Web Scraping Tables using Pandas

Estimated Effort: 5 mins

The Pandas library in Python contains a function `read_html()` that can be used to extract tabular information from any web page.

Consider the following example:

Let us assume we want to extract the list of the largest banks in the world by market capitalization, from the following link:

```
URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'
```

We may use `pandas.read_html()` function in python to extract all the tables in the web page directly.

A snapshot of the webpage is shown below.

We can see that the required table is the first one in the web page.

> Note: This is a live web page and it may get updated over time. The image shown above has been captured in November 2023. The process of data extraction remains the same.

We may execute the following lines of code to extract the required table from the web page.

```
import pandas as pd
URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'
tables = pd.read_html(URL)
df = tables[0]
print(df)
```

This will extract the required table as a dataframe `df`. The output of the print statement would look as shown below.

	Rank	Bank name	Market cap(US$ billion)
0	1	JPMorgan Chase	419.25
1	2	Bank of America	231.52
2	3	Industrial and Commercial Bank of China	194.56
3	4	Agricultural Bank of China	160.68
4	5	HDFC Bank	157.91
5	6	Wells Fargo	155.87
6	7	HSBC Holdings PLC	148.90
7	8	Morgan Stanley	140.83
8	9	China Construction Bank	139.82
9	10	Bank of China	136.81

Although convenient, this method comes with its own set of limitations.
Firstly, web pages may have content saved in them as tables but they may not appear as tables on the web page.

For instance, consider the following URL showing the list of countries by GDP (nominal).

```
URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
```

The images on the web page are also saved in tabular format. A snapshot of the web page is shared below.

Secondly, the contents of the tables in the web pages may contain elements such as hyperlink text and other denoters, which are also scraped directly using the pandas method. This may lead to a requirement of further cleaning of data.
A closer look at table 3 in the image shown above indicates that there are many hyperlink texts which are also going to be treated as information by the pandas function.

We can extract the table using the code shown below.

```
import pandas as pd
URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
tables = pd.read_html(URL)
df = tables[2] # the required table will have index 2
print(df)
```

The output of the print statement is shown below.

	Country/Territory	UN region	IMF[1][13]		World Bank[14]		United Nations[15]	
	Country/Territory	UN region	Forecast	Year	Estimate	Year	Estimate	Ye
0	World	—	104476432	2023	100562011	2022	96698005	20
1	United States	Americas	26949643	2023	25462700	2022	23315081	20
2	China	Asia	17700899	[n 1]2023	17963171	[n 3]2022	17734131	[n 1]20
3	Germany	Europe	4429838	2023	4072192	2022	4259935	20
4	Japan	Asia	4230862	2023	4231141	2022	4940878	20
..	...	...	...	...	...	...	...	.
209	Palau	Oceania	267	2023	—	—	218	20
210	Kiribati	Oceania	246	2023	223	2022	227	20
211	Nauru	Oceania	150	2023	151	2022	155	20
212	Montserrat	Americas	—	—	—	—	72	20
213	Tuvalu	Oceania	63	2023	60	2022	60	20

Note that the hyperlink texts have also been retained in the code output.

It is further prudent to point out, that this method exclusively operates only on tabular data extraction. BeautifulSoup library still remains the default method of extracting any kind of information from web pages.

Author(s)

Abhishek Gagneja