30538 Problem Set 5: Web Scraping

Peter Ganong, Maggie Shi, Akbar Saputra, and Will Pennington

2024-11-03

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):
    - Partner 2 (name and cnet ID):
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: **___** **___**
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: **___** Late coins left after submission: **___**
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

(30 points) Step 1: Develop initial scraper and crawler

1. (Partner 1) **Scraping**: Go to the first page of the HHS OIG's "Enforcement Actions" page and scrape and collect the following into a dataset:

    - Title of the enforcement action
    - Date
    - Category (e.g, "Criminal and Civil Actions")
    - Link associated with the enforcement action

Collect your output into a tidy dataframe and print its `head`.

2. (Partner 1) **Crawling**: Then for each enforcement action, click the link and collect the name of the agency involved (e.g., for this link, it would be U.S. Attorney's Office, Eastern District of Washington).

Update your dataframe with the name of the agency and print its `head`.

Hint: if you go to James A. Robinson's profile page at the Nobel Prize website here, right-click anywhere along the line "Affiliation at the time of the award: University of Chicago, Chicago, IL, USA", and select Inspect, you'll see that this affiliation information is located at the third `<p>` tag out of five `<p>` tags under the `<div class="content">`. Think about how you can select the third element of `<p>` out of five `<p>` elements so you're sure you scrape the affiliation information, not other. This way, you can scrape the name of agency to answer this question.

(30 points) Step 2: Making the scraper dynamic

1. **Turning the scraper into a function**: You will write a function that takes as input a month and a year, and then pulls and formats the enforcement actions like in Step 1 starting from that month+year to today.

    - This function should first check that the year inputted >= 2013 before starting to scrape. If the year inputted < 2013, it should print a statement reminding the user to restrict to year >= 2013, since only enforcement actions after 2013 are listed.
    - It should save the dataframe output into a .csv file named as "enforcement_actions_*year_month*.csv" (do not commit this file to git)
    - If you're crawling multiple pages, always add 1 second wait before going to the next page to prevent potential server-side block. To implement this in Python, you may look up `.sleep()` function from `time` library.

a. (Partner 2) Before writing out your function, write down pseudo-code of the steps that your function will go through. If you use a loop, discuss what kind of loop you will use and how you will define it.

b. (Partner 2) Now code up your dynamic scraper and run it to start collecting the enforcement actions since January 2023. How many enforcement actions do you get in your final dataframe? What is the date and details of the earliest enforcement action it scraped?
c. (Partner 1) Now, let's go a little further back. Test your partner's code by collecting the actions since January 2021. *Note that this can take a while.* How many enforcement actions do you get in your final dataframe? What is the date and details of the earliest enforcement action it scraped? Use the dataframe from this process for every question after this.

Hint:

a. If you go to the next page in this HHS OIG's "Enforcement Actions" page, you'll notice a pattern:

   - Second page URL: https://oig.hhs.gov/fraud/enforcement/?page=2
   - Third page URL: https://oig.hhs.gov/fraud/enforcement/?page=3
   - and so on …

b. Write a pseudo-code to help you think about how to make the crawler dynamic. You need to loop through all the pages in the website. *Hint: Note that a simple* `for` *loop may not be sufficient for what this crawler requires. Use online resources to look into different types of loops or different ways of using* `for` *loops to see if there is something that is more appropriate for this task.*

(15 points) Step 3: Plot data based on scraped data (using `altair`)

1. (Partner 2) Plot a line chart that shows: **the number of enforcement actions** over time (aggregated to each month+year) overall since January 2021,
2. (Partner 1) Plot a line chart that shows: **the number of enforcement actions** split out by:

   - "Criminal and Civil Actions" vs. "State Enforcement Agencies"
   - Five topics in the "Criminal and Civil Actions" category: "Health Care Fraud", "Financial Fraud", "Drug Enforcement", "Bribery/Corruption", and "Other". *Hint*: You will need to divide the five topics manually by looking at the title and assigning the relevant topic. For example, if you find the word "bank" or "financial" in the title of an action, then that action should probably belong to "Financial Fraud" topic.

(15 points) Step 4: Create maps of enforcement activity

For these questions, use this US Attorney District shapefile (link) and a Census state shapefile (link)

1. (Partner 1) **Map by state**: Among actions taken by state-level agencies, clean the state names you collected and plot a choropleth of the number of enforcement actions for each state. *Hint*: look for "State of" in the agency info!
2. (Partner 2) **Map by district**: Among actions taken by US Attorney District-level agencies, clean the district names so that you can merge them with the shapefile, and then plot a choropleth of the number of enforcement actions in each US Attorney District. *Hint*: look for "District" in the agency info.

(10 points) Extra credit: Calculate the enforcement actions on a per-capita basis

(Both partners can work together)

1. Use the zip code shapefile from the previous problem set and merge it with zip code-level population data. (Go to Census Data Portal, select "ZIP Code Tabulation Area", check "All 5-digit ZIP Code Tabulation Areas within United States", and under "P1 TOTAL POPULATION" select "2020: DEC Demographic and Housing Characteristics". Download the csv.).
2. Conduct a spatial join between zip code shapefile and the district shapefile, then aggregate to get population in each district.
3. Map the ratio of enforcement actions in each US Attorney District. You can calculate the ratio by aggregating the number of enforcement actions since January 2021 per district, and dividing it with the population data.