Pset V

Summer Negahdar & Jenny Zhong

Partner 1: Summer Negahdar(samarneg) Partner 2: This submission is our work alone and complies with the 30538 integrity policy.” Add your initials to indicate your agreement: **___** **“I have uploaded the names of anyone else other than my partner and I worked with on the problem set here” Late coins used this pset:** Late coins left after submission: **___**

Develop Initial scraper and crawler

1.

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

def scrape_all_pages(base_url):
    current_url = base_url  # Start with the base URL
    all_data = []  # List to store all scraped data

    while True:
        response = requests.get(current_url)
        soup = BeautifulSoup(response.content, 'html.parser')

        # Find all actions based on the main <li> tag containing each card
        actions = soup.find_all('li', class_='usa-card card--list pep-card--minimal
↪   mobile:grid-col-12')

        for action in actions:
            title_tag = action.find('h2', class_='usa-card__heading')
            title = title_tag.get_text(strip=True) if title_tag else 'No Title Provided'

            link = title_tag.find('a')['href'] if title_tag and title_tag.find('a') else
↪   'No Link Provided'
            link = f"https://oig.hhs.gov{link}" if link.startswith('/') else link

            # Correctly extract the date from the new structure
            date_div = action.find('div', class_='font-body-sm margin-top-1')
            date = date_div.find('span', class_='text-base-dark
↪   padding-right-105').get_text(strip=True) if date_div else 'No Date Provided'
```

```
            # Correctly extract the category from the new structure
            category_ul = action.find('ul', class_='display-inline add-list-reset')
            category = category_ul.find('li').get_text(strip=True) if category_ul else
 ↪  'No Category Provided'

            all_data.append({
                'Title': title,
                'Date': date,
                'Category': category,
                'Link': link
            })

        # Find the second 'ul' which contains the pagination and access the 6th 'li' for
         ↪  the next link
        pagination = soup.find_all('ul', class_='pagination')
        if len(pagination) >= 2 and len(pagination[1].find_all('li')) >= 6:
            next_page_li = pagination[1].find_all('li')[5]  # The sixth <li> in the
 ↪  second <ul>
            next_link = next_page_li.find('a')
            if next_link and 'href' in next_link.attrs:
                current_url = f"https://oig.hhs.gov{next_link['href']}"
            else:
                break  # Stop if no next link
        else:
            break  # Stop if pagination is missing or not enough items

        time.sleep(1)  # Sleep for 1 second between page requests to be polite to the
 ↪  server

    return pd.DataFrame(all_data)

# Base URL of the site to scrape
base_url = 'https://oig.hhs.gov/fraud/enforcement/'

# Scrape the data
final_df = scrape_all_pages(base_url)

# Print the first few rows of the DataFrame to check
print(final_df.head())

# Save the DataFrame to a CSV file
final_df.to_csv("enforcement_actions_all_pages.csv", index=False)
print("Data scraped and saved to enforcement_actions_all_pages.csv")
```

```
/Users/samarnegahdar/Desktop/untitled folder/problem-set-5-summer-jenny/Pset
V/.venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2
only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL
2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
                                               Title              Date  \
0  Pharmacist and Brother Convicted of $15M Medic...  November 8, 2024   
1  Boise Nurse Practitioner Sentenced To 48 Month...  November 7, 2024   
2  Former Traveling Nurse Pleads Guilty To Tamper...  November 7, 2024   
3  Former Arlington Resident Sentenced To Prison ...  November 7, 2024   
4  Paroled Felon Sentenced To Six Years For Fraud...  November 7, 2024   

                   Category  \
0  Criminal and Civil Actions   
1  Criminal and Civil Actions   
2  Criminal and Civil Actions   
3  Criminal and Civil Actions   
4  Criminal and Civil Actions   

                                                Link  
0  https://oig.hhs.gov/fraud/enforcement/pharmaci...  
1  https://oig.hhs.gov/fraud/enforcement/boise-nu...  
2  https://oig.hhs.gov/fraud/enforcement/former-t...  
3  https://oig.hhs.gov/fraud/enforcement/former-a...  
4  https://oig.hhs.gov/fraud/enforcement/paroled-...  
Data scraped and saved to enforcement_actions_all_pages.csv
```

2.

```python
# Initialize an empty list to store agency names
agencies = []

# Loop through each link in final_df
for index, row in final_df.iterrows():
    link = row['Link']
    if link:  # Only proceed if the link is valid
        try:
            response = requests.get(link)  # Request the page using the link
            soup = BeautifulSoup(response.text, 'html.parser')  # Parse the content of
 ↪  the page

            # Find all <ul> elements with the specified class containing the agency
             ↪  details
            uls = soup.find_all("ul", class_="usa-list usa-list--unstyled margin-y-2")

            # Initialize a placeholder for the agency name
            agency_name = 'N/A'

            # Iterate over each <ul> element
            for ul in uls:
                # Find all <span> elements within each <ul> that match the class
                spans = ul.find_all("span", class_="padding-right-2 text-base")

                # Ensure there are enough <span> tags to avoid IndexError
                if len(spans) > 1:
                    agency = spans[1]  # Select the second <span>, which contains
 ↪  "Agency:"

                    # Use next_sibling to access the text following the <span>
                    agency_name = agency.next_sibling.strip() if agency.next_sibling else
 ↪  'N/A'

                    # Stop after finding the first matching <ul> and <span> structure
                    break

            # Append the extracted agency name to the agencies list
            agencies.append(agency_name)

        except requests.exceptions.RequestException as e:
            print(f"Error fetching {link}: {e}")
            agencies.append('N/A')

# Add agency names to the DataFrame and print its head
final_df['Agency'] = agencies  # Create a new column in our original df called Agency
print(final_df.head(5))
```

```
                                               Title              Date  \
0  Pharmacist and Brother Convicted of $15M Medic...  November 8, 2024
1  Boise Nurse Practitioner Sentenced To 48 Month...  November 7, 2024
2  Former Traveling Nurse Pleads Guilty To Tamper...  November 7, 2024
3  Former Arlington Resident Sentenced To Prison ...  November 7, 2024
4  Paroled Felon Sentenced To Six Years For Fraud...  November 7, 2024

                     Category  \
0  Criminal and Civil Actions
1  Criminal and Civil Actions
2  Criminal and Civil Actions
3  Criminal and Civil Actions
4  Criminal and Civil Actions

                                                Link  \
0  https://oig.hhs.gov/fraud/enforcement/pharmaci...
1  https://oig.hhs.gov/fraud/enforcement/boise-nu...
2  https://oig.hhs.gov/fraud/enforcement/former-t...
3  https://oig.hhs.gov/fraud/enforcement/former-a...
4  https://oig.hhs.gov/fraud/enforcement/paroled-...

                                              Agency
0                         U.S. Department of Justice
1  November 7, 2024; U.S. Attorney's Office, Dist...
2  U.S. Attorney's Office, District of Massachusetts
3  U.S. Attorney's Office, Eastern District of Vi...
4  U.S. Attorney's Office, Middle District of Flo...
```

```
##for jenny
## since you need the date column to be a date, I will convert it for you

final_df['Date'] = pd.to_datetime(final_df['Date'], errors='coerce')

# Check the data type to confirm
print(final_df.dtypes)
print(final_df.head())
```

```
Title               object
Date        datetime64[ns]
Category            object
Link                object
Agency              object
dtype: object
                                               Title       Date  \
0  Pharmacist and Brother Convicted of $15M Medic... 2024-11-08
1  Boise Nurse Practitioner Sentenced To 48 Month... 2024-11-07
2  Former Traveling Nurse Pleads Guilty To Tamper... 2024-11-07
3  Former Arlington Resident Sentenced To Prison ... 2024-11-07
4  Paroled Felon Sentenced To Six Years For Fraud... 2024-11-07

                     Category  \
0  Criminal and Civil Actions
1  Criminal and Civil Actions
2  Criminal and Civil Actions
3  Criminal and Civil Actions
4  Criminal and Civil Actions

                                                Link  \
0  https://oig.hhs.gov/fraud/enforcement/pharmaci...
1  https://oig.hhs.gov/fraud/enforcement/boise-nu...
2  https://oig.hhs.gov/fraud/enforcement/former-t...
3  https://oig.hhs.gov/fraud/enforcement/former-a...
4  https://oig.hhs.gov/fraud/enforcement/paroled-...

                                              Agency
0                         U.S. Department of Justice
1  November 7, 2024; U.S. Attorney's Office, Dist...
2  U.S. Attorney's Office, District of Massachusetts
3  U.S. Attorney's Office, Eastern District of Vi...
4  U.S. Attorney's Office, Middle District of Flo...
```

Making the scraper dynamic

1.

a. the pseudo code for writing the function will be like:

1. going thorugh every row in the df Summer has created.

2. extracting the dates from date column
3. there will be two types of date
    I. after 2013 » continue with the rest of function

    II. before 2013 » show me an error sign that says this is before our desired timeline
4. save all the extracted dates on a csv file called “enfrocment_actions_month_year.csv”
5. do not push it to git
6. print the head

b.

```python
desired_dates= []
for index, row in final_df.iterrows():
    date = row['Date']

    # Check if date is before or after 2013
    if date.year >= 2013:
        # If date is after 2013, add it to the list
        desired_dates.append(date)
    else:
        # If date is before 2013, print an error message
        print("outside desired period")

# Create a new DataFrame to save the valid dates
desired_dates_df = pd.DataFrame(desired_dates, columns=['Date'])

desired_dates_df.to_csv("enforcement_actions_month_year.csv", index=False)
print("Unique years in the data:", final_df['Date'].dt.year.unique())
```

```
Unique years in the data: [2024]
```

c.

Plot data based on scraped data (using Altair)

1.

2.

Create maps of enforcement activity

1.

2.

Extra Credit: Calculate the enforcement actions on a per-capita basis

1.

2.

3.