Problem Set 4

Cassie Liu

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: CL

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

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

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

Step 1: Develop initial scraper and crawler

```python
import requests
from bs4 import BeautifulSoup
from datetime import datetime
```

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

li_cards = soup.find_all('li', class_ = 'usa-card')
li_cards[0:5]
len(li_cards)
```

```
20
```

```python
rows = []
for card in li_cards:
  a = card.find("a")
  title = a.get_text(strip = True) if a else None
  link = a['href'] if a else None
  full_link = 'https://oig.hhs.gov' + link if link else None

  date = card.find('span').get_text(strip = True)
  li_card_cat = card.select_one("ul.add-list-reset li")
  category = li_card_cat.get_text(strip = True) if li_card_cat else None

  rows.append({
    'title': title,
    'link': full_link,
    'date': date,
    'category': category
  })

df_card = pd.DataFrame(rows)
print(df_card)
```

```
                                                title  \
0   Houston Transplant Doctor Indicted For Making ...
1   MultiCare Health System to Pay Millions to Set...
2   Brooklyn Banker Pleads Guilty to Laundering Pr...
3   Delafield Man Sentenced to 18 Months’ Imprison...
4   Former NFL Player Convicted for $197M Medicare...
5   Attorney General Hanaway Obtains Medicaid Frau...
6   AG's Office Secures Indictments Against Peabod...
7   Florida Man Pleads Guilty to Conspiracy to Vio...
8   Forefront Living Hospice Agreed to Pay $1.9 Mi...
9   Attorney General Jeff Jackson Announces Health...
10  Yadkinville Woman Sentenced in Connection with...
11  Attorney General Labrador Announces Sentencing...
12  Attorney General Hanaway Obtains Medicaid Frau...
13  Holmes Regional Medical Center Agreed to Pay $...
14  Slidell Chiropractor Sentenced for Health Care...
15  Repeat Health Care Fraud Offender Sentenced fo...
16  Scranton Heart Institute Agrees To Pay $48,709...
17  Rheumatologist Agrees To Resolve False Claims ...
18  Attorney General James Uthmeier Announces Arre...
19  Cordell Memorial Hospital Agreed to Pay $40,00...

                                                 link              date  \
0   https://oig.hhs.gov/fraud/enforcement/houston-...  February 5, 2026
1   https://oig.hhs.gov/fraud/enforcement/multicar...  February 4, 2026
2   https://oig.hhs.gov/fraud/enforcement/brooklyn...  February 3, 2026
3   https://oig.hhs.gov/fraud/enforcement/delafiel...  February 3, 2026
4   https://oig.hhs.gov/fraud/enforcement/former-n...  February 3, 2026
5   https://oig.hhs.gov/fraud/enforcement/attorney...  February 3, 2026
6   https://oig.hhs.gov/fraud/enforcement/ags-offi...  February 2, 2026
7   https://oig.hhs.gov/fraud/enforcement/florida-...  January 30, 2026
8   https://oig.hhs.gov/fraud/enforcement/forefron...  January 30, 2026
9   https://oig.hhs.gov/fraud/enforcement/attorney...  January 30, 2026
10  https://oig.hhs.gov/fraud/enforcement/yadkinvi...  January 29, 2026
11  https://oig.hhs.gov/fraud/enforcement/attorney...  January 29, 2026
```

```
12  https://oig.hhs.gov/fraud/enforcement/attorney...  January 29, 2026
13  https://oig.hhs.gov/fraud/enforcement/holmes-r...  January 28, 2026
14  https://oig.hhs.gov/fraud/enforcement/slidell-...  January 28, 2026
15  https://oig.hhs.gov/fraud/enforcement/repeat-h...  January 28, 2026
16  https://oig.hhs.gov/fraud/enforcement/scranton...  January 28, 2026
17  https://oig.hhs.gov/fraud/enforcement/rheumato...  January 28, 2026
18  https://oig.hhs.gov/fraud/enforcement/attorney...  January 28, 2026
19  https://oig.hhs.gov/fraud/enforcement/cordell-...  January 27, 2026

                         category
0       Criminal and Civil Actions
1       Criminal and Civil Actions
2                         COVID-19
3       Criminal and Civil Actions
4       Criminal and Civil Actions
5       State Enforcement Agencies
6       State Enforcement Agencies
7       Criminal and Civil Actions
8           Fraud Self-Disclosures
9       State Enforcement Agencies
10      Criminal and Civil Actions
11      State Enforcement Agencies
12      State Enforcement Agencies
13  CMP and Affirmative Exclusions
14                        COVID-19
15      Criminal and Civil Actions
16      Criminal and Civil Actions
17      Criminal and Civil Actions
18      State Enforcement Agencies
19  CMP and Affirmative Exclusions
```

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code

1. If year_start < 2013: print 'Warning: year_start should >= 2013' return None

2. Set up a date_start, date_start = the first day of year_start-month_start

3. Pre setting for the function: rows = [] page = 1 keep_scraping = True

4. While keep_scraping = True: Building the url: url_n = ‘https://oig.hhs.gov/fraud/enforcement/’ + ‘?page=’ + page Request + using Beautiful Soup Find all li cards which class = usa_card if no cards are found: break for each card in cards: get the title, link, date, category convert date into ‘datetime’ d if d >= date_start: append({title, link, date, category}) to rows else: return reach_old_date = True After reaching all the cards: if reach_old_date = true: set keep_scraping = False else: page += 1 use sleep() to avoid blocks
5. Turn rows into dataframe
6. Save dataframe into csv files named as (enforcement_actions_year_month.csv)
7. Return dataframes

I used a loop while here instead of loop for because we don’t know how many pages are there. With a loop while, we can keep scraping until the threshold.

- b. Create Dynamic Scraper

```python
def scrape_enforcement_actions(year_start: int,
                               month_start: int,
                               run_scrape: bool = True) -> pd.DataFrame:

    # 0. indicator: read csv without crawling every time
    out_file = f"enforcement_actions_{year_start}_{month_start:02d}.csv"
    if not run_scrape:
        return pd.read_csv(out_file)

    # 1. start year check
    if year_start < 2013:
        print("Warning: year_start should >= 2013")
        return pd.DataFrame(columns=["title", "date", "category", "link"])

    # 2. defining the starting date
    date_start = datetime(year_start, month_start, 1)
    base_domain = "https://oig.hhs.gov"

    rows = []
    page = 1
    keep_scraping = True

    #3. starting the loop
    while keep_scraping:
        url = f"https://oig.hhs.gov/fraud/enforcement/?page={page}"
        response = requests.get(url)
        soup = BeautifulSoup(response.text, "lxml")

        cards = soup.find_all("li", class_="usa-card")
        if len(cards) == 0:
            break

        reach_old_date = False

        for card in cards:
            #title + link
            a = card.find("a")
            title = a.get_text(strip = True) if a else None
            href = a["href"] if a else None
            full_link = base_domain + href if href else None

            #date
            span = card.find("span")
            date = span.get_text(strip = True) if span else None

            #convert time into datetime
            try:
                d = datetime.strptime(date, "%B %d, %Y")
            except ValueError:
                continue

            #category
            li_cat = card.select_one("ul.add-list-reset li")
            category = li_cat.get_text(strip = True) if li_cat else None

            #filter those that date >= start_date
            if d >= date_start:
                rows.append({
                    "title": title,
                    "date": date,
                    "category": category,
                    "link": full_link
                })
            else:
                reach_old_date = True

        if reach_old_date:
            keep_scraping = False
        else:
            page += 1
            time.sleep(1)

    df = pd.DataFrame(rows)

    #save to csv
    df.to_csv(out_file, index = False)

    return df
```

```
df_2024 = scrape_enforcement_actions(2024, 1, run_scrape=True)
actions = len(df_2024)
print("Number of actions since 2024:", actions)

df_2024["date_dt"] = pd.to_datetime(df_2024["date"], format = "%B %d, %Y",
↪   errors = "coerce")
early1 =  df_2024.sort_values('date_dt', ascending = True).iloc[0]
print('Title for the earlist action:', early1['title'])
print('Link for the earlist action:', early1['link'])
print('Date for the earlist action:', early1['date'])
print('Category for the earlist action:', early1['category'])

print(df_2024.head(5))
```

```
Number of actions since 2024: 1787
Title for the earlist action: Former Nurse Aide Indicted In Death Of
Clarksville Patient Arrested In Georgia
Link for the earlist action:
https://oig.hhs.gov/fraud/enforcement/former-nurse-aide-indicted-in-death-of-clarksville-pati
Date for the earlist action: January 3, 2024
Category for the earlist action: State Enforcement Agencies
                                                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

                     category  \
0  Criminal and Civil Actions
1  Criminal and Civil Actions
2                    COVID-19
```

```
3  Criminal and Civil Actions
4  Criminal and Civil Actions

                                                link    date_dt
0  https://oig.hhs.gov/fraud/enforcement/houston-... 2026-02-05
1  https://oig.hhs.gov/fraud/enforcement/multicar... 2026-02-04
2  https://oig.hhs.gov/fraud/enforcement/brooklyn... 2026-02-03
3  https://oig.hhs.gov/fraud/enforcement/delafiel... 2026-02-03
4  https://oig.hhs.gov/fraud/enforcement/former-n... 2026-02-03
```

- c. Test Your Code

```
df_2022 = scrape_enforcement_actions(2022, 1, run_scrape=True)
actions22 = len(df_2022)
print("Number of actions since 2024:", actions22)

df_2022["date_dt"] = pd.to_datetime(df_2022["date"], format = "%B %d, %Y",
 ↪  errors = "coerce")
early1 =  df_2022.sort_values('date_dt', ascending = True).iloc[0]
print('Title for the earlist action:', early1['title'])
print('Link for the earlist action:', early1['link'])
print('Date for the earlist action:', early1['date'])
print('Category for the earlist action:', early1['category'])

print(df_2022.head(5))
```

```
Number of actions since 2024: 3377
Title for the earlist action: Integrated Pain Management Medical Group Agreed
to Pay $10,000 for Allegedly Violating the Civil Monetary Penalties Law by
Employing Excluded Individuals
Link for the earlist action:
https://oig.hhs.gov/fraud/enforcement/integrated-pain-management-medical-group-agreed-to-pay-
Date for the earlist action: January 4, 2022
Category for the earlist action: Fraud Self-Disclosures
                                               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

                     category  \
0  Criminal and Civil Actions
```

```
1  Criminal and Civil Actions
2                    COVID-19
3  Criminal and Civil Actions
4  Criminal and Civil Actions

                                                link    date_dt
0  https://oig.hhs.gov/fraud/enforcement/houston-... 2026-02-05
1  https://oig.hhs.gov/fraud/enforcement/multicar... 2026-02-04
2  https://oig.hhs.gov/fraud/enforcement/brooklyn... 2026-02-03
3  https://oig.hhs.gov/fraud/enforcement/delafiel... 2026-02-03
4  https://oig.hhs.gov/fraud/enforcement/former-n... 2026-02-03
```

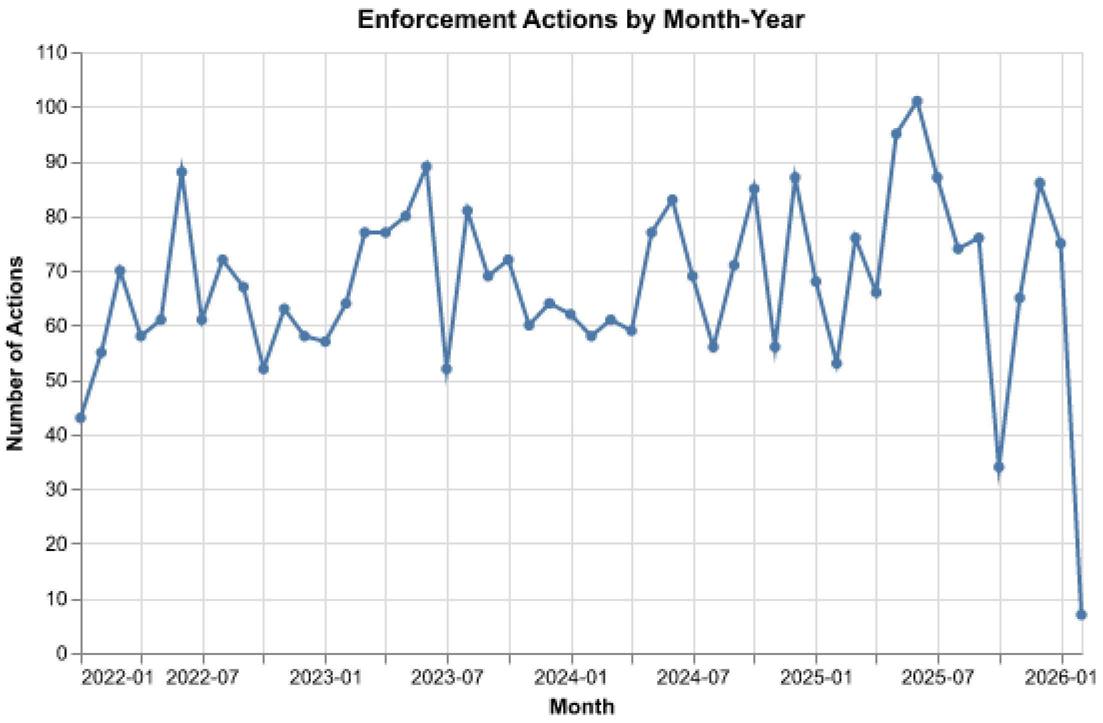
Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time

```python
df = pd.read_csv("/Users/ycliu/Documents/Uchi/Python2/github/ps4/"
"ps4-ycliu33/enforcement_actions_2022_01.csv")
df["date_dt"] = pd.to_datetime(df["date"], format = "%B %d, %Y", errors =
 ↪  "coerce")
df['year_month'] = df['date_dt'].dt.to_period('M').dt.to_timestamp()

month_count = (df.groupby('year_month').size().reset_index(name = 'mcount'))
```

```python
chart1 = alt.Chart(month_count).mark_line(point = True).encode(
  x = alt.X('year_month:T', title = 'Month', axis=alt.Axis(format='%Y-%m')),
  y = alt.Y('mcount:Q', title = 'Number of Actions'),
  tooltip = [alt.Tooltip('year_month:T', title = 'Month'),
  alt.Tooltip('mcount:Q', title = 'Number of Actions')]
).properties(
  title = 'Enforcement Actions by Month-Year',
  width = 500,
  height = 300
)
chart1
```



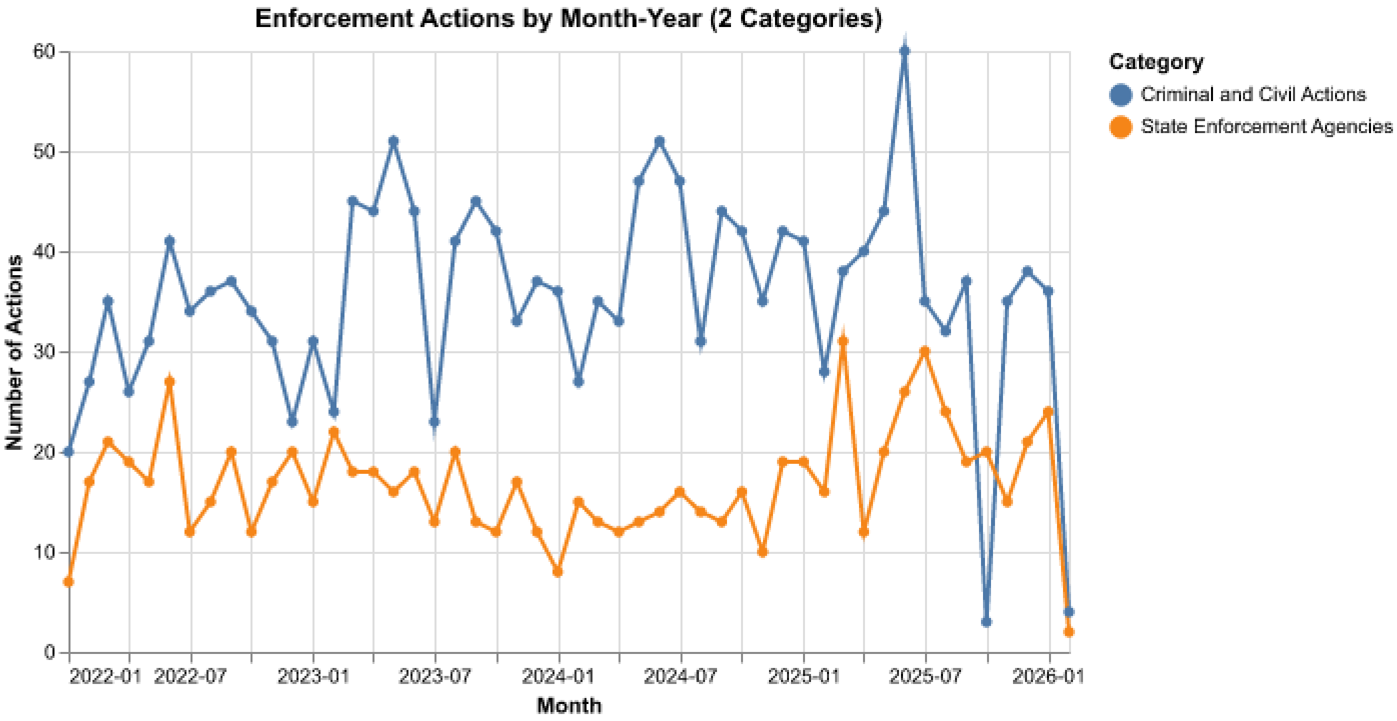
2. Plot the number of enforcement actions categorized:

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

```
df_2a = df[df['category'].isin(['Criminal and Civil Actions',
  'State Enforcement Agencies'])]
month_count_2a = (df_2a.groupby(['year_month', 'category'])
  .size().reset_index(name = 'mcount'))
```

```
chart2a = alt.Chart(month_count_2a).mark_line(point = True).encode(
  x = alt.X('year_month:T', title = 'Month', axis=alt.Axis(format='%Y-%m')),
  y = alt.Y('mcount:Q', title = 'Number of Actions'),
  color=alt.Color("category:N", title = "Category"),
  tooltip = [alt.Tooltip('year_month:T', title = 'Month'),
  alt.Tooltip('mcount:Q', title = 'Number of Actions')]
).properties(
  title = 'Enforcement Actions by Month-Year (2 Categories)',
  width = 500,
  height = 300
)
```

```
chart2a
```

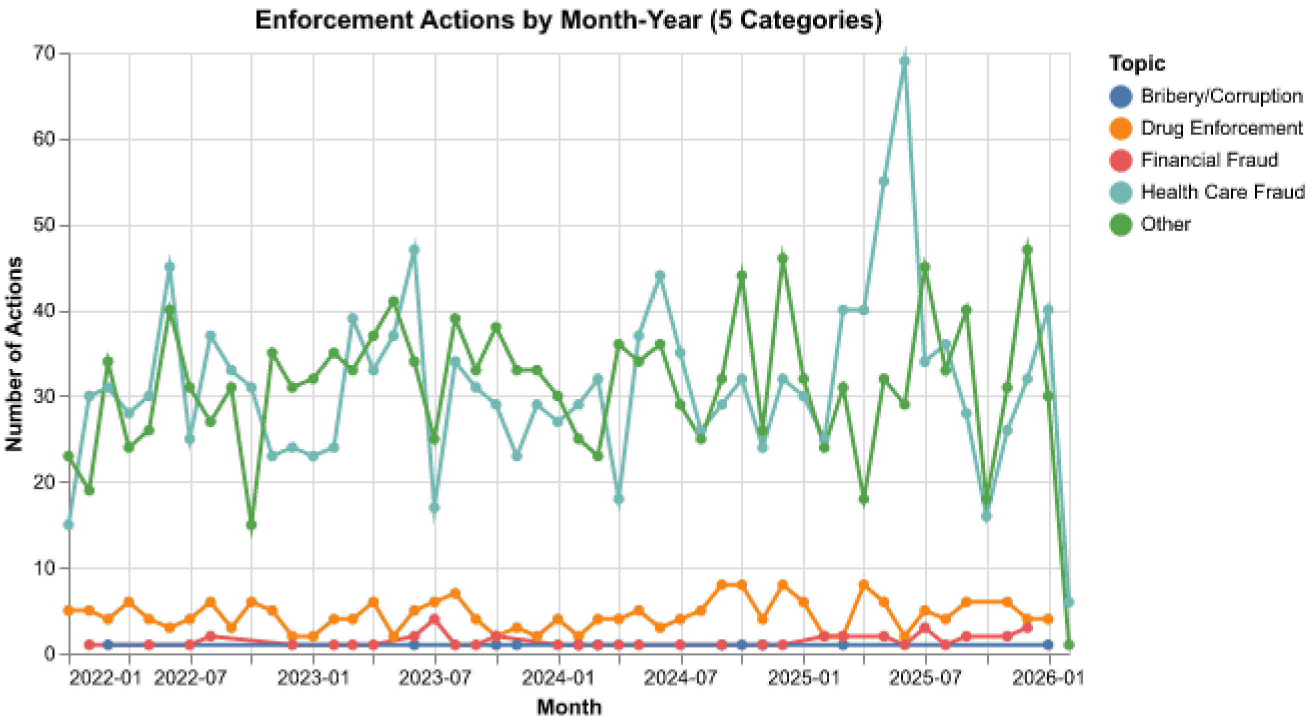


- based on five topics

```
def classify_topic(title):
    t = title.lower()
    if "health" in t or "medicare" in t or "medicaid" in t:
        return "Health Care Fraud"
    elif "bank" in t or "financial" in t or "money" in t:
        return "Financial Fraud"
    elif "drug" in t or "opioid" in t or "controlled substance" in t:
        return "Drug Enforcement"
    elif "bribe" in t or "corruption" in t:
        return "Bribery/Corruption"
    else:
        return "Other"

df['topic'] = df['title'].apply(classify_topic)
month_topic = (df.groupby(['year_month', 'topic']).size()
  .reset_index(name = 'mcount'))
```

```
chart2b = (alt.Chart(month_topic).mark_line(point=True).encode(
      x = alt.X("year_month:T", title = "Month",
 ↪  axis=alt.Axis(format='%Y-%m')),
      y = alt.Y("mcount:Q", title = "Number of Actions"),
      color = alt.Color("topic:N", title = "Topic"),
      tooltip = [
        alt.Tooltip('year_month:T', title = 'Month'),
        alt.Tooltip('topic:N', title = 'Topics'),
        alt.Tooltip('mcount:Q', title = 'Number of Actions')]
    )
    .properties(
        title = "Enforcement Actions by Month-Year (5 Categories)",
        width = 500,
        height = 300
    )
)
chart2b
```


Enforcement Actions by Month-Year (5 Categories)
Number of Actions
Month
Topic
● Bribery/Corruption
● Drug Enforcement
● Financial Fraud
● Health Care Fraud
● Other