Problem Set 4

Rachel Alper

2001-02-07

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: RA

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 requests
from bs4 import BeautifulSoup

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

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

Step 1: Develop initial scraper and crawler

```
with open(r'hhs_website.html', 'r') as page:
    hhs_website = page.read()
    soup = BeautifulSoup(hhs_website, 'lxml')
```

```
records = []

cards = soup.find_all("li", class_="usa-card")

for card in cards:

    title = card.find("h2").get_text(strip=True)

    date = card.find("span").get_text(strip=True)

    category_tag = card.find("li", class_="usa-tag")
    category = category_tag.get_text(strip=True) if category_tag else None

    records.append({
        "title": title,
        "date": date,
        "category": category
    })


df = pd.DataFrame(records)
df["date"] = pd.to_datetime(df["date"])
```

```
df.head()
```

	title	date	category
0	Brooklyn Banker Pleads Guilty to Laundering Pr...	2026-02-03	COVID-19
1	Delafield Man Sentenced to 18 Months' Imprison...	2026-02-03	Criminal and Civil Actions
2	Former NFL Player Convicted for $197M Medicare...	2026-02-03	Criminal and Civil Actions
3	AG's Office Secures Indictments Against Peabod...	2026-02-02	State Enforcement Agencies
4	Florida Man Pleads Guilty to Conspiracy to Vio...	2026-01-30	Criminal and Civil Actions

Step 2: Making the scraper dynamic

1. Turning the scraper into a function

- a. Pseudo-Code
- b. Create Dynamic Scraper
- c. Test Your Code

Step 3: Plot data based on scraped data

1. Plot the number of enforcement actions over time

2. Plot the number of enforcement actions categorized:

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