Introduction to Python and Web Scraping

Alumni Mentor: Yunxin Fan
13rd October 2017

1. Introduction

Welcome! This workshop will cover the fundamentals of Python programming as well as an introduction to web-scraping with a focus on applications to social science research.
We will then demonstrate how to use a number of different web scraping frameworks using Congressional data from GovTrack (http://www.govtrack.us) as one case study.
The purpose of this document is to provide directions on technical set-up for this workshop. Workshop attendees are assumed to be proficient in R and/or have a basic knowledge of Python.

Overall, there are four tools we will need for our workshop:

- **Python** - a programming language.
- **Anaconda** - a high-level environment for Python, optimized for data science.
- **Web scraping packages** - third-party Python web-scraping libraries we will install into our environment.
- **Jupyter** - an easy-to-use document format for sharing code, visualizations, and equations.

2. Setup

2.1 Overview

If you are an experienced Python user, the following installations will be sufficient for the workshop:

- An installation of Python 2.7 *or* an installation of Anaconda 4.3 1

The following Python packages installed:

- beautifulsoup4
- mechanize
- tweepy
- pdfminer
- pyocr
- A current installation of Jupyter.

If you have already installed the above modules on the machine you will be using during the workshop, you are all set for the workshop. If you have not, please continue reading for instructions on setting up the necessary modules.

2.2 Installation

Anaconda + Python

Note to Linux users: if you are not familiar with the command line interface on Terminal, please refer to the Appendix for an introduction to basic commands before proceeding.

For ease of use, we will use the Anaconda environment, an all-inclusive data science-oriented environment for Python, in our workshop. The version of Python we will be using is **Python 2.7**. Installing Anaconda will automatically install the appropriate version of Python on our machine.

To install Anaconda:

1. Visit https://www.continuum.io/downloads and click on the appropriate tab.
2. Download the specific installer for **Python 2.7**:

- If you are a Mac or Windows user, double-click the installer file (**.pkg** or **.exe**). Follow the directions for a successful installation of Anaconda.
- If you are a Linux user, after downloading the installer script, open the Terminal application and change directories to the folder containing the downloaded file (e.g. /Downloads) and type:

```
bash Anaconda2-4.3.1-Linux-x86_64.sh
```

2. Follow the directions for a successful installation.
3. Open the **Anaconda-Navigator** app on your machine. The application window should look like:

You now have both Anaconda and Python on your machine.

Web scraping packages

We will proceed to install the third-party Python packages used in this workshop.

We will be using the beautifulsoup4 package for data munging, the mechanize package for more advanced web site parsing, the tweepy package as a Python API for Twitter, and PDFMiner, a package for parsing text out of PDF documents.

The Anaconda environment automatically indexes and manages your Python packages. To view all included packages, click on the Environment tab:

Fortunately, the beautifulsoup4 package is already included which you can confirm using the Search packages field.

To install the remaining packages, we will use the command line tool conda to install them. For Linux and Mac OS users, the command line can be accessed by opening the Terminal application. For Windows users, the command line can be accessed by selecting the Command Prompt application.

Open up your terminal and run the following commands in succession:

```
conda install -c conda-forge mechanize=0.2.5
conda install -c conda-forge tweepy=3.5.0
conda install -c conda-forge pdfminer=20140328
```

To confirm successful installation, find each package in your default environment (root):

Jupyter

For the workshop code demos, we will use Jupyter notebooks (**.ipynb**) to write and execute code in a block-by-block fashion and also create notes in Markdown along the way as needed.

To create your own Jupyter notebook for this workshop, open your **Anaconda Navigator** application and on the Home screen, click to launch:

Your browser should have automatically opened to the Jupyter notebook server.If not, navigating to localhost:8888/tree on a browser of your choice will also allow you access the application server. To create a notebook, navigate to a directory of your choice and select New as follows:

In your newly created notebook, you can create **cells** which may either contain code that can be dynamically executed (R interpreter style) or markdown text.

Appendix: Resources

For more details on command line usage, Python, and Jupyter, we recommend the following resources:

https://www.tjhsst.edu/~dhyatt/superap/unixcmd.html : **Basic Unix commands**.

http://commandwindows.com/command3.htm : **Basic DOS (Windows) commands**.

https://python.swaroopch.com/about_python.html : **"A Byte of Python", a free online tutorial book on setting up and using Python.**

https://jupyter-notebook-beginner-guide.readthedocs.io : **Installation and setup of Jupyter.**

https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook: **Advanced configuration of Jupyter.**