PS 5

Yuting Meng, Yunzhou Guo

2024-11-07

Due 11/9 at 5:00PM Central. Worth 100 points + 10 points extra credit.

Submission Steps (10 pts)

1. This problem set is a paired problem set.
2. Play paper, scissors, rock to determine who goes first. Call that person *Partner 1*.
   - Partner 1 (name and cnet ID): Yuting Meng, yutingm
   - Partner 2 (name and cnet ID): Yunzhou Guo, guoy
3. Partner 1 will accept the `ps5` and then share the link it creates with their partner. You can only share it with one partner so you will not be able to change it after your partner has accepted.
4. "This submission is our work alone and complies with the 30538 integrity policy." Add your initials to indicate your agreement: YM, YG
5. "I have uploaded the names of anyone else other than my partner and I worked with on the problem set **here**" (1 point)
6. Late coins used this pset: 0 Late coins left after submission: 3
7. Knit your `ps5.qmd` to an PDF file to make `ps5.pdf`,
   - The PDF should not be more than 25 pages. Use `head()` and re-size figures when appropriate.
8. (Partner 1): push `ps5.qmd` and `ps5.pdf` to your github repo.
9. (Partner 1): submit `ps5.pdf` via Gradescope. Add your partner on Gradescope.
10. (Partner 1): tag your submission in Gradescope

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

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

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

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

Step 1: Develop initial scraper and crawler

1. Scraping (PARTNER 1)

```python
url = "https://oig.hhs.gov/fraud/enforcement/"

response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

titles = []
dates = []
categories = []
links = []

for action in soup.select('li.usa-card.card--list.pep-card--minimal'):
    title_tag = action.select_one('h2.usa-card__heading a')
    title = title_tag.get_text(strip=True)
    link = f"https://oig.hhs.gov{title_tag['href']}"

    date = action.select_one('span.text-base-dark').get_text(strip=True)
    category = action.select_one('li.usa-tag').get_text(strip=True)

    titles.append(title)
    dates.append(date)
    categories.append(category)
    links.append(link)
```

```
data = {
    "Title": titles,
    "Date": dates,
    "Category": categories,
    "Link": links
}
df = pd.DataFrame(data)

df
```

	Title	Date	Category
0	Former Arlington Resident Sentenced To Prison ...	November 7, 2024	Criminal and Civil Action
1	Paroled Felon Sentenced To Six Years For Fraud...	November 7, 2024	Criminal and Civil Action
2	Former Licensed Counselor Sentenced For Defrau...	November 6, 2024	Criminal and Civil Action
3	Macomb County Doctor And Pharmacist Agree To P...	November 4, 2024	Criminal and Civil Action
4	Rocky Hill Pharmacy And Its Owners Indicted Fo...	November 4, 2024	Criminal and Civil Action
5	North Texas Medical Center Pays $14.2 Million ...	November 4, 2024	Criminal and Civil Action
6	New England Doctor Pleads Guilty To Drug Distr...	November 4, 2024	Criminal and Civil Action
7	St. Louis County Woman Accused Of $3 Million H...	November 1, 2024	Criminal and Civil Action
8	Lab Owner And Marketing Company Owner Both Fou...	November 1, 2024	Criminal and Civil Action
9	Compound Ingredient Supplier Medisca Inc., To ...	November 1, 2024	Criminal and Civil Action
10	Columbus Doctor, His Clinic Convicted of $1.5 ...	October 31, 2024	State Enforcement Agenci
11	Quincy-Based Physician Group To Pay $650,000 T...	October 30, 2024	State Enforcement Agenci
12	Boise Health Care Company And Former Therapist...	October 29, 2024	Criminal and Civil Action
13	Drug Dealer Sentenced To Nine Years In Prison ...	October 29, 2024	Criminal and Civil Action
14	Former Sioux City Plastic Surgeon Agrees To Pa...	October 29, 2024	Criminal and Civil Action
15	Medical Billing Company Owner Pleads Guilty To...	October 29, 2024	Criminal and Civil Action
16	Doctor Sentenced For $54M Medicare Fraud Scheme	October 29, 2024	Criminal and Civil Action
17	AG's Office Reaches Settlement With Swampscott...	October 29, 2024	State Enforcement Agenci
18	Physician Charged In Scheme To Illegally Sell ...	October 25, 2024	Criminal and Civil Action
19	Washington Doctor Settles Allegations He Submi...	October 25, 2024	Criminal and Civil Action

2. Crawling (PARTNER 1)

```
titles = []
dates = []
categories = []
```

```python
links = []
agencies = []
for action in soup.select('li.usa-card.card--list.pep-card--minimal'):
    title_tag = action.select_one('h2.usa-card__heading a')
    title = title_tag.get_text(strip=True)
    link = f"https://oig.hhs.gov{title_tag['href']}"

    date = action.select_one('span.text-base-dark').get_text(strip=True)
    category = action.select_one('li.usa-tag').get_text(strip=True)

    titles.append(title)
    dates.append(date)
    categories.append(category)
    links.append(link)

    try:
        action_response = requests.get(link)
        action_soup = BeautifulSoup(action_response.text, 'html.parser')

        agency_name = "Not Found"
        for label in action_soup.find_all('span'):
            if "Agency:" in label.get_text():
                agency_name = label.find_next_sibling(text=True).strip() if label.find_next_sibling(text=True) else "Not Found"
                break
    except Exception as e:
        agency_name = "Not Found"

    agencies.append(agency_name)

min_length = min(len(titles), len(dates), len(categories), len(links), len(agencies))
data = {
    "Title": titles[:min_length],
    "Date": dates[:min_length],
    "Category": categories[:min_length],
    "Link": links[:min_length],
    "Agency": agencies[:min_length]
}

df = pd.DataFrame(data)
```

```
df
```

	Title	Date	Category
0	Former Arlington Resident Sentenced To Prison ...	November 7, 2024	Criminal and Civil Action
1	Paroled Felon Sentenced To Six Years For Fraud...	November 7, 2024	Criminal and Civil Action
2	Former Licensed Counselor Sentenced For Defrau...	November 6, 2024	Criminal and Civil Action
3	Macomb County Doctor And Pharmacist Agree To P...	November 4, 2024	Criminal and Civil Action
4	Rocky Hill Pharmacy And Its Owners Indicted Fo...	November 4, 2024	Criminal and Civil Action
5	North Texas Medical Center Pays $14.2 Million ...	November 4, 2024	Criminal and Civil Action
6	New England Doctor Pleads Guilty To Drug Distr...	November 4, 2024	Criminal and Civil Action
7	St. Louis County Woman Accused Of $3 Million H...	November 1, 2024	Criminal and Civil Action
8	Lab Owner And Marketing Company Owner Both Fou...	November 1, 2024	Criminal and Civil Action
9	Compound Ingredient Supplier Medisca Inc., To ...	November 1, 2024	Criminal and Civil Action
10	Columbus Doctor, His Clinic Convicted of $1.5 ...	October 31, 2024	State Enforcement Agenci
11	Quincy-Based Physician Group To Pay $650,000 T...	October 30, 2024	State Enforcement Agenci
12	Boise Health Care Company And Former Therapist...	October 29, 2024	Criminal and Civil Action
13	Drug Dealer Sentenced To Nine Years In Prison ...	October 29, 2024	Criminal and Civil Action
14	Former Sioux City Plastic Surgeon Agrees To Pa...	October 29, 2024	Criminal and Civil Action
15	Medical Billing Company Owner Pleads Guilty To...	October 29, 2024	Criminal and Civil Action
16	Doctor Sentenced For $54M Medicare Fraud Scheme	October 29, 2024	Criminal and Civil Action
17	AG's Office Reaches Settlement With Swampscott...	October 29, 2024	State Enforcement Agenci
18	Physician Charged In Scheme To Illegally Sell ...	October 25, 2024	Criminal and Civil Action
19	Washington Doctor Settles Allegations He Submi...	October 25, 2024	Criminal and Civil Action

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code (PARTNER 2)

1. Input Validation: Check if the year is greater than or equal to 2013. If the year is less than 2013, print a reminder to restrict the year to >= 2013.
2. URL Construction: Based on the input month and year, construct the starting URL for scraping (e.g., page 1, page 2, etc.). Loop through multiple pages to gather all the data.
3. Scraping and Storing Data: Scrape the enforcement actions from each page (titles, dates, categories, links, agencies). Store the scraped data in lists. After scraping all pages, save the data into a DataFrame.
4. Save to CSV: After scraping all enforcement actions, save the data to a .csv file named enforcement_actions_year_month.csv.

- b. Create Dynamic Scraper (PARTNER 2)

```
import aiohttp
import asyncio
from bs4 import BeautifulSoup
import pandas as pd
from datetime import datetime
import nest_asyncio

nest_asyncio.apply()

async def fetch(session, url):
    async with session.get(url) as response:
        return await response.text()

async def fetch_agency(session, link):
    """Fetches the agency name from the action detail page."""
    try:
        html = await fetch(session, link)
        soup = BeautifulSoup(html, 'html.parser')

        agency_name = "Not Found"
        for label in soup.find_all('span'):
            if "Agency:" in label.get_text():
                agency_name = label.find_next_sibling(text=True).strip() if
↪   label.find_next_sibling(text=True) else "Not Found"
                break
        return agency_name
    except Exception as e:
        print(f"Error fetching agency for {link}: {e}")
        return "Not Found"

async def scrape_page(session, page_number, start_date, titles, dates,
↪   categories, links, agencies):
    url = f"https://oig.hhs.gov/fraud/enforcement/?page={page_number}"
    print(f"Scraping page {page_number}...")

    html = await fetch(session, url)
    soup = BeautifulSoup(html, 'html.parser')
    actions = soup.select('li.usa-card.card--list.pep-card--minimal')

    if not actions:
        print(f"No actions found on page {page_number}.")
        return False

    page_reached_start_date = False

    for action in actions:
        title_tag = action.select_one('h2.usa-card__heading a')
        title = title_tag.get_text(strip=True)
        link = f"https://oig.hhs.gov{title_tag['href']}"

        date_str =
 ↪  action.select_one('span.text-base-dark').get_text(strip=True)
        action_date = datetime.strptime(date_str, "%B %d, %Y")

        if action_date < start_date:
            page_reached_start_date = True
            break  # Stop processing further actions on this page if before
             ↪  start_date

        category = action.select_one('li.usa-tag').get_text(strip=True)

        titles.append(title)
        dates.append(date_str)
        categories.append(category)
        links.append(link)

        agency_name = await fetch_agency(session, link)
        agencies.append(agency_name)

    return not page_reached_start_date

async def scrape_enforcement_actions(year, month, max_pages=480,
 ↪  batch_size=10):
    start_date = datetime(year, month, 1)
    titles, dates, categories, links, agencies = [], [], [], [], []

    async with aiohttp.ClientSession() as session:
        for start_page in range(1, max_pages + 1, batch_size):
            tasks = [
                scrape_page(session, page_number, start_date, titles, dates,
 ↪  categories, links, agencies)
                for page_number in range(start_page, min(start_page +
                 ↪  batch_size, max_pages + 1))
            ]

            results = await asyncio.gather(*tasks)
            if not all(results):
                print("Stopping scraping as reached entries before
                 ↪  start_date.")
                break

    data = {
        "Title": titles,
        "Date": dates,
        "Category": categories,
        "Link": links,
        "Agency": agencies
    }
    df = pd.DataFrame(data)
    csv_filename = f"enforcement_actions_{year}_{month}.csv"
    df.to_csv(csv_filename, index=False)

    print(f"Data saved to {csv_filename}")
    print(f"Total records: {len(df)}")
    print(f"Earliest date in data: {df['Date'].iloc[-1] if not df.empty else
     ↪  'No data'}")
    return df

year, month = 2023, 1
await scrape_enforcement_actions(year, month)
```

```
Scraping page 1...
Scraping page 2...
Scraping page 3...
Scraping page 4...
Scraping page 5...
Scraping page 6...
Scraping page 7...
Scraping page 8...
Scraping page 9...
Scraping page 10...
Scraping page 11...
Scraping page 12...
Scraping page 13...
Scraping page 14...
Scraping page 15...
Scraping page 16...
Scraping page 17...
Scraping page 18...
Scraping page 19...
Scraping page 20...
Scraping page 21...
Scraping page 22...
Scraping page 23...
Scraping page 24...
Scraping page 25...
Scraping page 26...
Scraping page 27...
Scraping page 28...
Scraping page 29...
Scraping page 30...
Scraping page 31...
Scraping page 32...
Scraping page 33...
Scraping page 34...
Scraping page 35...
Scraping page 36...
Scraping page 37...
Scraping page 38...
Scraping page 39...
Scraping page 40...
Scraping page 41...
Scraping page 42...
Scraping page 43...
Scraping page 44...
Scraping page 45...
Scraping page 46...
Scraping page 47...
Scraping page 48...
Scraping page 49...
Scraping page 50...
Scraping page 51...
Scraping page 52...
Scraping page 53...
Scraping page 54...
Scraping page 55...
Scraping page 56...
Scraping page 57...
Scraping page 58...
Scraping page 59...
Scraping page 60...
Scraping page 61...
Scraping page 62...
Scraping page 63...
Scraping page 64...
Scraping page 65...
Scraping page 66...
Scraping page 67...
Scraping page 68...
Scraping page 69...
Scraping page 70...
Scraping page 71...
Scraping page 72...
Scraping page 73...
Scraping page 74...
Scraping page 75...
Scraping page 76...
Scraping page 77...
Scraping page 78...
Scraping page 79...
Scraping page 80...
Stopping scraping as reached entries before start_date.
Data saved to enforcement_actions_2023_1.csv
Total records: 1510
Earliest date in data: January 19, 2023
```

	Title	Date	Category
0	Tennessee Business Owner Convicted of $35M Fra...	October 25, 2024	Criminal and Civil Actio
1	Buffalo Pharmacy Pays More Than $140,000 To Se...	August 23, 2024	Criminal and Civil Actio
2	Attorney General Griffin Announces Medicaid Fr...	October 25, 2024	State Enforcement Agen
3	Home Health Care Companies Owner Convicted Of ...	September 23, 2024	Criminal and Civil Actio
4	The Pathway Home Abuse Investigation – Victims...	September 27, 2024	Criminal and Civil Actio
...	...	...	...
1505	Twenty-Three Individuals Charged In $61.5 Mill...	February 7, 2023	Criminal and Civil Actio
1506	AG Campbell Secures $2.5 Million In Relief Fro...	February 16, 2023	State Enforcement Agen
1507	Former Louisiana Health Clinic CEO Sentenced T...	January 19, 2023	Criminal and Civil Actio
1508	Attorney General Ford Announces Sentencing of ...	February 16, 2023	State Enforcement Agen
1509	Man Pleads Guilty To Committing Multi-Million-...	January 19, 2023	Criminal and Civil Actio

There are 1510 records that I got. The earliest date in data was on Feb 7, 2023. Twenty-Three Individuals Charged In $61.5 Mill… February 7, 2023 Criminal and Civil Actions https://oig.hhs.gov/fraud/enforcement/twenty-t… U.S. Department of Justice.

- c. Test Partner's Code (PARTNER 1)

```
# Set the start date to January 2021
year, month = 2021, 1
await scrape_enforcement_actions(year, month, batch_size=50)
```

```
Scraping page 1...
Scraping page 2...
Scraping page 3...
Scraping page 4...
Scraping page 5...
Scraping page 6...
Scraping page 7...
Scraping page 8...
Scraping page 9...
Scraping page 10...
Scraping page 11...
Scraping page 12...
Scraping page 13...
Scraping page 14...
Scraping page 15...
Scraping page 16...
Scraping page 17...
Scraping page 18...
Scraping page 19...
Scraping page 20...
Scraping page 21...
Scraping page 22...
Scraping page 23...
Scraping page 24...
Scraping page 25...
Scraping page 26...
Scraping page 27...
Scraping page 28...
Scraping page 29...
Scraping page 30...
Scraping page 31...
Scraping page 32...
Scraping page 33...
Scraping page 34...
Scraping page 35...
Scraping page 36...
Scraping page 37...
Scraping page 38...
Scraping page 39...
Scraping page 40...
Scraping page 41...
Scraping page 42...
Scraping page 43...
Scraping page 44...
Scraping page 45...
Scraping page 46...
Scraping page 47...
Scraping page 48...
Scraping page 49...
Scraping page 50...
Scraping page 51...
Scraping page 52...
Scraping page 53...
Scraping page 54...
Scraping page 55...
Scraping page 56...
Scraping page 57...
Scraping page 58...
Scraping page 59...
Scraping page 60...
Scraping page 61...
Scraping page 62...
Scraping page 63...
Scraping page 64...
Scraping page 65...
Scraping page 66...
Scraping page 67...
Scraping page 68...
Scraping page 69...
Scraping page 70...
Scraping page 71...
Scraping page 72...
Scraping page 73...
Scraping page 74...
Scraping page 75...
Scraping page 76...
Scraping page 77...
Scraping page 78...
Scraping page 79...
Scraping page 80...
Scraping page 81...
Scraping page 82...
Scraping page 83...
Scraping page 84...
Scraping page 85...
Scraping page 86...
Scraping page 87...
Scraping page 88...
Scraping page 89...
Scraping page 90...
Scraping page 91...
Scraping page 92...
Scraping page 93...
Scraping page 94...
Scraping page 95...
Scraping page 96...
Scraping page 97...
Scraping page 98...
Scraping page 99...
Scraping page 100...
Scraping page 101...
Scraping page 102...
Scraping page 103...
Scraping page 104...
Scraping page 105...
Scraping page 106...
Scraping page 107...
Scraping page 108...
Scraping page 109...
Scraping page 110...
Scraping page 111...
Scraping page 112...
Scraping page 113...
Scraping page 114...
Scraping page 115...
Scraping page 116...
Scraping page 117...
Scraping page 118...
Scraping page 119...
Scraping page 120...
Scraping page 121...
Scraping page 122...
Scraping page 123...
Scraping page 124...
Scraping page 125...
Scraping page 126...
Scraping page 127...
Scraping page 128...
Scraping page 129...
Scraping page 130...
Scraping page 131...
Scraping page 132...
Scraping page 133...
Scraping page 134...
Scraping page 135...
Scraping page 136...
Scraping page 137...
Scraping page 138...
Scraping page 139...
Scraping page 140...
Scraping page 141...
Scraping page 142...
Scraping page 143...
Scraping page 144...
Scraping page 145...
Scraping page 146...
Scraping page 147...
Scraping page 148...
Scraping page 149...
Scraping page 150...
Stopping scraping as reached entries before start_date.
Data saved to enforcement_actions_2021_1.csv
Total records: 2998
Earliest date in data: January 20, 2021
```

	Title	Date	Category
0	Attorney General Paxton’s Medicaid Fraud Contr...	June 17, 2024	State Enforcement Agencie
1	The Pathway Home Abuse Investigation – Victims...	September 27, 2024	Criminal and Civil Actions
2	U.S. Attorney Announces Charges Against Five I...	September 13, 2023	COVID-19
3	Trans-Care Ambulance Agreed to Pay $239,000 fo...	November 16, 2023	Fraud Self-Disclosures
4	Opioid Manufacturer Endo Health Solutions Inc....	May 3, 2024	Criminal and Civil Actions

	Title	Date	Category
...	...	...	...
2993	Miami-Based CareCloud Health, Inc. Agrees to P...	April 30, 2021	Criminal and Civil Actions
2994	Pathology Practice Agrees to Pay $2.4 Million ...	December 7, 2021	Criminal and Civil Actions
2995	Two Defendants Plead Guilty to Multi-Million D...	January 20, 2021	Criminal and Civil Actions
2996	Attorney General Moody Announces Multimillion-...	January 20, 2021	State Enforcement Agenci
2997	Grace and Theophilus Egbujor Excluded for Mate...	January 20, 2021	Stipulated Penalties and M

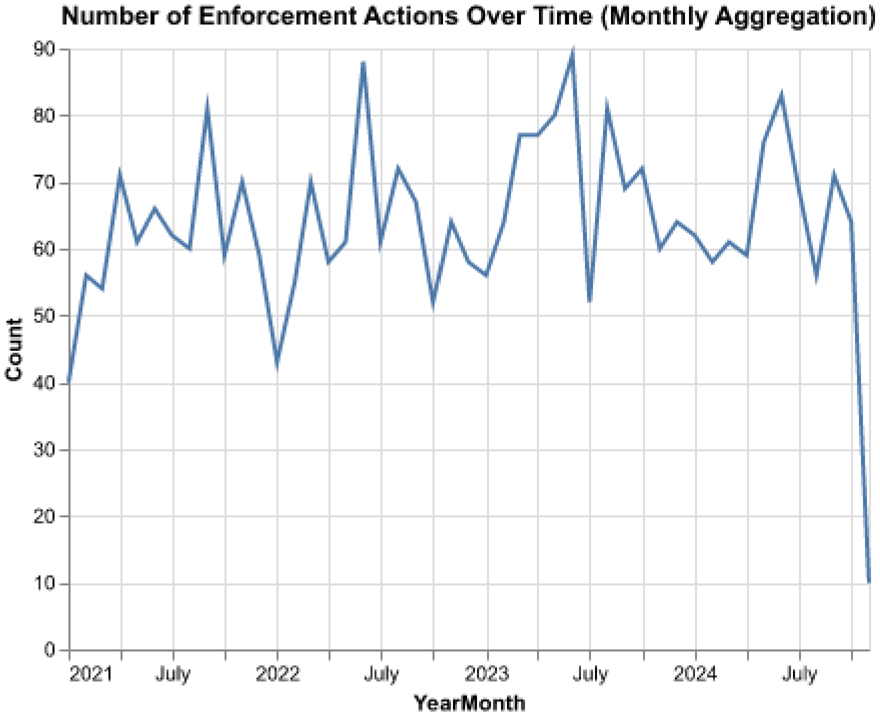
There are 2998 records that I got. The earliest date in data was on September 17, 2021. Gloucester County Man Charged with Fraud for R… September 17, 2021 Criminal and Civil Actions https://oig.hhs.gov/fraud/enforcement/gloucest…

Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time (PARTNER 2)

```
import pandas as pd
import altair as alt

df = pd.read_csv('enforcement_actions_2021_1.csv')
df['Date'] = pd.to_datetime(df['Date'])

# Aggregate the data by year and month to get monthly counts of enforcement
 ↪  actions
df['YearMonth'] = df['Date'].dt.to_period('M')
monthly_counts = df.groupby('YearMonth').size().reset_index(name='Count')
monthly_counts['YearMonth'] = monthly_counts['YearMonth'].dt.to_timestamp()

# Plotting the line chart
chart = alt.Chart(monthly_counts).mark_line().encode(
    x='YearMonth:T',
    y='Count:Q',
    tooltip=['YearMonth:T', 'Count:Q']
).properties(
    title="Number of Enforcement Actions Over Time (Monthly Aggregation)",
    width=400,
    height=300
)
```

```
chart
```


Number of Enforcement Actions Over Time (Monthly Aggregation)
Count
90
80
70
60
50
40
30
20
10
0
2021
July
2022
July
2023
July
2024
July
YearMonth

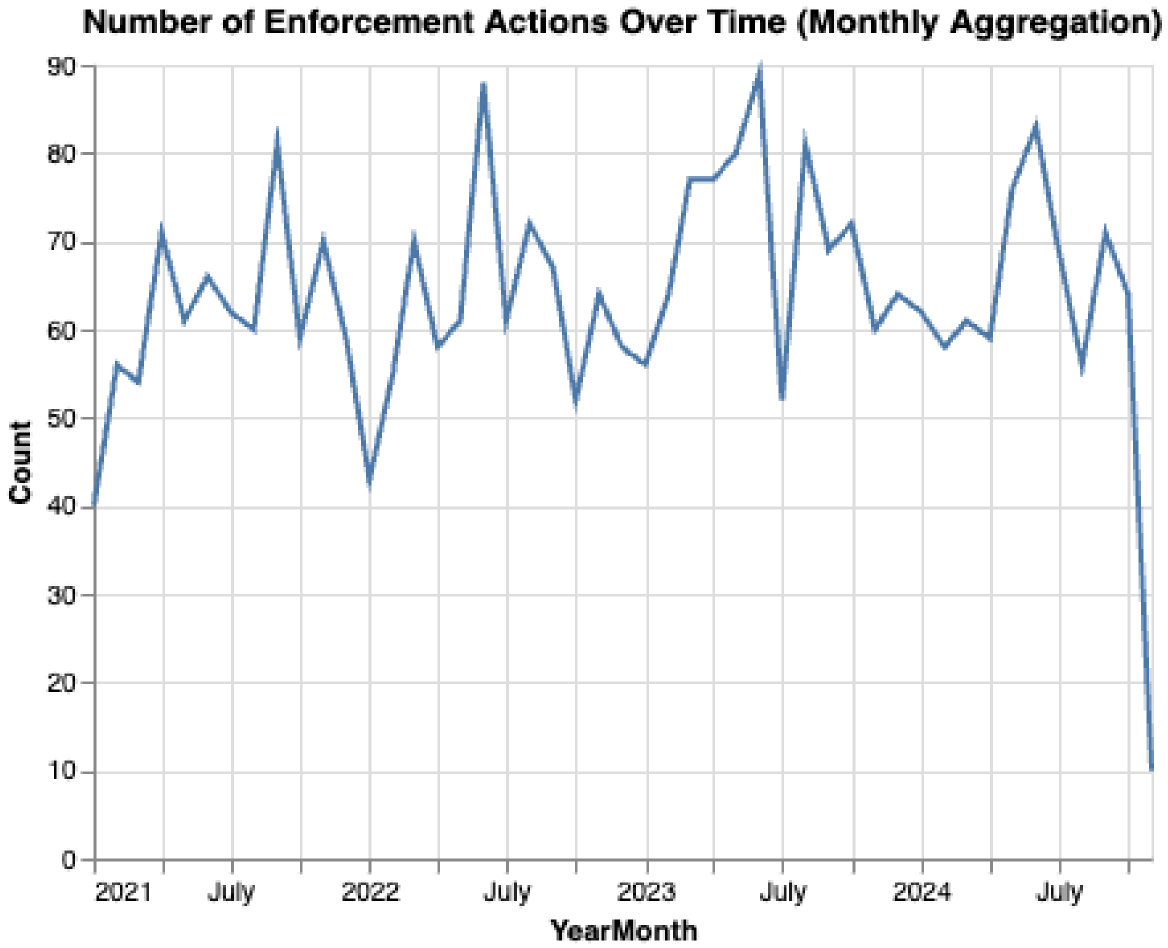


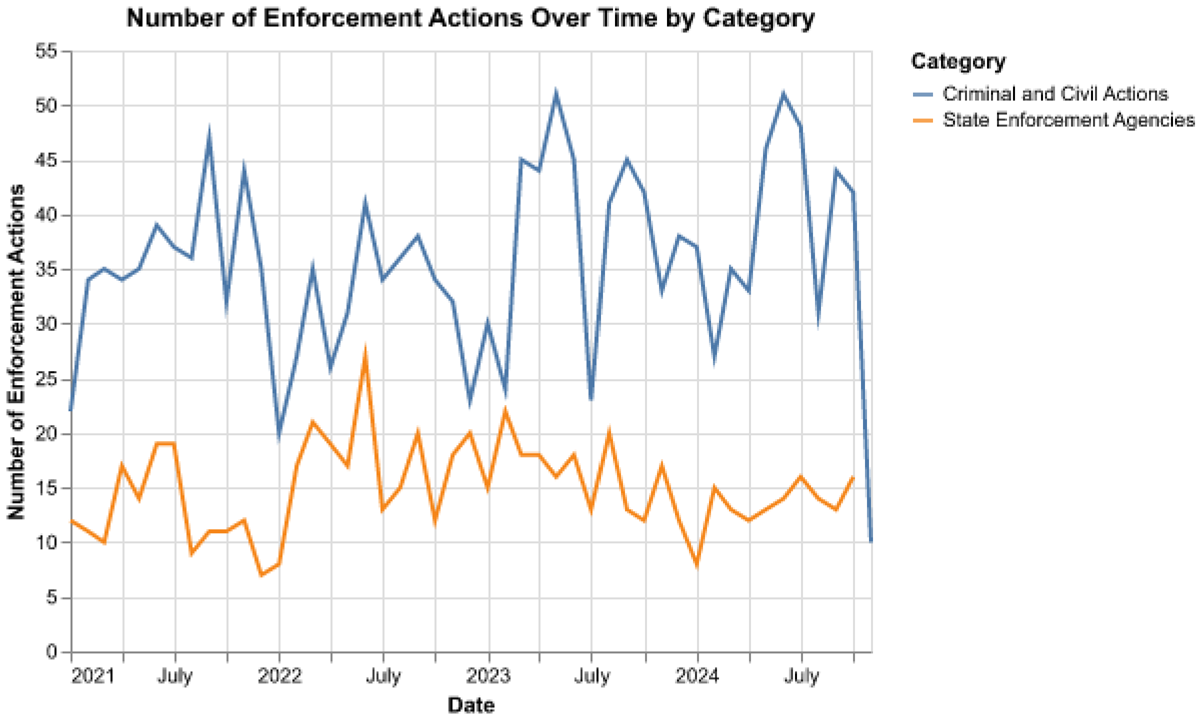
Figure 1: Number of Enforcement Actions Over time

2. Plot the number of enforcement actions categorized: (PARTNER 1)

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

```
df['Date'] = pd.to_datetime(df['Date'])

df['Year_Month'] = df['Date'].dt.to_period('M')

monthly_counts = df.groupby(['Year_Month',
↪  'Category']).size().reset_index(name='Count')
monthly_counts['Year_Month'] = monthly_counts['Year_Month'].dt.to_timestamp()
↪
```

```
monthly_counts = monthly_counts[monthly_counts['Category'].isin(["Criminal
 ↪  and Civil Actions", "State Enforcement Agencies"])]

line_chart = alt.Chart(monthly_counts).mark_line().encode(
    x=alt.X('Year_Month:T', title='Date'),
    y=alt.Y('Count:Q', title='Number of Enforcement Actions'),
    color=alt.Color('Category:N', title='Category'),
    tooltip=['Year_Month:T', 'Category:N', 'Count:Q']
).properties(
    title='Number of Enforcement Actions Over Time by Category',
    width=400,
    height=300
)

line_chart
```



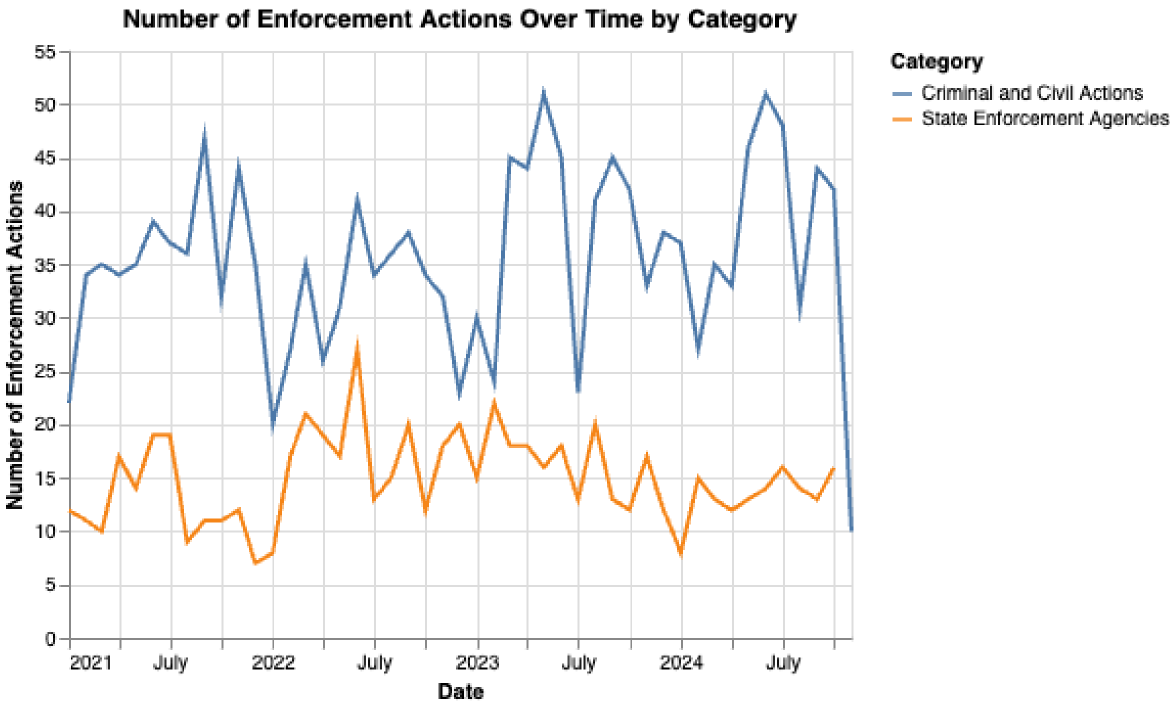


Figure 2: criminal vs. enforcement

- based on five topics

```
def classify_topic(title):
    """Classifies each action title into one of the five topics or 'State
     ↪  Enforcement Agencies'."""
    title = title.lower()
    if "health" in title or "care" in title:
        return "Health Care Fraud"
    elif "financial" in title or "bank" in title or "money" in title:
        return "Financial Fraud"
    elif "drug" in title or "narcotics" in title:
        return "Drug Enforcement"
    elif "bribery" in title or "corruption" in title or "bribe" in title:
        return "Bribery/Corruption"
    else:
        return "Other"

df['Topic'] = df.apply(
    lambda row: classify_topic(row['Title']) if row['Category'] == "Criminal
     ↪  and Civil Actions" else "State Enforcement Agencies",
    axis=1
)

monthly_counts = df.groupby(['Year_Month', 'Topic']).size().reset_index(name='Count')
monthly_counts['Year_Month'] = monthly_counts['Year_Month'].dt.to_timestamp()

line_chart = alt.Chart(monthly_counts).mark_line().encode(
    x=alt.X('Year_Month:T', title='Date'),
    y=alt.Y('Count:Q', title='Number of Enforcement Actions'),
    color=alt.Color('Topic:N', title='Topic', scale=alt.Scale(domain=[
        "Health Care Fraud", "Financial Fraud", "Drug Enforcement",
        "Bribery/Corruption", "Other", "State Enforcement Agencies"
    ])),
    tooltip=['Year_Month:T', 'Topic:N', 'Count:Q']
).properties(
    title='Number of Enforcement Actions Over Time by Topic',
    width=400,
    height=300
)

line_chart
```

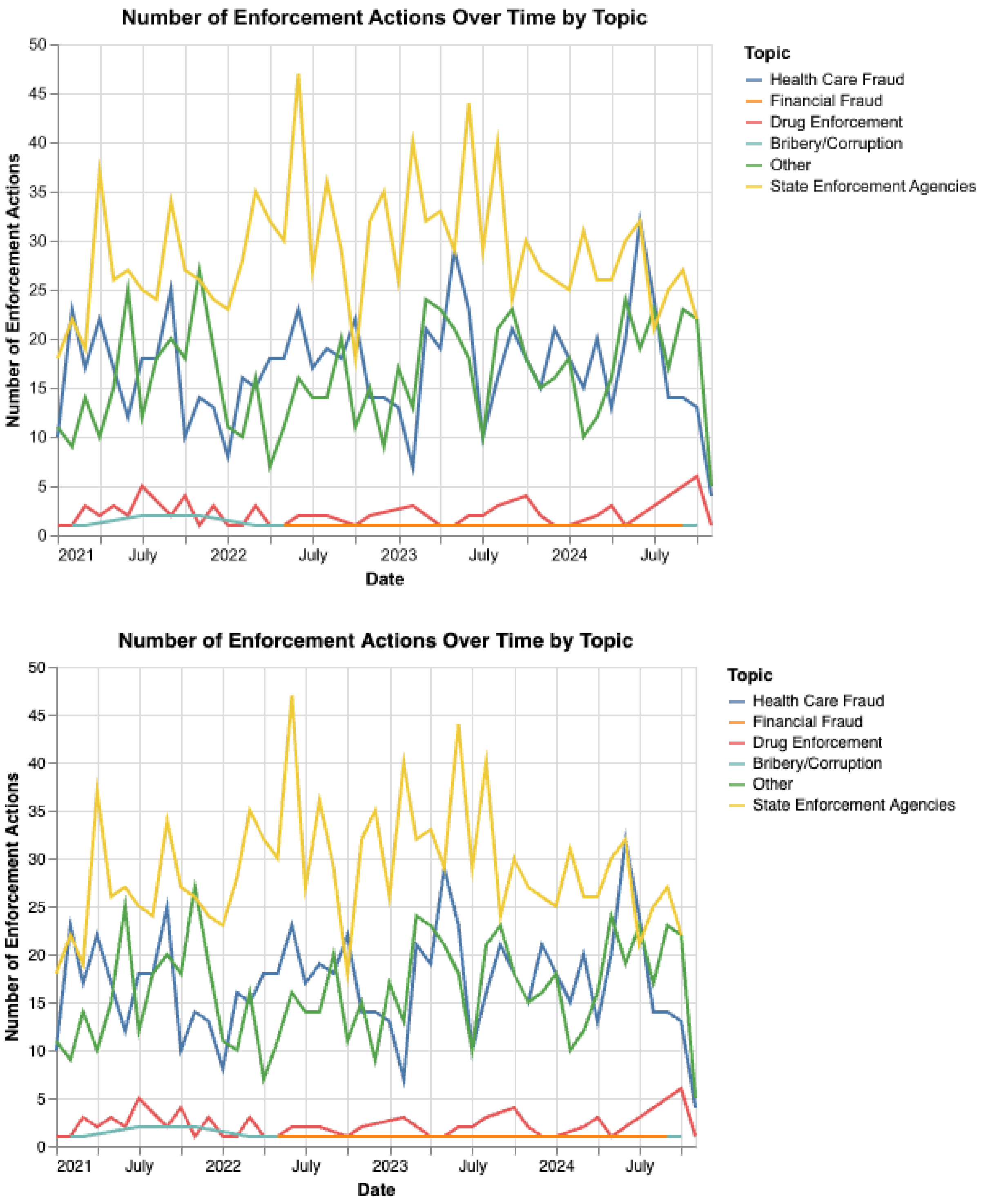


Figure 3: five topics

Step 4: Create maps of enforcement activity

1. Map by State (PARTNER 1)

```python
import geopandas as gpd
import matplotlib.pyplot as plt

enforcement_data = pd.read_csv('enforcement_actions_2023_1.csv')

state_shapefile_path = 'cb_2018_us_state_5m.shp'
states = gpd.read_file(state_shapefile_path)
```

```python
state_actions =
 ↪  enforcement_data[enforcement_data['Agency'].str.contains("State of",
 ↪  na=False)]

state_actions['State'] = state_actions['Agency'].str.extract(r"State of
 ↪  (\w+)")
state_actions['State'] = state_actions['State'].str.strip()

state_counts = state_actions['State'].value_counts().reset_index()
state_counts.columns = ['State', 'Enforcement_Count']

state_choropleth = states.merge(state_counts, how="left", left_on="NAME",
 ↪  right_on="State")

plt.figure(figsize=(15, 10))
state_choropleth.plot(column='Enforcement_Count', cmap='OrRd', legend=True,
 ↪  edgecolor="black")
plt.title("State-Level Enforcement Actions by State")
plt.axis("off")
plt.show()
```

```
<Figure size 4500x3000 with 0 Axes>
```

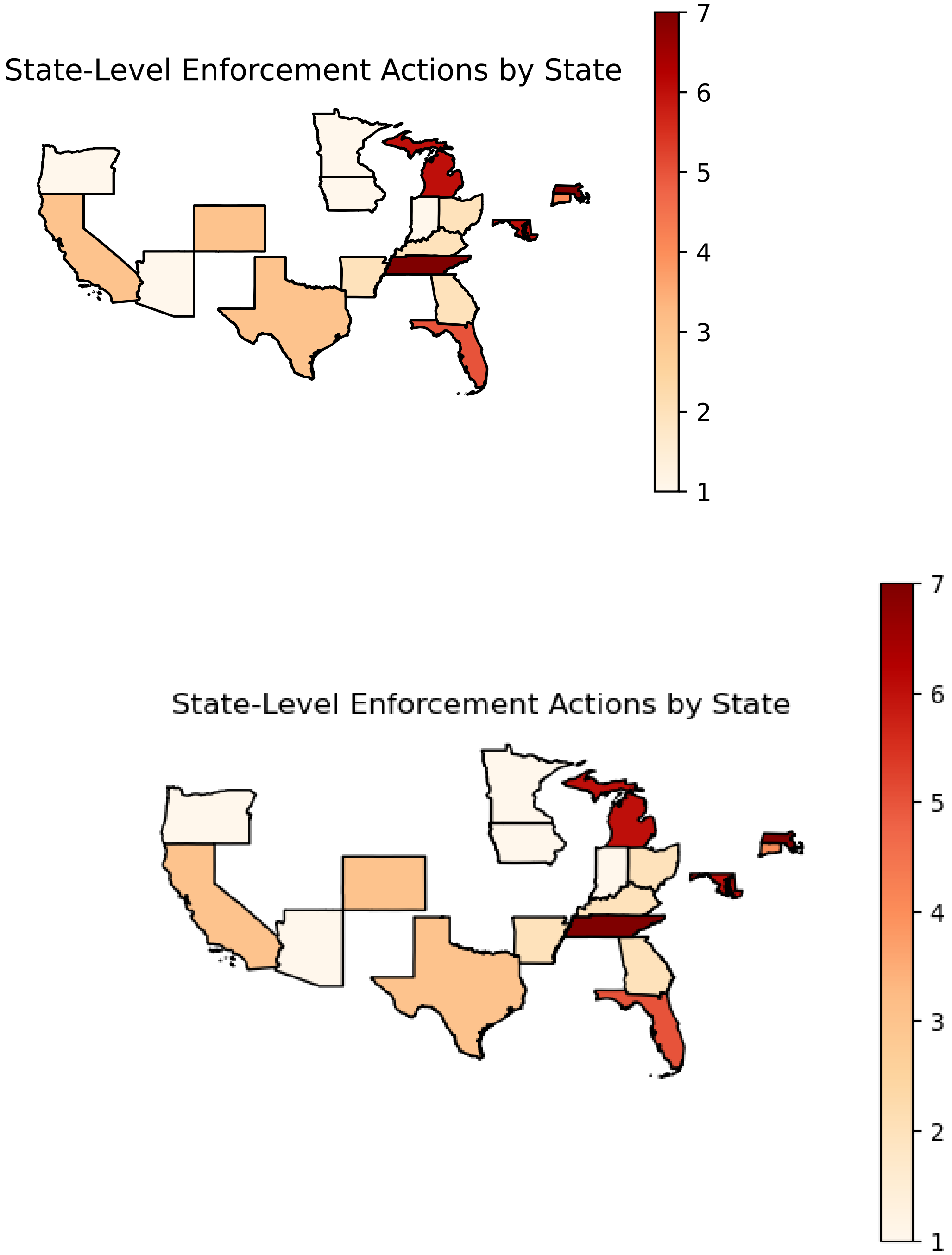

State-Level Enforcement Actions by State
7
6
5
4
3
2
1
State-Level Enforcement Actions by State
7
6
5
4
3
2
1

Figure 4: map by state

2. Map by District (PARTNER 2)

```
import re

district_shapefile_path =
 ↪  'geo_export_fcd06d4e-838a-449a-979d-dfc51a522ff4.shp'
district = gpd.read_file(district_shapefile_path)
enforcement_data = pd.read_csv('enforcement_actions_2023_1.csv')

district_actions =
 ↪  enforcement_data[enforcement_data['Agency'].str.contains("District",
 ↪  na=False)]

district_names =
 ↪  district_actions['Agency'].str.extract(r"(Western|Eastern|Northern|Southern|Central)?\s?D
 ↪  of (\w+)")

district_actions['District'] = district_names[0].fillna('') + ' District of '
 ↪  + district_names[1]

district_counts = district_actions['District'].value_counts().reset_index()
district_counts.columns = ['District', 'Enforcement_Count']

district_counts['District'] = district_counts['District'].str.strip()
district['judicial_d'] = district['judicial_d'].str.strip()

district_choropleth = district.merge(district_counts, how="left",
 ↪  left_on="judicial_d", right_on="District")

fig, ax = plt.subplots(figsize=(10, 5))

district_choropleth.plot(column='Enforcement_Count', cmap='Blues',
 ↪  legend=True, edgecolor="black", ax=ax)

ax.set_xlim(-130, -65)
ax.set_ylim(24, 50)

plt.title("US Attorney District-Level Enforcement Actions", fontsize=18)
plt.axis("off")

plt.show()
```

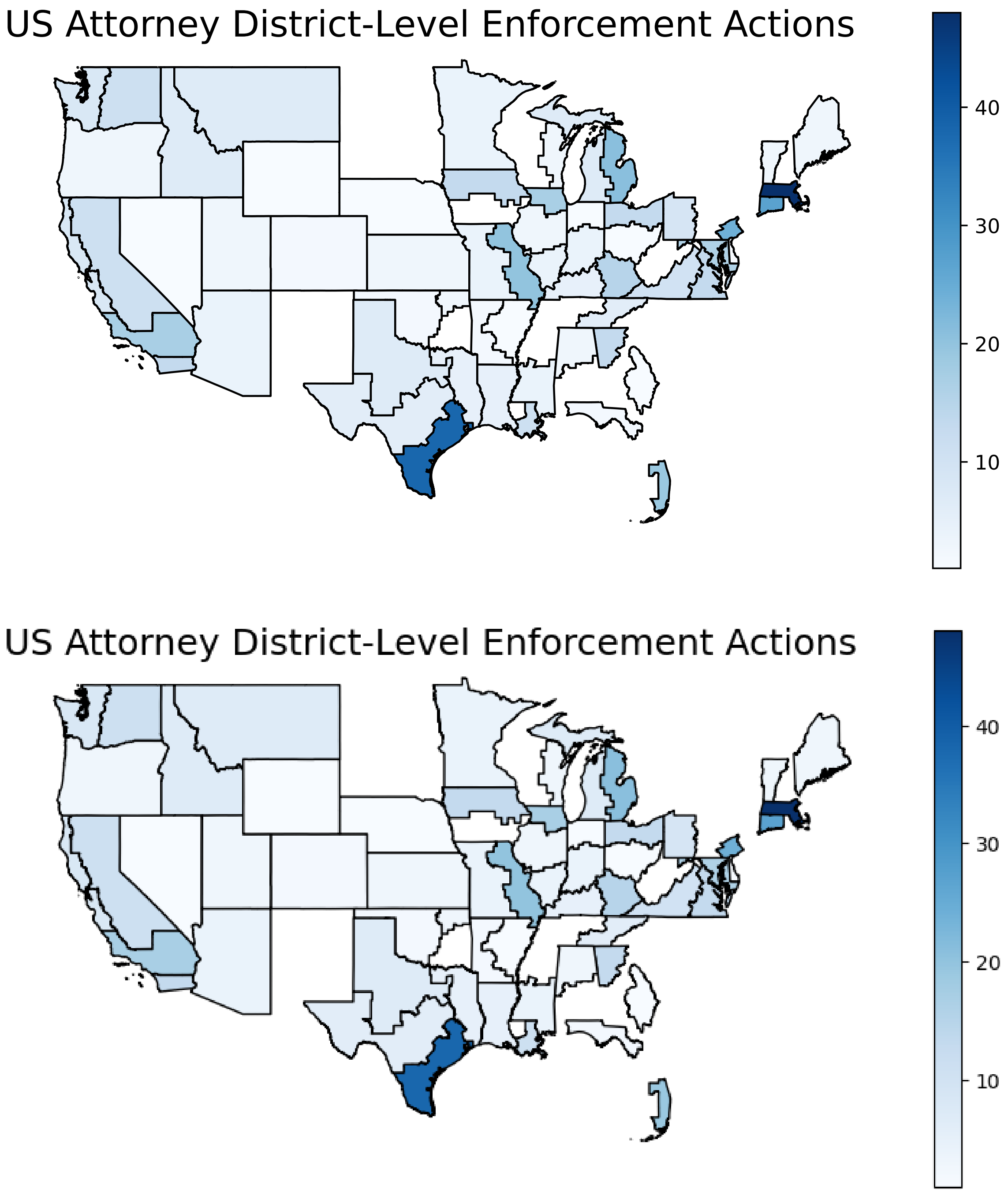

US Attorney District-Level Enforcement Actions
40
30
20
10
US Attorney District-Level Enforcement Actions
40
30
20
10

Figure 5: map by district