M5: XML Web Scraping Assignment

Disclaimer

As of 8/28/19 the web page that this assignment uses has a programming error. The assignment instructions ask you to click on the "XML" button. Doing so, however, causes you to follow a broken link to this address:

https://www.ncdc.noaa.gov/cag/statewide/rankings/44-tavg-201807/data/data.xml

That link should be this instead:

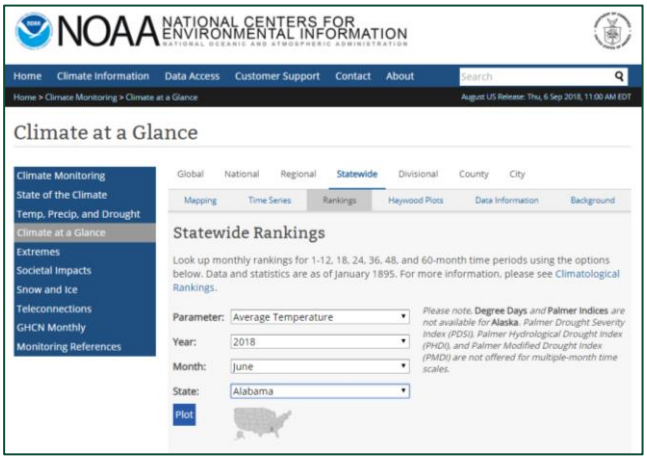
https://www.ncdc.noaa.gov/cag/statewide/rankings/44-tavg-201807/data.xml

So, please refer to this link as you proceed through the assignment.

Instructions

You will scrape data from an NOAA website in this assignment. These instructions provide considerable detail to help you construct your code.

- Use Google Chrome for the initial steps of investigating the NOAA site and determining the URL structure.
- Go to the web page provided below and choose the parameters noted below before left-clicking on the `Plot` button:
    - https://www.ncdc.noaa.gov/cag/statewide/rankings
    - Parameters:
        - `Parameter`: Average Temperature
        - `Year`: 2018
        - `Month`: November
        - `State`: Virginia



- Left-click on the `Plot` button.
- Right Click on "`XML`" button and "`Open link in new tab`."
- Observe the URL specification in that new tab and how the search parameters are embedded in it.
- Write a Python program named `xml_scrape.py` to access average temperature data in XML format using the URL structure found above. Follow these specifications:
    - Note the URL in your web browser that resulted from querying the previous parameter set and how the `Parameter`, `State`, `Month`, and `Year` parameters are indicated in the URL.
    - Build a URL to obtain the XML data for the following parameter set as directed in subsequent bullet points:
        - Parameter Set
            - `Parameter`: Average Temperature
            - `State`: Virginia
            - `Month`: August
            - `Year`: 2018
        - Create a string variable for each of the parameters above to store string values in the form that the URL requires them.
        - Assign appropriate values to those variables to obtain XML data for the parameter set above.
        - Embed these parameters into a URL using the string substitution method (see coding hint below). Start with a string that represents a "fixed" part of the URL that remains constant regardless of what parameters are of interest. Then use string substitution to embed the parameters.

- Retrieve the XML data with your program (or view it initially in a browser) and notice how you can identify the portion of the XML file that is associated with a five-month window, April–August 2018.
- Parse the data with the `lxml` package and print each of these data items on a separate line for the five-month period, April–August, 2018, without any other printed text:
  - Your W&M username (this doesn't come from the web page)
  - `value`
  - `mean`
  - `departure`
  - `lowRank`
  - `highRank`
- Name your Python file `xml_scrape.py`.
- Submit your Python code file to the assignment page in the LMS.

Coding Hint: String Substitution

Use the string substitution method for URL generation.

- Create a string template with the symbols '`%s`' as placeholders for where you will insert the values for month, year, etc.
- Here is an example in another context:
  - `template = "My name is %s, %s"`
- Then you can substitute string values for the '%s' symbols using a statement like this:
  - `last_name = 'Bradley'`
  - `first_name = 'Jim'`
  - `print template % (last_name, first_name)`
- This results in a printout of `'My name is Bradley, Jim'`
- You can use the same approach for this assignment by creating variable names for `Parameter`, `Year`, `Month`, and `State`, and substituting those values into a string template for the NOAA web page.