Web Scraping and Text Processing with Python 2015

Tuesday, May 26th – Friday, May 29th
Morning Session: 10:00am – 11:30am
Afternoon Session: 1:30pm – 3:00pm
Location: Sherrerd Hall 101

Instructor:	Hubert Jin
Office:	Corwin 029
Email:	hubertj@princeton.edu
Office Hours:	3:30-4:30PM

Description Over the last decade, both the variety and amount of data available to social scientists have expanded. These new data sources include administrative records (e.g., voter files, campaign finance and lobbying records), geo-referenced data (e.g., satellite maps, geocoded event data), and texts (e.g., speeches, court rulings, legislative bills). Many of these data sources can be accessed through the World Wide Web and as a consequence, techniques such as web scraping have become an essential part of social scientists' toolkit. The objective of this workshop is to introduce basic tools and techniques for automatic content extraction, parsing and other data-handling tasks that are commonly encountered in data-intensive research projects. The course will be taught in Python, and requires the basic knowledge of Python programming (such as the Introduction to Python workshop taught by the Office of Population Research). We will cover techniques ranging from Python regular expressions and file manipulation, to the popular HTML/XML parsing library "Beautiful Soup" and PDF content extraction. The course ends with an introduction to the Twitter API for accessing Twitter content.

Prerequisites. Although anyone can sit in on the short course, the following is assumed as a starting point:

- Basic knowledge of the Python programming language. The Office of Population Research will offer a Python workshop in early/mid May 2015. It is highly recommended that students take a class like that. (here is their last year's website http://opr.princeton.edu/workshops/201404/)
- Access to a computer that has been setup according to provided instructions (see below for details).

Structure. Because a lot of material will be covered over the course of a week, this camp is very much an immersion. We will meet each day from Tuesday through Friday during both a morning session and an afternoon session. These sessions will be on the *interactive* and *hands-on* end of the spectrum, so bringing a laptop to the camp is strongly recommended (and necessary for getting the most out of the sessions). Prior to some sessions, handouts introducing material with demonstrations will be distributed. Then, during the session, we will work through applying these techniques to actual data harvest and scraping problems in Political Science.

Discussion Board. We will the Piazza discussion board (`https://piazza.com/`) to facilitate discussions and questions throughout the Web Scraping and Text Processing with Python short course. Piazza provides an interactive environment in which to both ask questions and answer those of others. To join the Web Scraping and Text Processing with Python Piazza site, click on "Search Your Classes" from the Piazza homepage. After specifying Princeton University as your school, search for "Web Scraping and Text Processing with Python". You will then be prompted to enter your `princeton.edu` email address to confirm your registration. Piazza can also be accessed from within Blackboard by going to the Web Scraping and Text Processing with Python organization page and clicking on the link to "Piazza Q&A". In addition, all class announcements will be made through Piazza. Blackboard will still be used for hosting all class materials and for submitting assignments.

Some tips and tricks for Piazza include:

- Piazza has apps available for the iOS and Android platforms. The apps are free downloads and provide complete access to all of Piazza's messageboard features.
- To insert LaTeX-formatted text in a post, place a double dollar sign ("$$") on both ends of the relevant text, or click the "fx" button in the Details toolbar above your post.
- To add formatted Python code to a post, click the "pre" button in the Details toolbar above your post. A grey text box will open up where you can paste code from Python. You can classify a post using pre-selected tags, or you can generate your own by prepending a hash (#) to your chosen label. Posts can then be sorted by these tags using the search bar in the left-hand column.
- We encourage you to mark helpful contributions (particularly those from classmates) using the "Thanks!" button at the bottom of each post.

Machine Setup. Instructions detailing setup instructions will be available on Blackboard through announcement. They mostly require registration for access to high-performance computing resources through Princeton University's Research Computing department.

Materials. No outside materials are necessary for this course. However, if your own work follows along the trajectory that we take in this course, the following *optional* supporting resources will likely be helpful.

- Learning Python (`http://shop.oreilly.com/product/0636920028154.do`). This is a a comprehensive, in-depth introduction to the core Python language with this hands-on book. Based on author Mark Lutz's popular training course, this updated fifth edition will help you quickly write efficient, high-quality code with Python. It's an ideal way to begin, whether youâĂŹre new to programming or a professional developer versed in other languages.
- HTML, The Web's Core Language (`http://www.w3.org/html/`). HTML is the Web's core language for creating documents and applications for everyone to use, anywhere.
- The Unicode Consortium (`http://unicode.org/`) The Unicode Consortium enables people around the world to use computers in any language. Our freely-available specifications and data form the foundation for software internationalization in all major operating systems, search engines, applications, and the World Wide Web. An essential part of our mission is to educate and engage academic and scientific communities, and the general public.

- Meet Scrapy (`http://scrapy.org/`). An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way.

- Beautiful Soup (`http://www.crummy.com/software/BeautifulSoup/`). Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping.

- PDFMiner (`http://www.unixuser.org/~euske/python/pdfminer/index.html`) PDFMiner is a tool for extracting information from PDF documents. Unlike other PDF-related tools, it focuses entirely on getting and analyzing text data. PDFMiner allows one to obtain the exact location of text in a page, as well as other information such as fonts or lines. It includes a PDF converter that can transform PDF files into other text formats (such as HTML). It has an extensible PDF parser that can be used for other purposes than text analysis.

Topics

Day	Session	Details
Tuesday (5/26)	AM	Important strategies and overview.
	PM	Introduction to WWW/HTML/XML.
		Practice wget/curl to retrieve web content.
		Quick review of Python.
		Preview of some examples.
		Examples and practice.
Wednesday (5/27)	AM	Parse web content using BeautifulSoup.
	PM	Examples and practice.
		Process web data in JSON and etc..
		Handle Non-ASCII web data in foreign languages.
		Examples and practice.
Thursday (5/28)	AM	Website authentication and cookies.
	PM	Basics of mechanize.
		Examples and practice.
		Python data scraping using Scrapy.
		Examples and practice.
Friday (5/29)	AM	Twitter APIs, Streaming and REST.
	PM	Examples and practice.
		OCR tools, PDF, tables, and handwritten documents.
		Text processing packages, and etc.
		Summary and review.