CSCI 452 (Data Mining)
Basic HTML Web Scraping
75 pts

Overview

For this assignment, you'll write several small python programs to scrape simple HTML data from several websites. You will use Python 3 with the following libraries:

- Beautiful Soup 4 (makes it easier to pull data out of HTML and XML documents)
- Requests (for handling HTTP requests from python)
- lxml (XML and HTML parser)

Here is a fairly simple example for finding out how many datasets can currently be searched/accessed on data.gov. You should make sure you can run this code before going on to the questions you'll be writing (the answer when I last ran this was 195,384).

```
import bs4
import requests
response = requests.get('http://www.data.gov/')
soup = bs4.BeautifulSoup(response.text,"lxml")
link = soup.select("small a")[0]
print(link.text)
#Credit to Dan Nguyen at Stanford's Computational Journalism program
```

Specification

Write python programs to answer the following questions. Be sure to follow the specified output format (including prompts) carefully since it will be autograded. You will need to do some reading/research regarding the Beautiful Soup interface and possibly on Python as well. There are links to the documentation on my website. Do not hardcode any data; everything should be dynamically scraped from the live websites. Points will be awarded on functionality, but there will be a code inspection. If values are hardcoded or if the style/commenting is insufficient, points will be deducted.

1. (30 pts) Data.gov (relevant url, http://catalog.data.gov/dataset?q=&sort=metadata_created+desc): accept an integer as input and find the name (href text) of the nth "most recent" dataset on data.gov. For example, if the user enters 1, print the name of the **first** dataset on data.gov when ordered by "date added". You can assume that the dataset appears on the first page. **name this program datagov.py**

Government shutdown workaround – use this link:
https://web.archive.org/web/20170529013103/https://catalog.data.gov/dataset?q=&sort=metadata_created+desc

Example (based on data when viewed on 8/15/2016, user input in bold):
Which dataset? **5**
NNDSS - Table II. Invasive Pneumococcal to Legionellosis

2. (45 pts) Texas Dept of Criminal Justice (relevant url: http://www.tdcj.state.tx.us/death_row/dr_executions_by_year.html: Accept two integers as input. You can assume that these values represent a valid starting and ending year within the range of the years in the table. Process the html and find the total number of executions in Texas between the starting year and the ending year (inclusive of the start and end years). **name this program texec.py**

   Example (user input in bold):
   Enter starting year: **1990**
   Enter ending year: **2000**
   Total executions: 206

Create a .zip file containing your three python programs, named as specified above. Submit your .zip file as the HTML Scraping lab to Autolab.