hw8_webscraping

October 23, 2018

1 Data-X Fall 2018: Homework 8

1.0.1 Webscraping

Authors: Alexander Fred-Ojala

In this homework, you will do some exercises with web-scraping.

STUDENT NAME : Louis Tilloy

SID : 3034388270

1.0.2 Fun with Webscraping & Text manipulation

1.1 1. Statistics in Presidential Debates

Your first task is to scrape Presidential Debates from the Commission of Presidential Debates website: http://www.debates.org/index.php?page=debate-transcripts.

To do this, you are not allowed to manually look up the URLs that you need, instead you have to scrape them. The root url to be scraped is the one listed above, namely: http://www.debates.org/index.php?page=debate-transcripts

1. By using `requests` and `BeautifulSoup` find all the links / URLs on the website that links to transcriptions of **First Presidential Debates** from the years [2012, 2008, 2004, 2000, 1996, 1988, 1984, 1976, 1960]. In total you should find 9 links / URLs tat fulfill this criteria. Print the urls.

2. When you have a list of the URLs your task is to create a Data Frame with some statistics (see example of output below):

    1. Scrape the title of each link and use that as the column name in your Data Frame.
    2. Count how long the transcript of the debate is (as in the number of characters in transcription string). Feel free to include \ characters in your count, but remove any breakline characters, i.e. `\n`. You will get credit if your count is +/- 10% from our result.
    3. Count how many times the word **war** was used in the different debates. Note that you have to convert the text in a smart way (to not count the word **warranty** for example, but counting **war.**, **war!**, **war,** or **War** etc.
    4. Also scrape the most common used word in the debate, and write how many times it was used. Note that you have to use the same strategy as in 3 in order to do this.

Print your final output result.

Tips:

In order to solve the questions above, it can be useful to work with Regular Expressions and explore methods on strings like `.strip()`, `.replace()`, `.find()`, `.count()`, `.lower()` etc. Both are very powerful tools to do string processing in Python. To count common words for example I used a `Counter` object and a Regular expression pattern for only words, see example:

```
from collections import Counter
import re

counts = Counter(re.findall(r"[\w']+", text.lower()))
```

Read more about Regular Expressions here: https://docs.python.org/3/howto/regex.html
Example output of all of the answers to Question 1.2:

.

```
In [1]: import requests
        import bs4 as bs

        source = requests.get("http://www.debates.org/index.php?page=debate-transcripts")
        soup = bs.BeautifulSoup(source.content, features='html.parser')
```

```
In [2]: blockquotes = soup.find_all("blockquote")
        first_links = [item.a.get("href") for item in blockquotes]
        first_names = [item.a.text for item in blockquotes]
        years = [2012, 2008, 2004, 2000, 1996, 1988, 1984, 1976, 1960]

        urls = []
        names = []
        for year in years:
            for link, name in zip(first_links, first_names):
                if str(year) in link:
                    urls.append(link)
                    names.append(name)
                    break

        urls
```

```
Out[2]: ['http://www.debates.org/index.php?page=october-3-2012-debate-transcript',
         'http://www.debates.org/index.php?page=2008-debate-transcript',
         'http://www.debates.org/index.php?page=october-13-2004-debate-transcript',
         'http://www.debates.org/index.php?page=october-3-2000-transcript',
         'http://www.debates.org/index.php?page=october-6-1996-debate-transcript',
         'http://www.debates.org/index.php?page=september-25-1988-debate-transcript',
         'http://www.debates.org/index.php?page=october-7-1984-debate-transcript',
         'http://www.debates.org/index.php?page=september-23-1976-debate-transcript',
         'http://www.debates.org/index.php?page=september-26-1960-debate-transcript']
```

```
In [3]: import re
        import pandas as pd


        def get_most_frequent_element(L):
            occurences = dict()
            for word in re.findall("\w+", text):
                try:
                    occurences[word.lower()] += 1
                except KeyError:
                    occurences[word.lower()] = 1

            word = max(occurences, key=occurences.get)
            n_word = occurences[word]
            return word, n_word


        df = pd.DataFrame(columns=names,
                          index=["debate char length", "war_count", "most_common_w",
                          "most_common_w_count"])

        for name, url in zip(names, urls):
            # Get the html code of the url
            source0 = requests.get(url)
            soup0 = bs.BeautifulSoup(source0.content, features='html.parser')
            text = soup0.find(id="content-sm").text

            # get the transcript length
            transcript_length = len(text.replace("\n", ""))

            # get the number of occurences of the words War, war, Wars, and wars
            n_wars = len(re.findall("\W[Ww]ars*\W", text))

            # get the most common word and its word count
            word, n_word = get_most_frequent_element(text)

            # adding created information to the dataframe
            df[name] = [transcript_length, n_wars, word, n_word]

        df
```

```
Out[3]:                     October 3, 2012: The First Obama-Romney Presidential Debate  \
        debate char length                                              94627
```

```
war_count                                                                5
most_common_w                                                          the
most_common_w_count                                                    757

                    September 26, 2008: The First McCain-Obama Presidential Debate  \
debate char length                                                  182422
war_count                                                               48
most_common_w                                                          the
most_common_w_count                                                   1470

                    October 13, 2004: The Third Bush-Kerry Presidential Debate  \
debate char length                                                   85259
war_count                                                               14
most_common_w                                                          the
most_common_w_count                                                    759

                    October 3, 2000: The First Gore-Bush Presidential Debate  \
debate char length                                                   91066
war_count                                                               11
most_common_w                                                          the
most_common_w_count                                                    919

                    October 6, 1996: The First Clinton-Dole Presidential Debate  \
debate char length                                                   93090
war_count                                                               15
most_common_w                                                          the
most_common_w_count                                                    876

                    September 25, 1988: The First Bush-Dukakis Presidential Debate  \
debate char length                                                   87494
war_count                                                               14
most_common_w                                                          the
most_common_w_count                                                    804

                    October 7, 1984: The First Reagan-Mondale Presidential Debate  \
debate char length                                                   86687
war_count                                                                3
most_common_w                                                          the
most_common_w_count                                                    867

                    September 23, 1976: The First Carter-Ford Presidential Debate  \
debate char length                                                   80737
war_count                                                                7
most_common_w                                                          the
most_common_w_count                                                    857

                    September 26, 1960: The First Kennedy-Nixon Presidential Debate
debate char length                                                   60937
```

```
war_count                                                               3
most_common_w                                                         the
most_common_w_count                                                   779
```

1.2 2. Download and read in specific line from many data sets

Scrape the first 27 data sets from this URL http://people.sc.fsu.edu/~jburkardt/datasets/regression/ (i.e.x01.txt - x27.txt). Then, save the 5th line in each data set, this should be the name of the data set author (get rid of the # symbol, the white spaces and the comma at the end).

Count how many times (with a Python function) each author is the reference for one of the 27 data sets. Showcase your results, sorted, with the most common author name first and how many times he appeared in data sets. Use a Pandas DataFrame to show your results, see example. Print your final output result.

Example output of the answer for Question 2:

```
In [4]: url_root = "http://people.sc.fsu.edu/~jburkardt/datasets/regression/"
        source = requests.get(url_root)
        soup = bs.BeautifulSoup(source.content, features='html.parser')

In [5]: first_urls = [item.get("href") for item in soup.findAll("a")][6:33]

        df = pd.DataFrame(columns=["Authors", "Counts"])
        df = df.set_index("Authors")
        authors = []
        for url in first_urls:
            source_u = requests.get(url_root + url)
            soup_u = bs.BeautifulSoup(source_u.content, features='html.parser')
            author = soup_u.text.split("\n")[4].replace(",", "").replace("#    ", "")
            authors.append(author)
            try:
                df.loc[author] += 1
            except KeyError:
                df.loc[author] = 1

        df = df.sort_values("Counts", ascending=False)

In [6]: authors

Out[6]: ['Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'R J Freund and P D Minton',
         'D G Kleinbaum and L L Kupper',
         'Helmut Spaeth',
         'D G Kleinbaum and L L Kupper',
         'K A Brownlee',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'S Chatterjee and B Price',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'R J Freund and P D Minton',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'Helmut Spaeth',
         'S Chatterjee B Price',
         'S Chatterjee B Price',
         'S Chatterjee B Price',
         'S C Narula J F Wellington',
         'S C Narula J F Wellington']
```

```
In [7]: df
```

```
Out[7]:                              Counts
        Authors
        Helmut Spaeth                    16
        S Chatterjee B Price              3
        R J Freund and P D Minton         2
        D G Kleinbaum and L L Kupper      2
        S C Narula J F Wellington         2
        K A Brownlee                      1
        S Chatterjee and B Price          1
```