zhuanlan.zhihu.com

天秀！Pandas还能用来写爬虫?

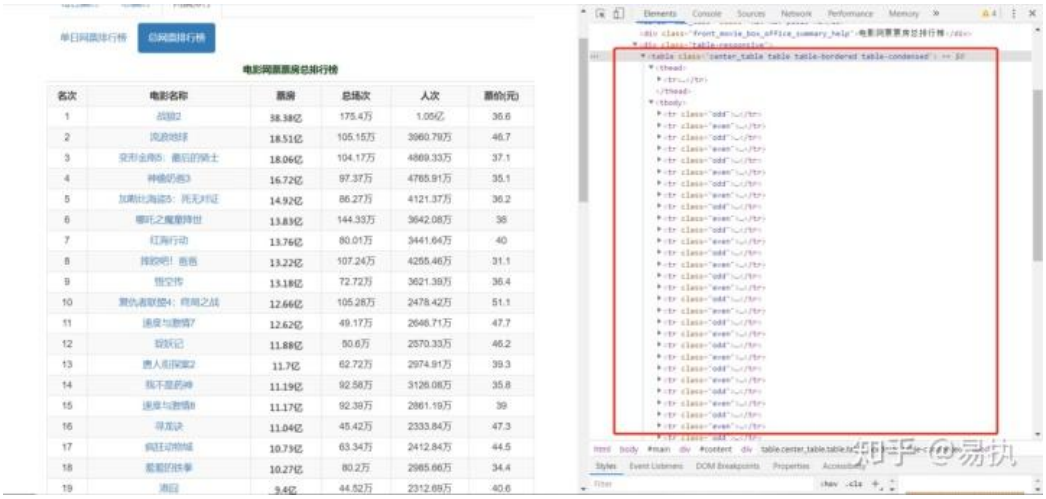
7-8 minutes

谈及Pandas的read.xxx系列的函数，大家的第一反应会想到比较常用的pd.read_csv()和pd.read_excel()，大多数人估计没用过pd.read_html()这个函数。

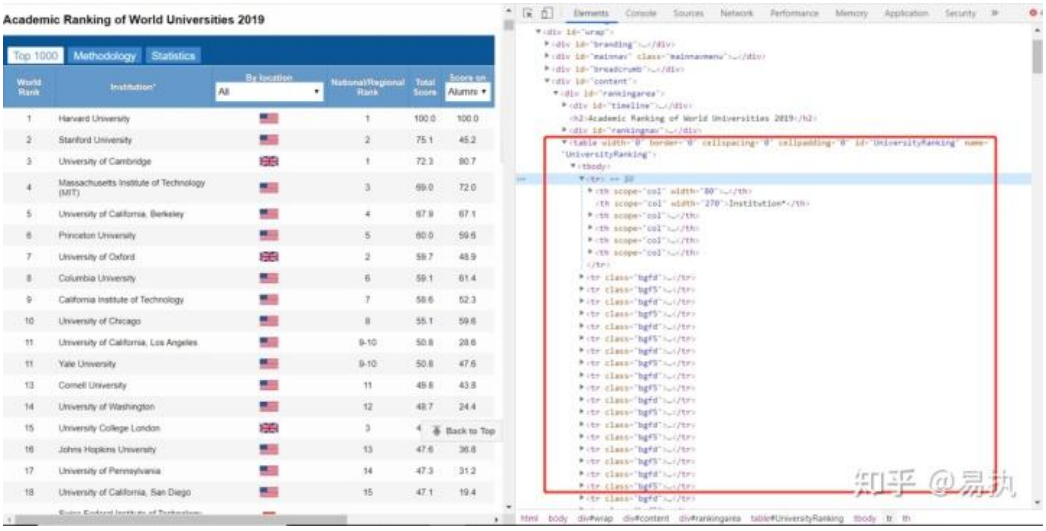
虽然它低调，但功能非常强大，用于抓取**Table表格型数据**时，简直是个神器。下面来详细介绍一下。

大家逛网页时，经常会看到这样一些数据表格，比如：

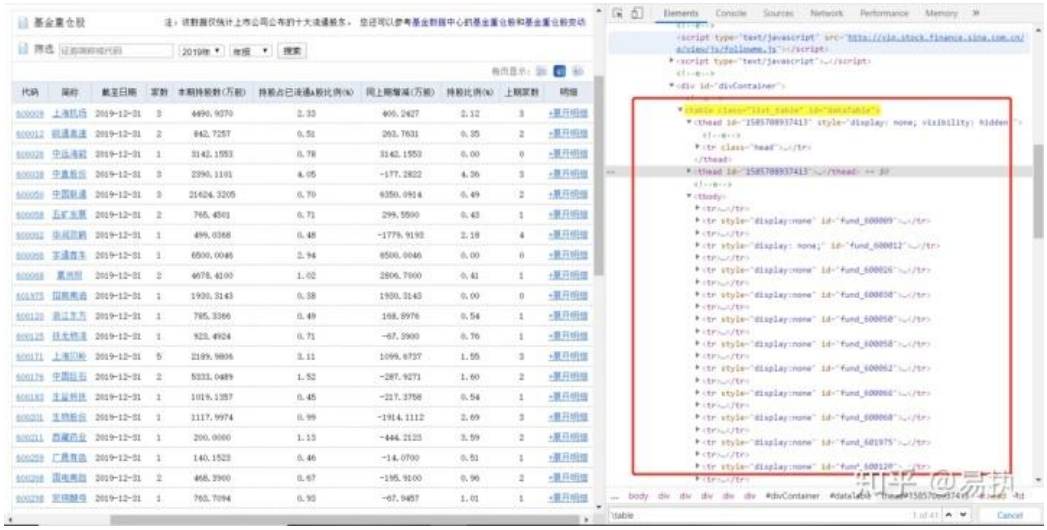
电影票房数据



世界大学排行榜数据



财经数据



如果查看一下网页的HTML结构（Chrome浏览器F12），会发现它们有个共同的特点，不仅是表格，还是以**Table结构**展示的表格数据，大致的网页结构如下

```
<table class="..." id="...">
      <thead>
      <tr>
      <th>...</th>
      </tr>
      </thead>
      <tbody>
         <tr>
             <td>...</td>
         </tr>
         <tr>...</tr>
         <tr>...</tr>
         ...
         <tr>...</tr>
         <tr>...</tr>
     </tbody>
</table>
```

针对网页结构类似的表格类型数据，`pd.read_html()`就派上了大用场了，它可以将网页上的表格都抓取下来，并以`DataFrame`的形式装在一个列表中返回。具体是这么个流程：

先介绍一下read_html的一些主要的参数

read_html

- io： str or file-like

  接收网址、文件、字符串。网址不接受https，尝试去掉s后爬去

- header：int or list-like or None

  指定列标题所在的行

- attrs：dict or None, optional

  传递一个字典，用其中的属性筛选出特定的表格

- parse_dates：bool

  解析日期

接下来以爬取新浪财经的基金重仓股为例演示一下，URL为：

http://vip.stock.finance.sina.com.cn/q/go.php/vComStockHold/kind/jjzc/index.phtml?p=1

这部分有6页，点击不同的页数可以发现，请求URL主要是p参数在变动，p=n代表了第n页，所以一个for循环就可以遍历所有网址啦。URL的变动规律了解之后，就可以愉快的爬数据了，上代码

```
import pandas as pd
df = pd.DataFrame()
for i in range(6):
    url =
'http://vip.stock.finance.sina.com.cn/q/go.php/vComStockHold/kind/jjzc/index.phtml?
p={page}'.format(page=i+1)
    df = pd.concat([df,pd.read_html(url)[0]])
    print("第{page}页完成~".format(page=i+1))
df.to_csv('./data.csv', encoding='utf-8', index=0)
```

	代码	简称	截至日期	家数	本期持股数(万股)	持股占已流通A股比例(%)	同上期增减(万股)
1	600009	上海机场	2019-12-31	3	4490.937	2.33	400.2427
2	600012	皖通高速	2019-12-31	2	842.7257	0.51	263.7631
3	600026	中远海能	2019-12-31	1	3142.1553	0.78	3142.1553
4	600038	中直股份	2019-12-31	3	2390.1101	4.05	-177.2822
5	600050	中国联通	2019-12-31	3	21624.3205	0.7	6350.0914
6	600058	五矿发展	2019-12-31	2	765.4501	0.71	299.55
7	600062	华润双鹤	2019-12-31	1	499.0368	0.48	-1779.9193
8	600066	宇通客车	2019-12-31	1	6500.0046	2.94	6500.0046
9	600068	葛洲坝	2019-12-31	2	4678.41	1.02	2806.7
10	601975	招商南油	2019-12-31	1	1930.3143	0.38	1930.3143
11	600120	浙江东方	2019-12-31	1	785.3366	0.49	168.8976
12	600125	铁龙物流	2019-12-31	1	923.4924	0.71	-67.39
13	600171	上海贝岭	2019-12-31	5	2189.9806	3.11	1099.6737
14	600176	中国巨石	2019-12-31	2	5333.0489	1.52	-287.9271
15	600183	生益科技	2019-12-31	1	1019.1357	0.45	-217.3758
16	600201	生物股份	2019-12-31	1	1117.9974	0.99	-1914.1112
17	600211	西藏药业	2019-12-31	1	200.0	1.13	-444.2123
18	600259	广晟有色	2019-12-31	1	140.1523	0.46	-14.07
19	600268	国电南自	2019-12-31	2	468.39	0.67	-195.91
20	600298	安琪酵母	2019-12-31	1	763.7094	0.93	-67.9457
21	600309	万华化学	2019-12-31	1	3453.17	1.1	-1408.1692
22	600316	洪都航空	2019-12-31	1	437.9005	2.61	-107.64
23	600317	营口港	2019-12-31	1	2015.9443	0.31	-90.03

整个过程不需要用到正则表达式或者xpath等工具，短短的几行代码就可以将数据嗖嗖地爬下来了，是不是超级无敌方便？

日后在爬一些小型数据时，只要遇到这种**Table类型的表格**，就可以直接祭出read_html这个神器啦，别人还在琢磨正则、xpath怎么写的时候，你已经把数据爬完了，想想就很舒服！

相关文章：

1. 提高数据的颜值！一起看看Pandas中的那些Style
2. Pandas数据处理三板斧——map、apply、applymap详解
3. Pandas数据分析——超好用的Groupby详解
4. Pandas数据分析——Merge数据拼接图文详解
5. Pandas数据处理——玩转时间序列数据
6. Pandas数据处理——盘点那些常用的函数（上）
7. Pandas数据处理——盘点那些常用的函数（下）
8. 提速百倍的Pandas性能优化方法，让你的Pandas飞起来！

原创不易，如果觉得有点用，希望可以**点个赞**，拜谢各位老铁！