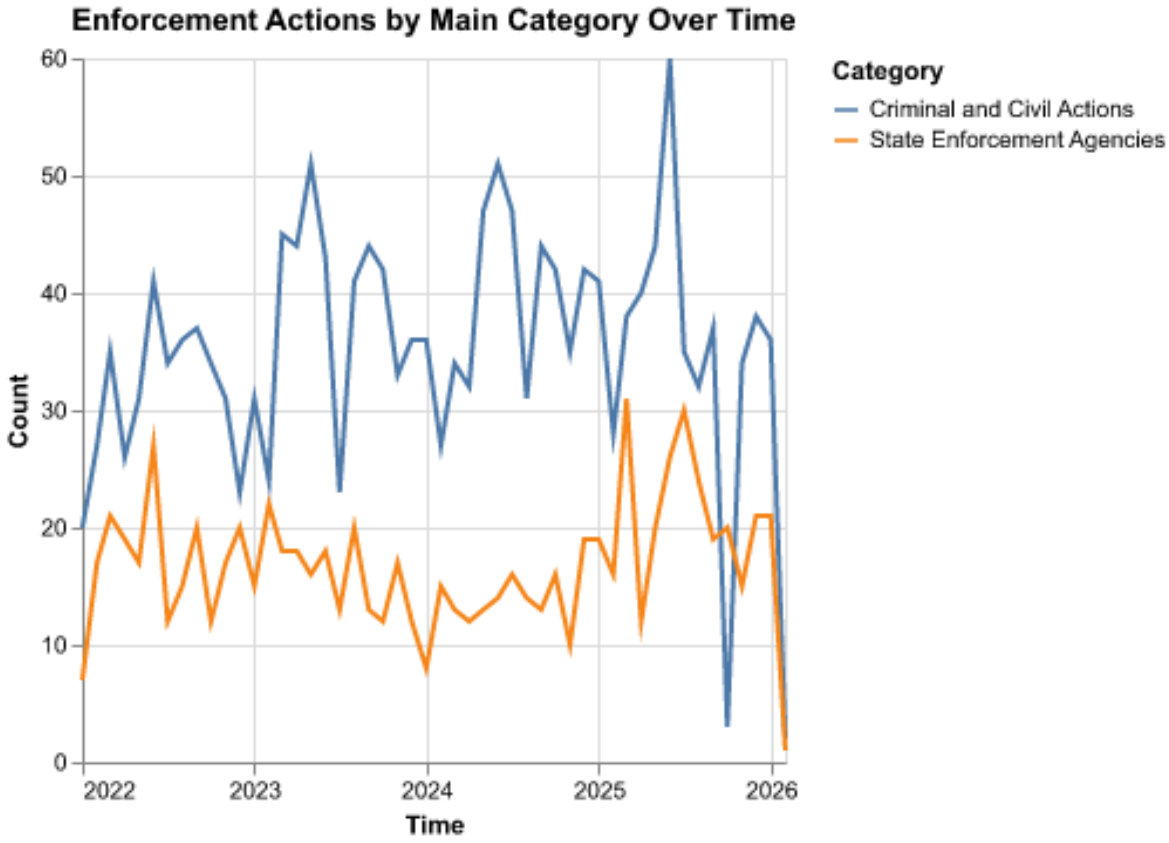
2. Plot the number of enforcement actions categorized:

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

```
df.head()
df_q1 = df.groupby(["Category", "Year_month"]).size().reset_index(
  name="Count"
)
df_q1
```

	Category	Year_month	Count
0	CIA Reportable Events	2022-03	1
1	CIA Reportable Events	2023-03	3
2	CIA Reportable Events	2023-04	1
3	CIA Reportable Events	2023-06	2
4	CIA Reportable Events	2023-12	1
...	...	...	...
290	State Enforcement Agencies	2025-12	21
291	State Enforcement Agencies	2026-01	21
292	State Enforcement Agencies	2026-02	1
293	Stipulated Penalties and Material Breaches	2025-04	3
294	Stipulated Penalties and Material Breaches	2025-08	1

```
line_1 = alt.Chart(df_q1).mark_line().encode(
  alt.X("Year_month:T").title("Time"),
  alt.Y("Count:Q"),
  color="Category:N"
).transform_filter(
  alt.FieldOneOfPredicate(field='Category', oneOf=["Criminal and Civil
 ↪  Actions", "State Enforcement Agencies"])
).properties(title="Enforcement Actions by Main Category Over Time")
line_1
```



- based on five topics

```
df_topics = df[df["Category"] == "Criminal and Civil Actions"]
df_topics
```

	Title	Date	Category
1	Delafield Man Sentenced to 18 Months' Imprison...	February 3, 2026	Criminal and Civil Actions
2	Former NFL Player Convicted for $197M Medicare...	February 3, 2026	Criminal and Civil Actions
4	Florida Man Pleads Guilty to Conspiracy to Vio...	January 30, 2026	Criminal and Civil Actions
5	Yadkinville Woman Sentenced in Connection with...	January 29, 2026	Criminal and Civil Actions
7	Repeat Health Care Fraud Offender Sentenced fo...	January 28, 2026	Criminal and Civil Actions
...	...	...	...
3348	UC San Diego Health Pays $2.98 Million to Reso...	January 11, 2022	Criminal and Civil Actions
3350	United States Attorney Announces Flint, MI, Ma...	January 6, 2022	Criminal and Civil Actions
3354	North Carolina Physician Indicted for Adultera...	January 5, 2022	Criminal and Civil Actions
3356	Central Medical Systems, LLC, Alan Trent Harle...	January 4, 2022	Criminal and Civil Actions

	Title	Date	Category
3357	Ohio home healthcare provider agrees to pay $5...	January 4, 2022	Criminal and Civil Actions

```
import numpy as np

def classify_topic(row):
    title = str(row['Title']).lower()
    category = str(row['Category']).lower()

    if any(w in title for w in ['medicare', 'medicaid', 'hospice',
     ↪  'physician', 'doctor', 'patient', 'nursing home', 'pharmacy',
     ↪  'ambulance', 'dental', 'medical', 'hospital', 'telehealth',
     ↪  'health care', 'healthcare']):
        return 'Health Care Fraud'

    # Financial Fraud
    elif any(w in title for w in ['bank', 'banker', 'finance',
     ↪  'financial', 'securities', 'tax', 'loan', 'wire fraud', 'money
     ↪  laundering', 'irs', 'grant', 'mortgage']):
        return 'Financial Fraud'

    # Drug Enforcement
    elif any(w in title for w in ['drug', 'opioid', 'pill', 'cocaine',
     ↪  'fentanyl', 'narcotic', 'controlled substance',
     ↪  'pharmaceutical', 'prescription']):
        return 'Drug Enforcement'

    # Bribery/Corruption
    elif any(w in title for w in ['kickback', 'bribe', 'corruption',
     ↪  'bribery', 'stark law', 'false claims act', 'councilman',
     ↪  'procurement']):
        return 'Bribery/Corruption'

    # other
    else:
        return 'Other'


df_topics['Topic'] = df_topics.apply(classify_topic, axis=1)

print(df_topics[['Title', 'Topic']].head())
```

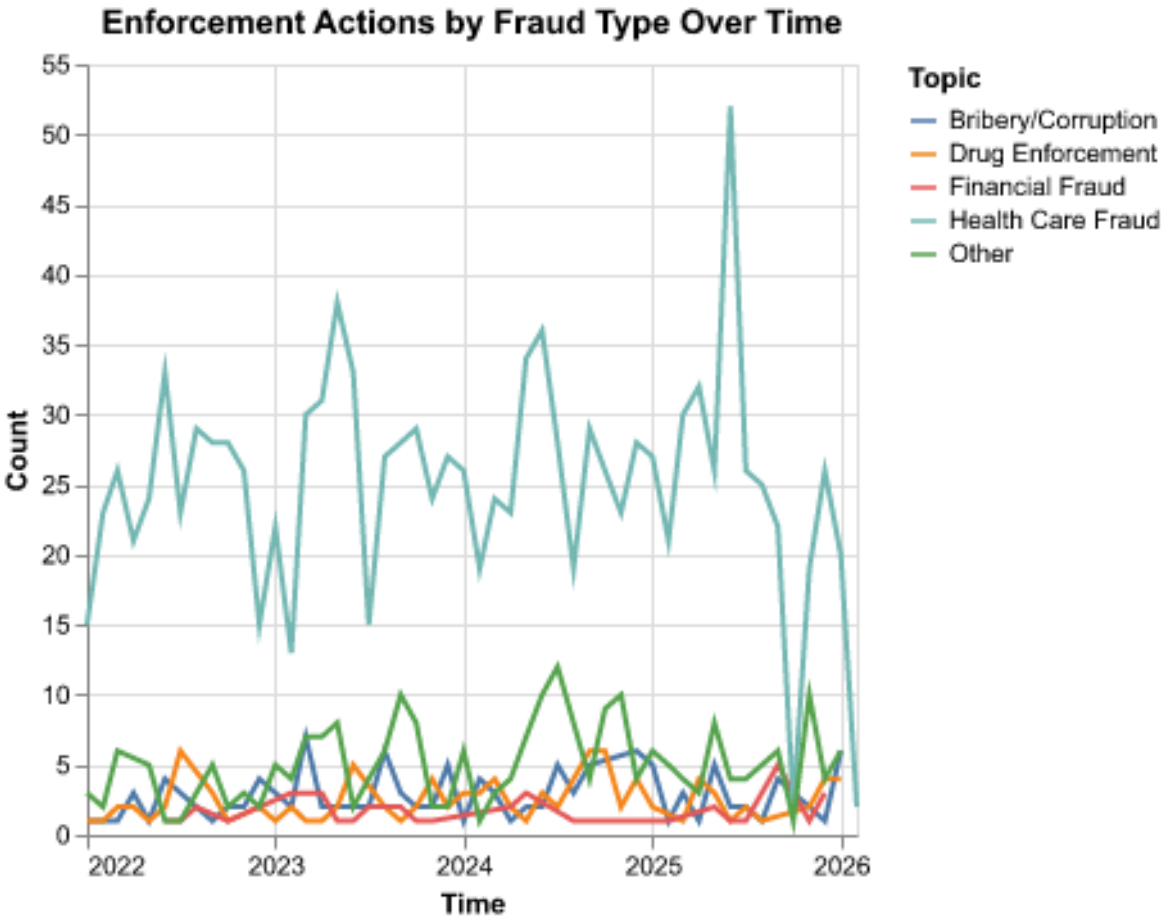
```
                                                Title                Topic
1  Delafield Man Sentenced to 18 Months' Imprison...   Health Care Fraud
2  Former NFL Player Convicted for $197M Medicare...   Health Care Fraud
4  Florida Man Pleads Guilty to Conspiracy to Vio...  Bribery/Corruption
5  Yadkinville Woman Sentenced in Connection with...   Health Care Fraud
7  Repeat Health Care Fraud Offender Sentenced fo...   Health Care Fraud
```

```
print(df_topics['Topic'].value_counts())
```

```
Topic
Health Care Fraud     1253
Other                  242
Bribery/Corruption     128
Drug Enforcement       104
Financial Fraud         41
Name: count, dtype: int64
```

```
df_chart = df_topics.groupby(["Topic",
 ↪  "Year_month"]).size().reset_index(name="Count")

line_2 = alt.Chart(df_chart).mark_line().encode(
  alt.X("Year_month:T").title("Time"),
  alt.Y("Count:Q"),
  color="Topic:N"
).properties(title="Enforcement Actions by Fraud Type Over Time")
line_2
```


Enforcement Actions by Fraud Type Over Time
Count
55
50
45
40
35
30
25
20
15
10
5
0
2022
2023
2024
2025
2026
Time
Topic
Bribery/Corruption
Drug Enforcement
Financial Fraud
Health Care Fraud
Other