

Data Mining and Accounting Analytics -Data Preprocessing

1

Dr. Yi Long (Neal)

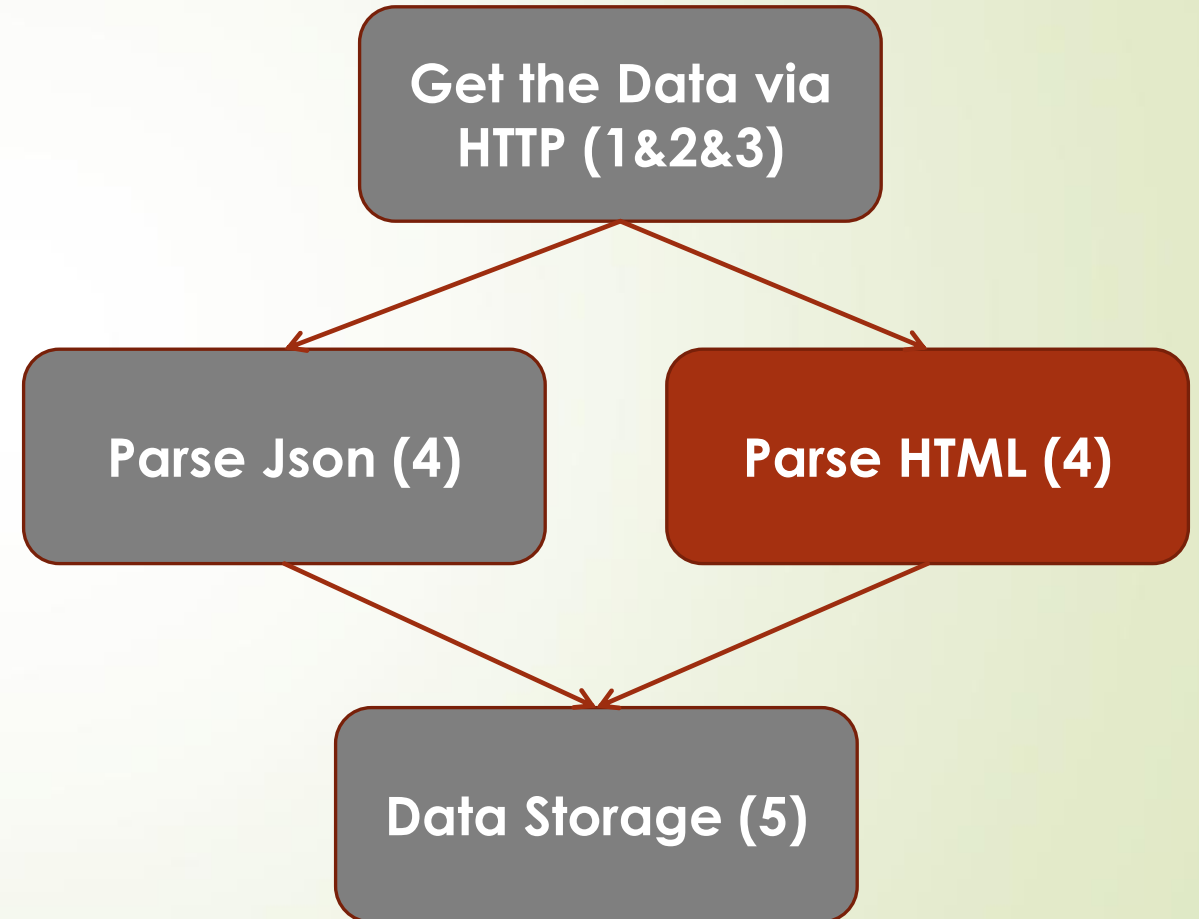
Most contents (text or images) of course slides are from the following textbook
Provost, Foster, and Tom Fawcett. Data Science for Business: What you need to
know about data mining and data-analytic thinking. " O'Reilly Media, Inc.", 2013

Outline

- Data Collection(HTML)
- Data Storage
- Lab Quiz

Web Scraping – Process

1. Send well-prepared HTTP requests to the desired webpage
2. Receive response from webpage server
3. Check the response
4. **Parse the webpage into structured data if necessary (HTML)**
5. Store the raw results/webpage



- ✓ HTML stands for **H**yper **T**ext **M**arkup **L**anguage



```
1 <!DOCTYPE html>
2 <html lang="zh-hans">
3 <head profile="http://www.w3.org/1999/xhtml/vocab">
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5   <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
6   <link rel="shortcut icon" href="http://www.cuhk.edu.cn/sites/default/files/l.png" type="image/png" />
7   <title>首页 | 香港中文大学（深圳）</title>
8   <meta charset="utf-8">
9   <meta http-equiv="X-UA-Compatible" content="IE=edge">
10  <meta name="viewport" content="width=device-width, initial-scale=1">
11  <!--[if lt IE 9]>
12    <script src="/sites/all/themes/cuhk/js/html5shiv.min.js"></script>
13    <script src="/sites/all/themes/cuhk/js/respond.min.js"></script>
14    <link href="/sites/all/themes/cuhk/css/ie8base.css" rel="stylesheet"/>
15    <link href="/sites/all/themes/cuhk/css/ie8index.css" rel="stylesheet"/>
16  <![endif]-->
17
18  <script type="text/javascript">
19    NAV_DATA = [{"mlid": "1558", "plid": "687", "hidden": "0", "language": "zh-hans", "link_title": "\u0938\u
```

Introduction to HTML (1)

- ▶ HTML describes the structure/display of Web pages using markup

- ✓ HTML elements are the building blocks of HTML pages
- ✓ HTML elements are represented by **tags**
- ✓ Each tag has a **tag name**, **attributes** with **values**, **text** and others

- ▶ HTML tags are element names surrounded by angle brackets:

<tagname id='123'>content goes here...</tagname>

- ✓ HTML tags normally **come in pairs** like <p> and </p>
- ✓ The first tag in a pair is the **start/opening tag**, the second tag is the **end/closing tag**
- ✓ The end tag is written like the start tag, but with a **forward slash** before the tag name
- ✓ **Tag name** is *tagname*
- ✓ **Values** for **attribute** 'id' is '123', **Text** is "content goes here.."

Introduction to HTML (2)

- The browser can render the content of a page based on its HTML content
 - ✓ HTML tags are predefined with display settings: `<table>` `<h1>` `<title>`
 - ✓ <https://www.w3schools.com/html/tryit.asp>
 - ✓ <https://htmlformatter.com/>

```
<html>
<body>
<p>每个表格由 table 标签开始。</p>
<p>每个表格行由 tr 标签开始。</p>
<p>每个表格数据由 td 标签开始。</p>
<h4>一行三列: </h4>
<table name="1" border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>
<h4>两行三列: </h4>
<table name="2" border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>
```



每个表格由 table 标签开始。

每个表格行由 tr 标签开始。

每个表格数据由 td 标签开始。

一行三列：

100	200	300
-----	-----	-----

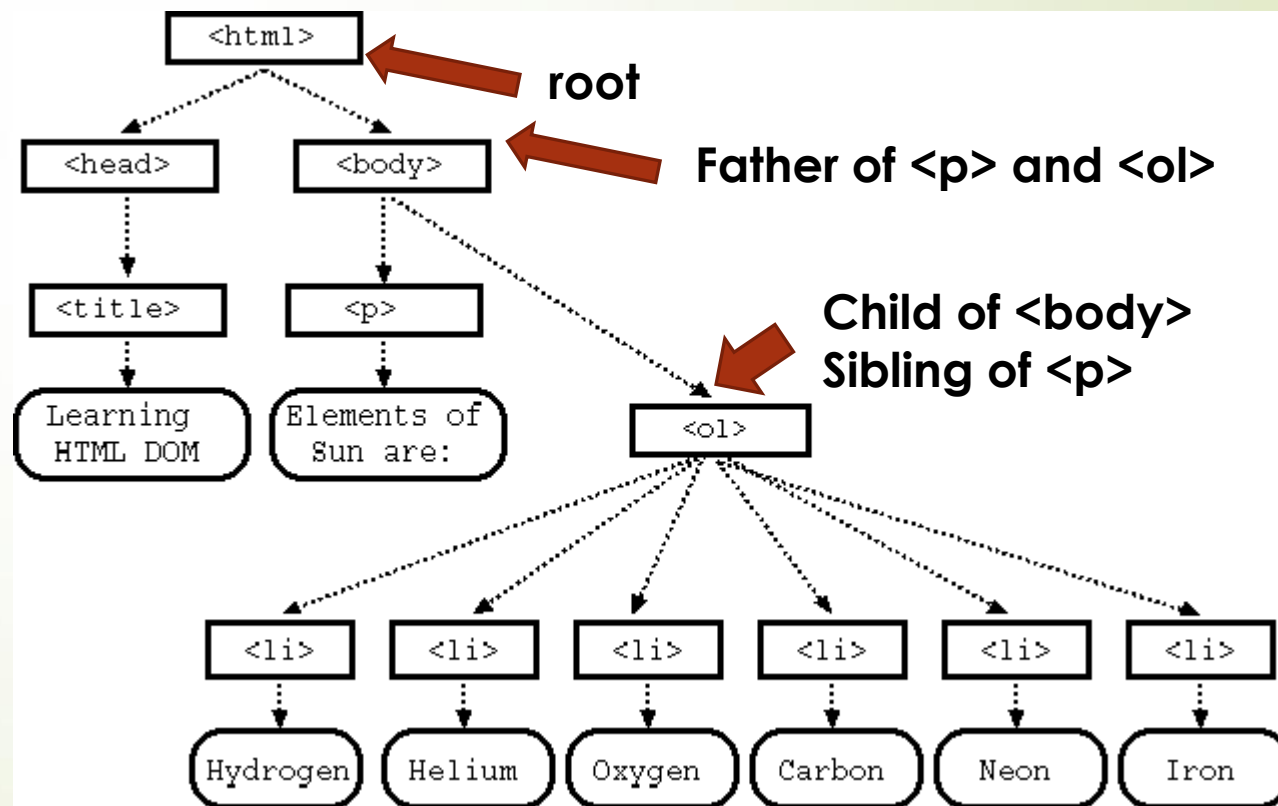
两行三列：

100	200	300
400	500	600

HTML DOM Tree

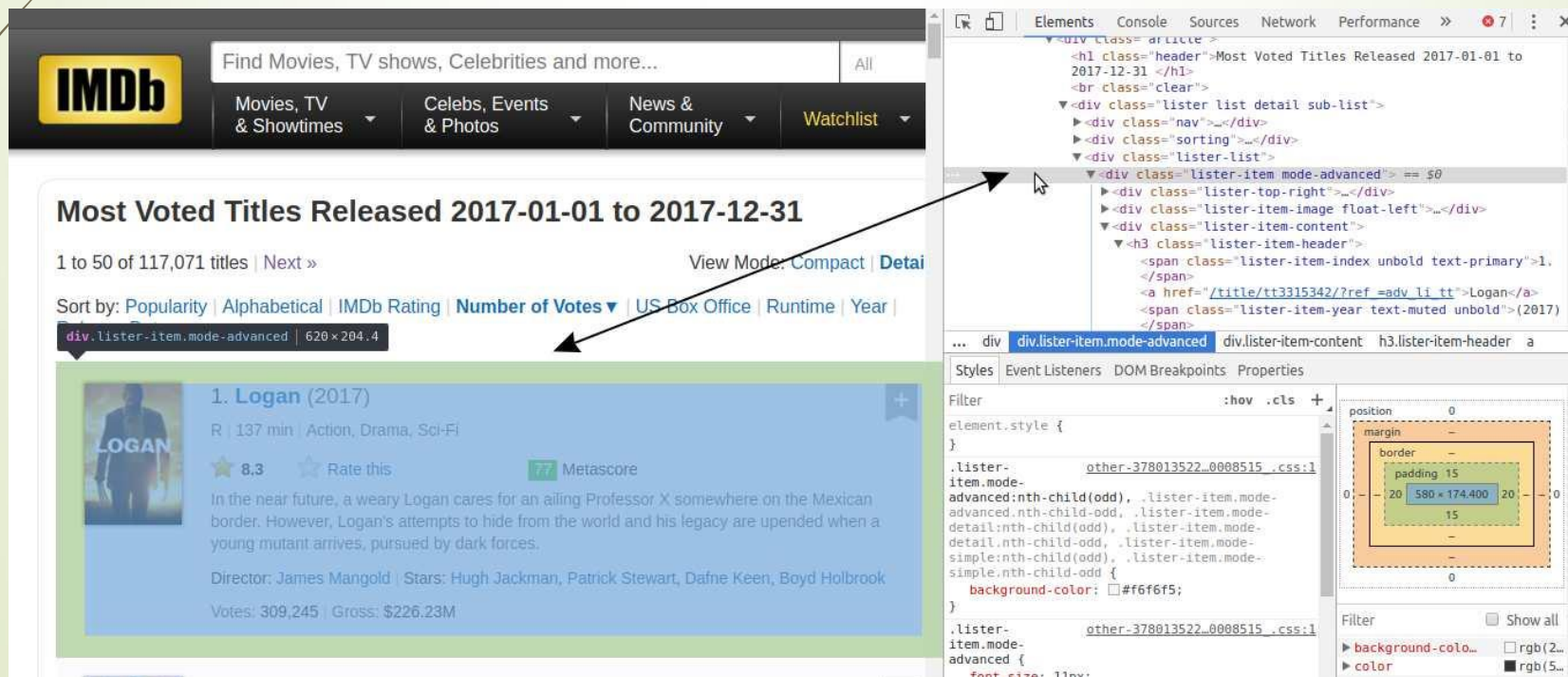
- HTML can be represented as a tree in Document Object Model (DOM)
 - ✓ Each tag can have multiple children tags

```
<!doctype html>
<html>
<head>
  <title>Learning HTML DOM</title>
</head>
<body>
<p> Elements of Sun are: </p>
<ol>
  <li>Hydrogen </li>
  <li>Helium </li>
  <li>Oxygen </li>
  <li>Carbon </li>
  <li>Neon </li>
  <li>Iron </li>
</ol>
</body>
</html>
```



HTML Parser (Beautifulsoup)

- We can easily get the content in different tags and their structures
 - ✓ HTML Parser: beautifulsoup (easy)、lxml (fast)
 - ✓ Package **Beautifulsoup** can parse and build DOM for html



Steps for Parsing HTML

- Load the HTML into parser :
 - BeautifulSoup(html_string,"html.parser")
- Locate the element in html
- Get the desired information from the located element

Locate Element in HTML

Locate the element in html

- ✓ By unique property/combination of properties
 - ✓ `tag.find_all()/find()` can search children tags under the given tag by name and attribute(/values of children tags:
 - ✓ `father.find(child_name, attrs={'key1':'val1','key2':'val2'})`
 - ✓ `father.find(child_name, string = "key word")`
- ✓ By the relative location to a located element, i.e., child, sibling, parent
 - ✓ **Children:** `tag.children` (only consider a tag's **direct** children)
 - ✓ **Descendants:** direct children, the children of its direct children...
 - ✓ **Sibling:** `tag.next_sibling`, `previous_sibling`, `find_next_sibling()` ...
 - ✓ **Parent:** `tag.parent`, `parents`, `find_parent()`,

```
7 <table id="1" border="1">
8 <tr>
9   <td>100</td>
10  <td>200</td>
11  <td>300</td>
12 </tr>
13 </table>
14 <h4>两行三列: </h4>
15 <table id="2" border="1">
16 <tr>
17   <td>100</td>
18   <td>200</td>
19   <td>300</td>
20 </tr>
```

`table = root.find('table',attrs={'id':2})`

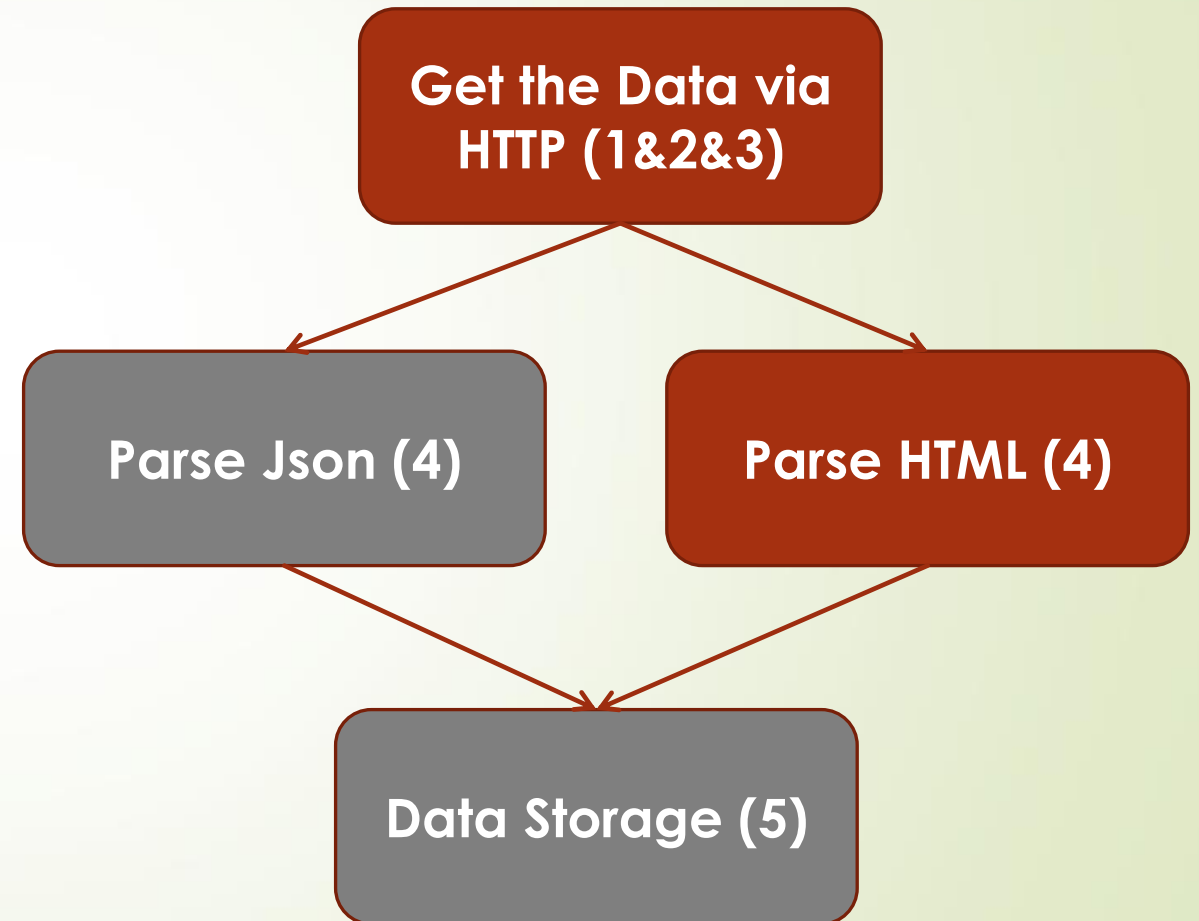
Extract the Desired Information

- Get the desired information from the located
 - ✓ Tag name: `tag.name`
 - ✓ Attribute: `tag.attrs`, `tag.attrs['attr_name']`, `tag['attr_name']`
 - ✓ Text: `tag.get_text()`, `tag.string`

`<tagname id='123'>content goes here...</tagname>`

Web Scraping – Process

1. Send well-prepared HTTP requests to the desired webpage
2. Receive response from webpage server
3. Check the response
4. Parse the webpage into structured data if necessary (HTML)
5. Store the raw results/webpage



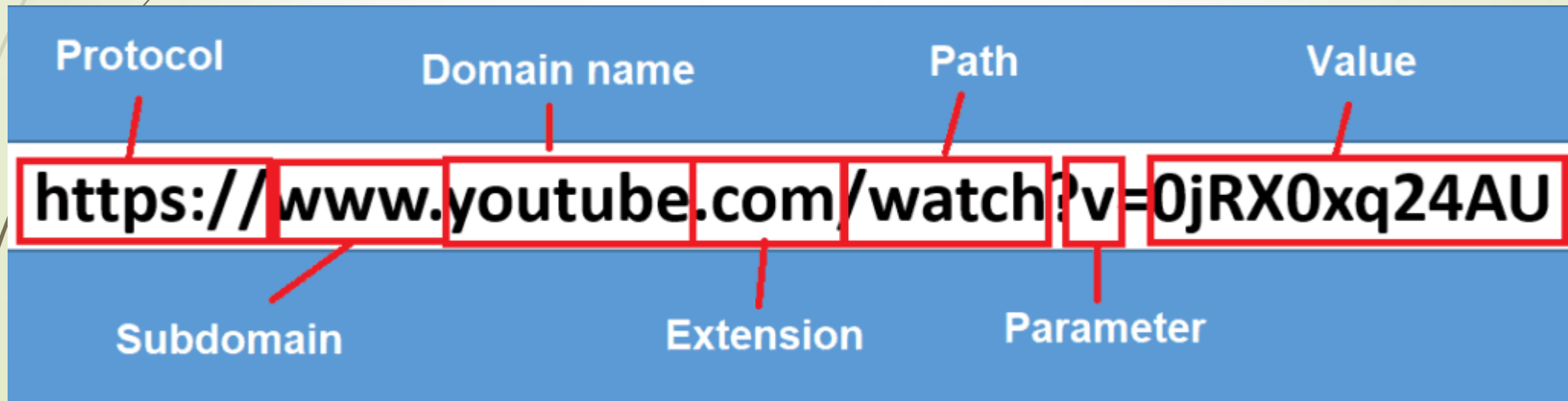
Character Encoding

- Character encoding is used to represent a repertoire of characters by some kind of encoding system.
 - Sometime referred as “character set”, “character map”, “codeset” and “code page”
 - ASCII can only handle 128 different characters , UTF-8 can handle 1,114,112 possible **characters**

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

Understand the URL

- URL with specified path/parameter values link to specified resource



Construct the URL

- Figure out how the server construct the URL to specified resource by comparison http://quotes.money.163.com/f10/gszl_600795.html#01f02
- String.format() is useful to construct correct URL

http://quotes.money.163.com/service/gszl_600795.html?type=cp

网易财经 个股行情 网易首页 > 网

行情中心 股票 新股

上证指数 2987.93 -25.12 -0.83% 4953亿

国电电力 **2.14**↑ 0.03 1.42%

(600795) 2020/02/26 15:59:39

收入构成

报告日期: 2019-06-30 [下载历史数据](#)

按产品	收入(万元)	成本(万元)	利润(万元)	毛利率	利润占比
热力产品	220,092	235,413	-15,322	-6.96%	-1.28%
其他产品	21,790	17,818	3,973	18.23%	0.33%
其他(补充)	105,327	75,277	30,050	28.53%	2.51%
煤炭产品	419,922	336,589	83,332	19.84%	6.95%
电力产品	4,833,438	3,744,598	1,088,841	22.53%	90.86%
减: 内部抵销数	-160,774	-162,212	1,438	-0.89%	0.12%
化工产品	109,244	103,159	6,085	5.57%	0.51%

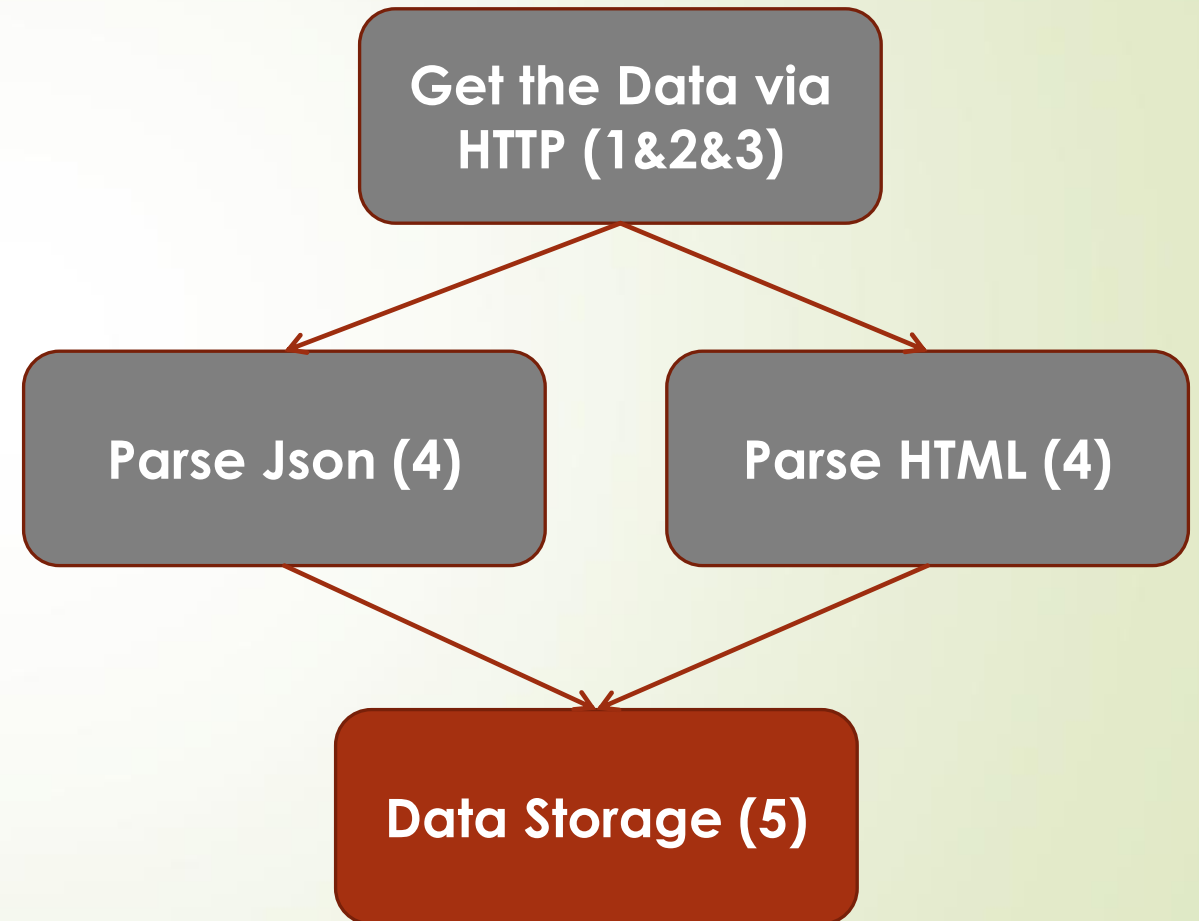
报告日期: 2019-06-30 [下载历史数据](#)

按行业	收入(万元)	成本(万元)	利润(万元)	毛利率	利润占比
-----	--------	--------	--------	-----	------

http://quotes.money.163.com/service/gszl_600795.html?type=hy

Web Scraping – Process

1. Send well-prepared HTTP requests to the desired webpage
2. Receive response from webpage server
3. Check the response
4. Parse the webpage into structured data if necessary
5. **Store the raw results/webpage**



Data Storage

- Structured data can be stored in various data structure in memory
 - ✓ Python list, tuple, set, dictionary ...
 - ✓ Pandas dataframe, Numpy ndarray ...
- How to store these data persistently and share
 - ✓ Python to python: **pickle**
 - ✓ Database: **SQLite**, MySQL, Oracle, MS SQL, Hbase, MongoDB ...
 - ✓ Text files: **txt**, **csv**, tsv
 - ✓ Data interchange format: XML(/html), **JSON**,
 - ✓ Others: xls/xlsx (**Excel**), dta(STATA), ...

Read and Write Text Files

- Users can easily write/read content to/from files
 - ✓ Open file with proper status: `f = open(file_path, 'r/w/rb/wb')`
 - ✓ Write/reader content: `content = f.read()` / `f.write(content)`
 - ✓ Can read and write by lines as well
 - ✓ Close file, this is important to save changes/ release file: `f.close()`
 - ✓ Use the “encoding” parameter to deterring the character encoding
- Use with statement to close file automatically

```
f=open("work_file",'w')  
f.write(haha)  
#other operation  
f.close()
```



```
with open("work_file",'w') as f:  
f.write(haha)  
#other operation
```

Memory VS. Disk

- Memory is usually much smaller than disk
 - ✓ Processing data in small batches (row by row) is favorable



Read File in Python

- **read()** : Reads the **entire** file in form of a string.
- **readlines()**: This reads **all lines** from the file object and returns them as a list
- **readline()** : Reads one line of the file and returns in form of a string.

```
>>> with open('dog_breeds.txt', 'r') as reader:  
>>>     # Read & print the entire file  
>>>     print(reader.read())
```

```
>>> with open('dog_breeds.txt', 'r') as reader:  
>>>     # Read and print the entire file line by line  
>>>     for line in reader:  
>>>         print(line, end='')
```

```
>>> with open('dog_breeds.txt', 'r') as reader:  
>>>     # Read and print the entire file line by line  
>>>     for line in reader:  
>>>         print(line, end='')
```



Write File in Python

Method	What It Does
<code>.write(string)</code>	This writes the string to the file.
<code>.writelines(seq)</code>	This writes the sequence to the file. No line endings are appended to each sequence item. It's up to you to add the appropriate line ending(s).

Read and Write Other Files

- It defaults to 'r' which means open for reading in text mode
- By default, 't' is included in the *mode* argument
- Can be used with combination, 'rb', 'wb', 'w+b'...

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of the file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open for updating (reading and writing)

CSV – Text File for Structured Data

- Users can store table-like data to text with CSV (Comma-separated values)
 - ✓ Store one record per line
 - ✓ Records are composed of fields/columns separated by delimiters, typically a single comma, semicolon, tab
 - ✓ Every record has the same sequence of fields
- Widely used by various platforms for data input/output
 - ✓ Excel, database system, pandas
 - ✓ R, STATA, SPSS, SAS

```
1 Date,Open,High,Low,Close,Adj Close,Volume
2 2017-01-03,20.549999,20.879998999999998,20.549999,20.73,20.047922,21701669
3 2017-01-04,20.74,20.950001,20.450001,20.85,20.1639730000000002,33155480
4 2017-01-05,20.85,21.23,20.7800010000000002,20.93,20.2413410000000002,31012563
5 2017-01-06,20.9400010000000002,21.040001,20.610001,20.639999,19.960882,23591954
6 2017-01-09,20.6,20.75,20.5300010000000002,20.66,19.980225,15095445
7 2017-01-10,20.67,20.6900010000000002,20.52,20.58,19.902857,15917148
8 2017-01-11,20.52,20.629998999999998,20.4,20.4,19.728779,16865220
```

```
1 Date Open High Low CloseAdj Close Volume
2 2017-01-03 20.549999 20.879998999999998 20.549999 20.73 20.047922 21701669
3 2017-01-04 20.74 20.950001 20.450001 20.85 20.1639730000000002 33155480
4 2017-01-05 20.85 21.23 20.7800010000000002 20.93 20.2413410000000002 31012563
5 2017-01-06 20.9400010000000002 21.040001 20.610001 20.639999 19.960882 23591954
6 2017-01-09 20.6 20.75 20.5300010000000002 20.66 19.980225 15095445
7 2017-01-10 20.67 20.6900010000000002 20.52 20.58 19.902857 15917148
8 2017-01-11 20.52 20.629998999999998 20.4 20.4 19.728779 16865220
```

Handle CSV in Python

- CSV can be handled in rows
 - ✓ Row is a sequence of values stored in list or tuple (name,gender,birth)
 - ✓ `reader=csv.reader(file_obj)`, `writer=csv.writer(file_obj)`
 - ✓ `next(reader)` will iterate row by row , `writer.writerow(row)` will save one row to file
 - ✓ Can handle those control characters (`'\n'`, `'.'`) automatically
 - ✓ Try to use `DictReader/DictWriter` for better understanding

```
8 import csv
9 csv_data_path='./data/complex_data.csv'
10
11 with open(csv_data_path,'w',newline='') as wf:
12     writer = csv.writer(wf)
13     writer.writerow((1,"Hello \n World"))
14     writer.writerow((2,"""This, is a "complex" \n record"""))
15     writer.writerow((3,"The end"))
16
17 with open(csv_data_path,'r',newline='') as rf:
18     reader = csv.reader(rf)
19     for row in reader:
20         print(row)
```




```
['1', 'Hello \n World']
['2', 'This, is a "complex" \n record']
['3', 'The end']
```


Dump/load Pickle File

- ➡ Pickle can serialize and de-serialize a Python object structure
 - ✓ Only applicable when sharing with other **Python** application
 - ✓ Can only dump/load in one batch (handle small data)

```
8 import pickle
9 data_path = './data/dict.pkl'
10
11 # An arbitrary collection of objects supported by pickle.
12 data = {
13     'a': [1, 2.0, 3, 4+6j],
14     'b': ("character string", b"byte string"),
15     'c': {None, True, False}
16 }
17
18 print("==Data dumped to pickle==")
19 print(data)
20 with open(data_path, 'wb') as wf:
21     # Pickle the 'data' dictionary
22     pickle.dump(data, wf)
23
24 with open(data_path, 'rb') as rf:
25     # Load the 'data' dictionary back from pickle
26     data_new = pickle.load(rf)
27 print("\n==Data loaded from pickle==")
28 print(data_new)
```



```
==Data dumped to pickle==
{'a': [1, 2.0, 3, (4+6j)], 'b': ('character string', b'byte string'),
 'c': {False, True, None}}
```

```
==Data loaded from pickle==
{'a': [1, 2.0, 3, (4+6j)], 'b': ('character string', b'byte string'),
 'c': {False, True, None}}
```

Dump/load JSON File

- ▶ The JSON package has the “dump” function which directly writes the dictionary to **a file** in the form of JSON
 - ▶ Share the data with other programming languages
 - ▶ Only following data types are allowed

PYTHON OBJECT	JSON OBJECT
dict	object
list, tuple	array
str	string
int, long, float	numbers
True	true
False	false
None	null

Database Systems

- ▶ Can store, manage and share large-scale data
- ▶ Two main types of databases: SQL vs. NoSQL
 - ✓ Relational databases(SQL) : MySQL, Oracle, MS SQL, **SQLite** ...
 - ✓ Non-relational databases (NoSQL): Neo4J, MongoDB, Redis, HBase
- ▶ Relational databases store and manage data in tables
 - ✓ Similar to dataframe, spreadsheet (rows=records, columns = fields)
 - ✓ Atomicity, Consistency, Isolation, Durability (ACID) are usually guaranteed
 - ✓ Users can insert, delete, update, search the data efficiently

Learn SQL online: <https://www.w3schools.com/sql/>

Save/load with Pandas

- The pandas I/O API is a set of top level reader functions accessed
 - pandas.**read_csv()** that generally return a pandas object.
 - The corresponding writer functions are object methods that are accessed like DataFrame.**to_csv()**.

https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html

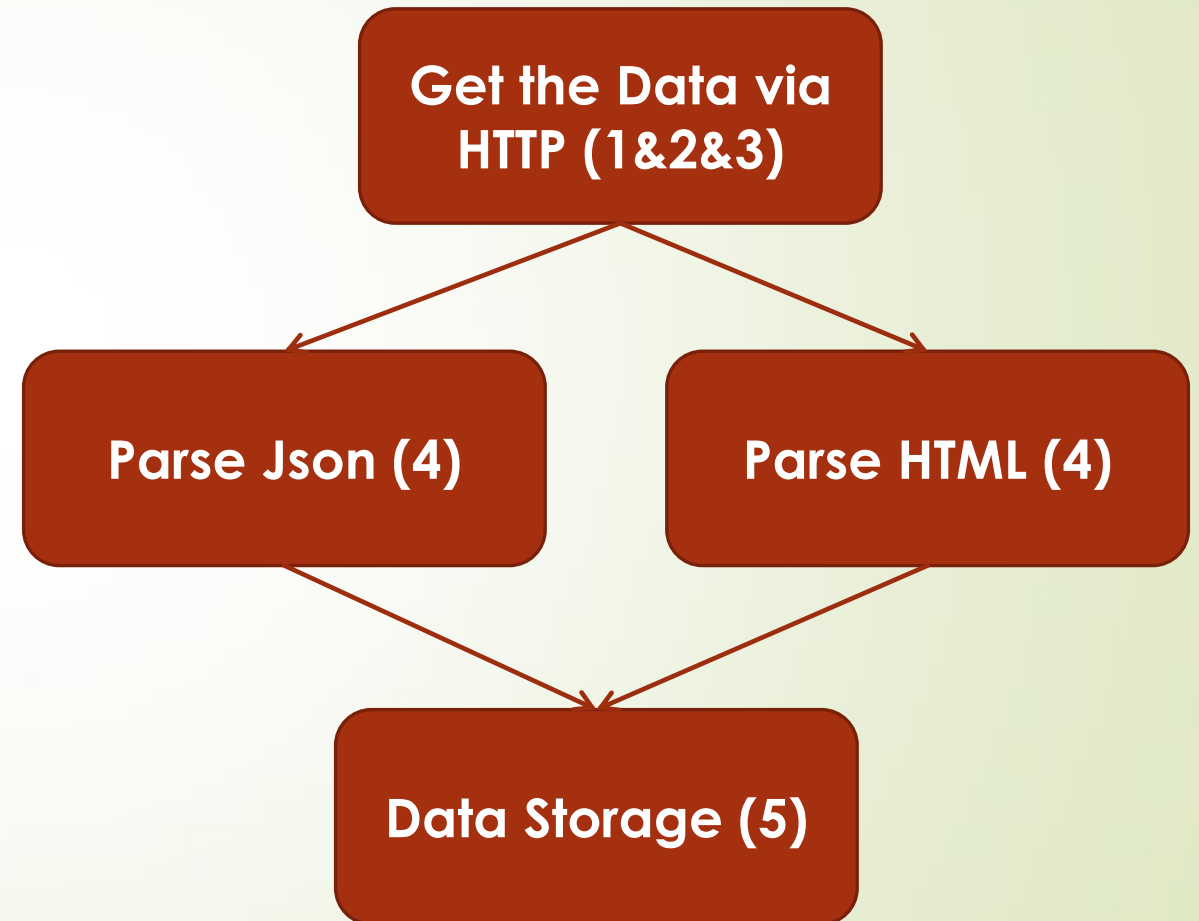
Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	Fixed-Width Text File	read_fwf	
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	Local clipboard	read_clipboard	to_clipboard
	MS Excel	read_excel	to_excel
binary	OpenDocument	read_excel	
binary	HDF5 Format	read_hdf	to_hdf
binary	Feather Format	read_feather	to_feather
binary	Parquet Format	read_parquet	to_parquet
binary	ORC Format	read_orc	
binary	Msgpack	read_msgpack	to_msgpack
binary	Stata	read_stata	to_stata
binary	SAS	read_sas	

Data Storage in Python (Summary)

- Text: unstructured data : allows mini-batch
- CSV: allows mini-batch
- Database : allows mini-batch
- Excel (pandas)
- JSON
- Pickle

Web Scraping – Process

1. Send well-prepared HTTP requests to the desired webpage
2. Receive response from webpage server
3. Check the response
4. Parse the webpage into structured data if necessary (HTML)
5. Store the raw results/webpage



Web Scraping in Practice

(000001) 平安银行：关于无固定期限资本债券（第二期）发行完毕的公告	
公告日期: 2020-02-26 00:00:00	
(000001) 平安银行：关于无固定期限资本债券（第二期）发行完毕的公告	
平安银行现发布关于无固定期限资本债券（第二期）发行完毕的公告	
(000001) 平安银行：2019年年度报告主要财务指标及分配预案	
公告日期: 2020-02-14 00:00:00	
(000001) 平安银行：2019年年度报告主要财务指标及分配预案	
每股收益（元）：1.54净资产收益率：11.30%分配预案为：每10股派发现金股利人民币2.18元（含税）。	
(000001) 平安银行：拟披露年报	
公告日期: 2019-12-31 00:00:00	
(000001) 平安银行：拟披露年报	
(000001) 平安银行：2019年度业绩快报公告	
公告日期: 2020-01-14 00:00:00	
(000001) 平安银行：2019年度业绩快报公告	



	A	B	C
1	股票代码	公告日期	公告标题
2	000001	2020/2/26 0:00	1) 平安银行：关于无固定期限资本债券（第二期）发行完毕的公告
3	000001	2020/2/14 0:00	1) 平安银行：2019年年度报告主要财务指标及分配预案
4	000001	2019/12/31 0:00	1) 平安银行：拟披露年报
5	000001	2020/1/14 0:00	1) 平安银行：2019年度业绩快报公告
6	000001	2020/1/7 0:00	1) 平安银行：关于董事任职资格核准的公告
7	000001	2019/12/28 0:00	1) 平安银行：董事会决议公告
8	000001	2019/12/27 0:00	1) 平安银行：关于无固定期限资本债券（第一期）发行完毕的公告
9	000001	2019/12/20 0:00	1) 平安银行：关于获准发行无固定期限资本债券的公告
10	000001	2019/11/29 0:00	1) 平安银行：董事会决议公告
11	000002	2019/12/31 0:00	2) 万科A：拟披露年报
12	000002	2020/2/20 0:00	2) 万科A：2019年面向合格投资者公开发行住房租赁专项公司债券（第一期）2020年付息公告
13	000002	2020/2/6 0:00	2) 万科A：2020年一月份销售及近期新增项目情况简报
14	000002	2020/1/4 0:00	2) 万科A：2019年十二月份销售及近期新增项目情况简报
15	000002	2019/12/20 0:00	2) 万科A：关于股东权益变动的提示性公告
16	000002	2019/12/18 0:00	2) 万科A：关于股东A股股份解除质押的公告
17	000002	2019/12/14 0:00	2) 万科A：关于股东A股股份解除质押的公告
18	000002	2019/12/13 0:00	2) 万科A：境内同步披露公告
19	000002	2019/12/7 0:00	2) 万科A：关于股东A股股份解除质押的公告

http://vip.stock.finance.sina.com.cn/corp/view/vCB_AllMemordDetail.php?stockid=000001

Resources for Web Crawler

- **Requests** <https://requests.readthedocs.io/en/master/>
- **JSON** https://www.w3schools.com/python/python_json.asp
- **BS4** <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- <http://httpbin.org/> for checking your request
- **Chrome** for monitoring the communication
- Hands-on Tutorial: <https://www.dataquest.io/blog/web-scraping-beautifulsoup/>

Outline

- Data Collection(HTML)
- Data Storage
- Lab Quiz

Lab Quiz-1

- **Deadline:** 12:59 p.m., ~~Feb. 21, 2020~~ **Feb. 28, 2020**
- Two question accounting for 2% of overall score
- **Upload** the **answer worksheet** and the accomplished **Python files** to the **Blackboard**
- You may submit **unlimited times** but only the **LAST** submission will be considered
- Note: **MUST attach ALL** the required files in every submission/resubmission, otherwise other files will be missing.

Lab Quiz-2

- **Deadline:** 12:59 p.m., Feb. 28, 2020
- Two questions accounting for 3% of overall score
- **Upload** the **answer worksheet** and the accomplished **Python files** to the **Blackboard**
- You may submit **unlimited times** but only the **LAST** submission will be considered
- Note: **MUST attach ALL** the required files in every submission/resubmission, otherwise other files will be missing.

Lab Quiz Submission

ASSIGNMENT INFORMATION

Due Date Friday, February 21, 2020 1:00 PM	Points Possible 100
---	-------------------------------

Please upload the following documents via this link:

1. Answer worksheet (.xlsx);
2. 2 Python files (.py).

Please make sure that you have uploaded ALL the required files in every submission.

ASSIGNMENT SUBMISSION

Text Submission




Write Submission

Attach Files

Browse My Computer

Browse Course

Attached files

File Name	Link Title	
 answer sheet.xlsx	answer sheet.xlsx	Do not attach
 Q1_prime_number.py	Q1_prime_number.py	Do not attach
 Q2_odd_number_mean.py	Q2_odd_number_mean.py	Do not attach

ADD COMMENTS

Comments

When finished, make sure to click **Submit**.
Optionally, click **Save as Draft** to save changes and continue working later, or click **Cancel** to quit without saving changes.
You are previewing the assignment - your submission will not be saved.

Cancel

Save Draft

Submit

One More Thing

- **What Q2 in Quiz-2 changed to**
 - The organization who hold second largest percentage of SZ50_top10 firm shares ~~in total~~ **on average** is
- Rewrite your solution to Q1 in Quiz-2 with `csv.DictWriter/DictReader`
- Rewrite your solution to Q2 in Quiz-2 with Pandas