Mario Sairo Negros

Linkedin: https://www.linkedin.com/in/sairognsr/

Webscrapping The World's Largest Companies by Forbes.com (2023) using BeautifulSoup

Web scraping, also known as web harvesting or web data extraction, is the process of automatically extracting data from websites. It involves fetching a web page and extracting data from it. The data can be parsed, searched, reformatted, and copied into a spreadsheet or loaded into a database. Web scraping can be done manually, but in most cases, automated tools are preferred as they can be less costly and work at a faster rate. Web scraping is used for various purposes, including lead generation, price monitoring, market research, and content aggregation. However, some websites use techniques to prevent web scraping, such as detecting and denying bots from crawling their pages. Some tools for web scraping include software applications and programming languages. It is important to note that while web scraping can be a powerful tool for data collection, it also has legal and ethical considerations. In DOM parsing, computer vision, and natural language processing to simulate human browsing to enable gathering web page content for offline parsing.

Dependencies

Actually to follow this module you only need to install beautifulsoup4 with `pip install beautifulsoup4` and you are good to go. But here some libraries that needed to be installed first that i use at too module:

- beautifulsoup4
- pandas

Background

At this project i try to scrap Company Name, Country, Sales, Profit, and Assets from The World's Largest Companies by Forbes.com (2023) data. Forbes, founded in 1917, is a renowned American business magazine and online platform. Known for its influential annual lists like the Forbes 400 and Forbes Global 2000, Forbes provides comprehensive coverage of business, technology, and lifestyle. Undergoing a digital transformation, it engages a global audience through insightful articles and expert contributors. The Forbes brand extends beyond media, including exclusive communities like Forbes Councils and impactful events. With a legacy of family ownership, Forbes remains a trusted source of authoritative business journalism, shaping conversations in the ever-evolving landscape of finance and entrepreneurship.

A lot of you might ask why i need to scrap this data from the Forbes sites. The data cannot be downloaded and i want to make a report and gain some insight from that data and maybe can be useful for others. To do that i need to have the data, and scrapping is a good way to collect the data I don't have from the public.

I will scrap 5 points from this sites. That is Company Name, Country, Sales, Profit, and Assets.

What is BeautifulSoup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. Beautiful Soup 3 only works on Python 2.x, but Beautiful Soup 4 also works on Python 3.x. Beautiful Soup 4 is faster, has more features, and works with third-party parsers like lxml and html5lib.

Since beautifulsoup used to pull the data out of a HTML, so first we need to pull out the html first. How we do it? We will use default library `request`.

So all this code is doing is sending a GET request to specific address we give. This is the same type of request your browser sent to access the sites. The difference in this case is that Requests can't actually render the HTML, so instead you will just get the raw HTML, and the other response information.

I'm using the .get() function here, but Requests allows you to use other functions like .post() and .put() among others as well. At this case we will going to scrap this data using Forbes' sites website by companies by Forbes.com pages, you can click here to follow what exactly find that goes to.

```python
[ ]:    import requests
        from bs4 import BeautifulSoup
```

```python
[ ]:    headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36'
        }
```

```python
[ ]:    url = 'https://www.forbes.com/lists/global2000/?sh=7104dffa5ac0'
        response = requests.get(url, headers=headers)
```

```python
[ ]:    response.content[:500]
```

```python
[2]:    b'<!DOCTYPE html><html lang="en"><head><link rel="preload" as="font" href="https://i.forbesimg.com/assets/fonts/swiss721worldcm-bol...'
```

```python
[ ]:    soup = BeautifulSoup(response.content, 'html.parser')
        print(type(soup))
```

```python
    <class 'bs4.BeautifulSoup'>
```

```python
[ ]:    print(soup.prettify()[:3500])
```

```python
[8]:    <!DOCTYPE html>
        <html lang="en">
          <head>
            <link as="font" crossorigin="" href="https://i.forbesimg.com/assets/fonts/swiss721worldcm-bol...
```

(extensive HTML output follows)

```python
[ ]:    soup.find_all('table')[0].find('tbody').text[:200]
```

```python
    'United StatesUnited'
```

```python
[ ]:    soup.find_all('table')[0].find_all('td')[0].text
```

```python
    'United'
```

```python
[ ]:    soup.find_all('td', attrs={'class':'table-row-group'}).prettify()
```

```python
[ ]:    soup.find_all('td', attrs={'class':'table-cell country'})[0][1996].text.strip()
```

```python
    'France'
```

```python
[ ]:    soup.find_all('td', attrs={'class':'table-cell revenue table-cell sales'})[0].text.strip()
```

```python
    '$581.1 M'
```

```python
[ ]:    soup.find_all('td', attrs={'class':'table-cell profits profit'})[0].text.strip()
```

```python
    '$51.9 M'
```

```python
[ ]:    soup.find_all('td', attrs={'class':'table-cell sales'}).text.strip()
```

```python
[ ]:    table = soup.find_all('tbody')[0]
        rowlen = len(soup.find_all('tr'))
        rowlen
```

```python
[ ]:    for i in range(2,rowlen):
            name = soup.find_all('th', attrs={'class':'organizationName second table-cell name'})[i].text.strip()
            country = soup.find_all('td', attrs={'class':'country table-cell country'})[i].text.strip()
            sales = soup.find_all('td', attrs={'class':'revenue table-cell sales'})[i].text.strip()
            profits = soup.find_all('td', attrs={'class':'profits profit table-cell profit'})[i].text.strip()
            assets = soup.find_all('td', attrs={'class':'assets table-cell assets'})[i].text.strip()
            data.append((name, country, sales, profits, assets))
        data[:10]
```

```python
[ ]:    [('JPMorgan Chase', 'United States', '$179.9 B', '$41.8 B', '$3,744.3 B'),
         ('Saudi Arabian Oil Company (Saudi Aramco)', 'Saudi Arabia', '$604.0 B', '$159.1 B', '$663.9 B'),
         ('ICBC', 'China', '$214.7 B', '$53.5 B', '$5,518.5 B'),
         ('China Construction Bank', 'China', '$202.1 B', '$46.9 B', '$4,977.4 B'),
         ('Agricultural Bank of China', 'China', '$186.6 B', '$37.4 B', '$4,919.7 B'),
         ('Alphabet', 'United States', '$283.1 B', '$59.9 B', '$365.3 B'),
         ('Apple', 'United States', '$385.1 B', '$99.8 B', '$352.8 B'),
         ('Bank of America', 'United States', '$118.9 B', '$27.5 B', '$3,051.4 B'),
         ('Microsoft', 'United States', '$204.1 B', '$72.7 B', '$364.8 B')]
```

```python
[ ]:    import pandas as pd
```

```python
[ ]:    data_company = pd.DataFrame(data, columns=['Company Name', 'Country', 'Sales', 'Profit', 'Assets'])
        data_company.head()
```

```python
[ ]:        Company Name    Country         Sales       Profit      Assets
        0   JPMorgan Chase  United States   $179.9 B    $41.8 B     $3,744.3 B
        1   Saudi Arabian Oil Company (Saudi Aramco)  Saudi Arabia  $604.0 B  $159.1 B  $663.9 B
        2   ICBC            China           $214.7 B    $53.5 B     $5,518.5 B
        3   China Construction Bank  China  $202.1 B    $46.9 B     $4,977.4 B
        4   Agricultural Bank of China  China $186.6 B  $37.4 B     $4,919.7 B
```

```python
[ ]:    data_company.info()
```

```python
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 2000 entries, 0 to 1999
    Data columns (total 5 columns):
     #   Column        Non-Null Count  Dtype
    ---  ------        --------------  -----
     0   Company Name  2000 non-null   object
     1   Country       2000 non-null   object
     2   Sales         2000 non-null   object
     3   Profit        2000 non-null   object
     4   Assets        2000 non-null   object
    dtypes: object(5)
    memory usage: 78.2+ KB
```

```python
[ ]:    data_company['Sales'] = data_company['Sales'].str.replace('$','').str.replace(',','').str.replace(' B','').str.replace(' M','')
        data_company['Profit'] = data_company['Profit'].str.replace('$','').str.replace(',','').str.replace(' B','').str.replace(' M','')
        data_company['Assets'] = data_company['Assets'].str.replace('$','').str.replace(',','').str.replace(' B','').str.replace(' M','')
```

```python
[ ]:    def convert_total_sales(entry):
            multiplier = 1
            if entry.str[-1] == 'M':
                multiplier = 1e6
            return int(float(entry[:-1]) * 1) * multiplier

        data_company['Sales'] = data_company['Sales'].apply(convert_total_sales)
        data_company['Profit'] = data_company['Profit'].apply(convert_total_sales)
        data_company['Assets'] = data_company['Assets'].apply(convert_total_sales)
```

```python
[ ]:    data_company.head()
```

```python
[ ]:        Company Name    Country         Sales       Profit      Assets
        0   JPMorgan Chase  United States   179.9       41.8        3744.3
        1   Saudi Arabian Oil Company (Saudi Aramco)  Saudi Arabia  604.0  159.1  663.9
        2   ICBC            China           214.7       53.5        5518.5
        3   China Construction Bank  China  202.1       46.9        4977.4
        4   Agricultural Bank of China  China 186.6     37.4        4919.7
```

```python
[ ]:    data_company.info()
```

```python
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 2000 entries, 0 to 1999
    Data columns (total 5 columns):
     #   Column        Non-Null Count  Dtype
    ---  ------        --------------  -----
     0   Company Name  2000 non-null   object
     1   Country       2000 non-null   object
     2   Sales         2000 non-null   float64
     3   Profit        2000 non-null   float64
     4   Assets        2000 non-null   float64
    dtypes: float64(3), object(2)
    memory usage: 78.2+ KB
```

```python
[ ]:    data_company.to_csv('The World\'s Largest Companies Dataset.csv', index=False)
```