

Web Scraping and Data Tools

Macroeconomics 3: TA class #6

Andrea Pasqualini

Bocconi University

15 March 2021

Web Scrapping: What Is It?

Definition

From Wikipedia: *Web Scrapping [...] is used for extracting data from websites. [...] The term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.*

Objective: Build a dataset from potentially unstructured data stored on a remote (web) server

Tools: Computer programs that navigate, identify and organize data into a structured dataset

Web Scraping: Why?

- ▶ Limited availability of structured data
 - ▶ Structured data are records assembled by somebody (e.g., gov't agency)
 - ▶ Cost of assembling records is high (e.g., data entry, data verification, methodologies)
 - ▶ Benefit must outweigh the costs
- ▶ The Internet as a medium of information exchange
 - ▶ The Internet grew fast because it's a business opportunity (e.g., Amazon)
 - ▶ Information is available for consumption reasons
 - ▶ Assembling structured *public* datasets carries little value for stakeholders
- ▶ The Internet as a platform for user-generated content
 - ▶ Users generate massive amounts of coded information (e.g., eBay)
 - ▶ Users generate massive amounts of uncoded information (e.g., Twitter)

Catch-all reason: uncover new evidence with novel data

Relevant reading: [Edelman, 2012]

Web Scraping: Examples (from Edelman, 2012)

Microeconomics

- ▶ **Patrick Bajari and Ali Hortacsu.** 2003. “The Winner’s Curse, Reserve Prices, and Endogenous Entry: Empirical Insights from eBay Auctions.” *RAND Journal of Economics* 34(2): 329–55.
- ▶ Bid data from coin sales on eBay reveal bidder behavior in auctions, including the magnitude of the winner’s curse

Macroeconomics

- ▶ **Alberto Cavallo.** 2015. “Scraped Data and Sticky Prices.” *NBER Working Paper*
- ▶ Daily price data from online supermarkets reveal price adjustment and price stickiness

Financial Economics

- ▶ **Werner Antweiler and Murray Z. Frank.** 2004. “Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards.” *Journal of Finance* 59(3): 1259–94.
- ▶ Finds that online discussions help predict market volatility; effects on stock returns are statistically significant but economically small

Web Scrapping: Which Tools and When to Use Them?

HTTP programming

(HTTP: HyperText Transfer Protocol)

- ▶ Carefully craft a willful URL
- ▶ Obtain and manage response from server
- ▶ Most popular: HTTP Application Programming Interfaces (API)

HTML parsing

(HTML: HyperText Markup Language)

- ▶ Write a program to “surf” specific HTML code
- ▶ Navigate to specific points
- ▶ Read and write info on separate data storage

Browser automation

(DOM: Document Object Model)

- ▶ Write a program to hijack your browser
- ▶ Make the browser navigate webpages, read and write info
- ▶ Useful for dynamically-generated HTML pages (e.g., Twitter)

The Internet: A Primer at Light Speed (1/3)

What happens when you write a URL in the address bar and press Enter?

1. Your browser sends a **HTTP request** for data to the destination web server
2. The web server receives the request and... serves it, sending a **HTTP response** back
3. Your browser receives the data, analyzes the metadata, and displays a payload

A **HTTP request** is a text file with the following components

- ▶ A request line
- ▶ Request header fields
- ▶ An empty line
- ▶ (Optional) A message body

A **HTTP response** is a text file with the following components

- ▶ A status line
- ▶ Response header fields
- ▶ An empty line
- ▶ (Optional) A message body

The Internet: A Primer at Light Speed (2/3)

Example HTTP request

```
1 GET / HTTP/1.1
2 Host: www.example.com
3
```

Example HTTP response

1 — Status line

2–9 — Header fields

11–18 — Message body

Status	Meaning
1xx	Information
2xx	Success
3xx	Redirection
4xx	Client error
5xx	Server error

```
1 HTTP/1.1 200 OK
2 Date: Mon, 23 May 2005 22:38:34 GMT
3 Content-Type: text/html; charset=UTF-8
4 Content-Length: 155
5 Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
6 Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
7 ETag: "3f80f-1b6-3e1cb03b"
8 Accept-Ranges: bytes
9 Connection: close
10
11 <html>
12   <head>
13     <title>An Example Page</title>
14   </head>
15   <body>
16     <p>Hello World, this is a HTML document.</p>
17   </body>
18 </html>
```

The Internet: A Primer at Light Speed (3/3)

Why do we care about HTTP requests and responses?

HTTP programming

- ▶ The response message body is the data we are after (e.g., API)
- ▶ Can be a { JSON, CSV, XLS, ... } file

HTML parsing

- ▶ The response message body is (static) HTML code
- ▶ We use a program to parse the HTML code

Browser automation

- ▶ The response message body may contain a `<script>` tag (e.g., JavaScript)
- ▶ The HTML code changes dynamically depending on circumstances
- ▶ Cannot handle this with a simple HTML parser

Luckily, Python can handle each and every of these scenarios!

Web Scrapping: The Tools in the Python Toolbox

HTTP programming

- ▶ `import requests`
- ▶ Craft custom requests
- ▶ Manage responses
- ▶ <https://requests.readthedocs.io/en/master/>

HTML parsing

- ▶ `import bs4`
- ▶ Access the HTML code in a response's message body
- ▶ Navigate the HTML document with a “Pythonic” interface
- ▶ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Browser automation

- ▶ `import selenium`
- ▶ Hijack your web browser, emulate human behavior with a browser
- ▶ Navigate the potentially-changing HTML document, trigger functions in web scripts
- ▶ <https://selenium-python.readthedocs.io/>

HTTP Programming: An Example

Go to <https://xkcd.com/2434/info.0.json> (write a HTTP request)

The message body of the HTTP response is (this is the pretty-print)

```
1 {  
2   "month": "3",  
3   "num": 2434,  
4   "link": "",  
5   "year": "2021",  
6   "news": "",  
7   "safe_title": "Vaccine Guidance",  
8   "transcript": "",  
9   "alt": "I can't wait until I'm fully vaccinated and can safely send chat  
10      messages in all caps again.",  
11   "img": "https://imgs.xkcd.com/comics/vaccine_guidance.png",  
12   "title": "Vaccine Guidance",  
13   "day": "8"  
}
```

All we need to do is

- ▶ Translate this JSON text into a Python dictionary
- ▶ Feed the dictionary into a `pandas.DataFrame`

I have permission to do this: see <https://xkcd.com/license.html>.

HTML Parsing / Browser Automation: Identifying Information

All website content is embedded into its HTML document

Objective: identify information of interest in the HTML tree

- ▶ Use *Web Developer Tools* in your browser to easily identify elements in the HTML tree
- ▶ Take notes: what HTML tags are used? How are elements *uniquely* identified?

My experience

- ▶ Tidy websites uniquely identify elements with CSS classes
- ▶ Messy websites require smart strategies (e.g., find table whose caption is [smth])
- ▶ Some web developers intentionally make seemingly messy HTML code
 - ▶ Web dev laziness, and/or
 - ▶ Sophisticated back-end design (e.g., Facebook), and/or
 - ▶ Deliberate attempt at making scraping difficult

HTML Parsing / Browser Automation: Example (1/2)

The screenshot shows a web browser displaying the XKCD comic 'Geothmetic Meandian'. The browser window has a single tab titled 'xkcd: Geothmetic Meandian' and the address bar shows 'https://xkcd.com'. The page layout includes a sidebar on the left with links to 'ARCHIVE', 'WHAT IF?', 'BLAG', 'HOW TO STORE', 'ABOUT', 'FEED', 'EMAIL', and 'TW · FB · IG'. The main content area features the 'xkcd' logo, the text 'A WEBCOMIC OF ROMANCE, SARCASM, MATH, AND LANGUAGE.', and a stick figure character with the text 'BLACK LIVES MATTER' and a 'HOW TO HELP' button. Below this is the comic title 'GEOTHMETIC MEANDIAN' and navigation buttons: '<|<', '< PREV', 'img 483.5 x 333', 'NEXT >|>', and '>|'. The comic itself is a hand-drawn illustration with mathematical formulas. The first formula is
$$F(x_1, x_2, \dots, x_n) = \left(\underbrace{\frac{x_1 + x_2 + \dots + x_n}{n}}_{\text{ARITHMETIC MEAN}}, \underbrace{\sqrt[n]{x_1 x_2 \dots x_n}}_{\text{GEOMETRIC MEAN}}, \underbrace{\frac{x_1 + x_2 + \dots + x_n}{2}}_{\text{MEDIAN MEAN}} \right)$$
 The second formula is
$$\text{GMDN}(x_1, x_2, \dots, x_n) = \underbrace{F(F(F(\dots F(x_1, x_2, \dots, x_n) \dots)))}_{\text{GEOTHMETIC MEANDIAN}}$$
 The third formula is
$$\text{GMDN}(1, 1, 2, 3, 5) \approx 2.089$$
 Below the formulas is a text box that says: 'STATS TIP: IF YOU AREN'T SURE WHETHER TO USE THE MEAN, MEDIAN, OR GEOMETRIC MEAN, JUST CALCULATE ALL THREE, THEN REPEAT UNTIL IT CONVERGES'. The right side of the image shows the browser's developer tools with the 'Inspector' tab open. The 'HTML' pane shows the document structure, with the following code highlighted:

```

```

 The 'Filter Styles' pane shows the 'image-orientation: none;' style applied to the image element. The 'Layout' pane shows the 'Flexbox' and 'Grid' properties, with a note that 'CSS Grid is not in use on this page'.

I have permission to do this: see <https://xkcd.com/license.html>.

HTML/Browser automation: Example (2/2)

Browser window showing the XKCD comic "Vaccine Guidance" (https://xkcd.com/2434/).

The comic is titled "VACCINE GUIDANCE" and is a four-panel comic strip. The panels contain the following text:

Panel 1: OUR NEW GUIDANCE: FULLY VACCINATED PEOPLE CAN GATHER PRIVATELY WITH NO MASKS OR DISTANCING, AND CAN VISIT WITH UNVACCINATED LOW-RISK PEOPLE IN ONE HOUSEHOLD. ANY QUESTIONS? (CDC logo)

Panel 2: IF MY NEIGHBORS AND I ARE ALL VACCINATED, CAN I VISIT THEM UNMASKED AND DRINK MILK DIRECTLY FROM THE JUG IN THEIR FRIDGE? I...YOU CAN VISIT, YES, AND THE JUG THING? ...NEXT QUESTION?

Panel 3: I'M FULLY VACCINATED. CAN I RIDE MY BIKE IN MY SISTER-IN-LAW'S HOUSE? IN HER HOUSE? LIKE, DOWN THE STAIRS. I GUESS YOU SHOULD AT LEAST WEAR A HELMET. EVEN IF SHE'S NOT HIGH-RISK? ANY OTHER QUESTIONS?

Panel 4: I'M TWO WEEKS PAST MY SECOND DOSE. CAN I GET A HORSE? THANK YOU ALL FOR COMING. WHAT IF I WEAR A MASK? WHAT IF THE HORSE DOES? (MEETING ENDED BY HOST)

Below the comic, there are navigation buttons: < PREV, img 740 x 350.5, NEXT >, and >|. Below the navigation buttons, there is a permanent link to this comic: <https://xkcd.com/2434/> and an image URL for hotlinking/embedding: https://imgs.xkcd.com/comics/vaccine_guidance.png.

The browser's developer tools (Inspector) are open, showing the HTML structure of the comic. The selected element is the `` tag for the comic image, with the following attributes:

```

```

The browser's layout tools (Layout) are also open, showing the Flexbox container for the comic image, with the following properties:

```
element {
  image-orientation: none;
}
img {
  border: 0;
}
Inherited from body
body {
  ...
}
```

I have permission to do this: see <https://xkcd.com/license.html>.

Web Scraping: Beware!

THIS SLIDE DOES NOT CONSTITUTE LEGAL ADVICE.

- ▶ Web scraping may be illegal in your jurisdiction
- ▶ Web scraping may be forbidden by a website's Terms and Conditions
- ▶ Web scraping may be used to collect sensitive personal information

It is important to take adequate precautions

- ▶ Read the Terms and Conditions
- ▶ Ask for permission to the website owner (see, “webmaster”)
- ▶ Seek advice from your University / Institution / Human Studies committee

Web Scraped! ... Now What?

- ▶ Keep the code for scraping separate from the rest
- ▶ Store the resulting (raw) dataset on disk, label it with a date
- ▶ Clean the dataset (will take a lot of time)
- ▶ Keep the code for cleaning separate from the rest
- ▶ Store the resulting (clean) dataset on disk
- ▶ Research away!

Practice Time

Moving to a Jupyter Notebook

Wrapping up my TA Classes: What to Remember for the Exam

- ▶ Solving Bellman Equation for the fixed point $V(\cdot)$
 - ▶ Value Function Iter., Policy Function Iter. (i.e., Howard's Improvement), Direct Projection
 - ▶ Mind the *Curse of Dimensionality*
 - ▶ Policy functions are generally well-behaved (e.g., capital accumulation almost linear)
- ▶ Adding exogenous stochastic variables
 - ▶ Tauchen, Tauchen-Hussey, Rouwenhorst are limited to AR(1) processes
 - ▶ Literature has come up with approaches for general continuous Markov processes
- ▶ Solving for the equilibrium prices
 - ▶ Define the net excess demand
 - ▶ Take it to zero with a zero-finding routine
- ▶ Solving for equilibrium prices with heterogeneous agents
 - ▶ Combine exogenous transition matrix with policy functions
 - ▶ Obtain endogenous transition matrix across the whole vector of state variables
 - ▶ Compute the ergodic distribution
 - ▶ Aggregate agents and define aggregate demand minus aggregate supply
- ▶ Combining idiosyncratic and aggregate shocks
 - ▶ Computationally expensive, but feasible (esp. by mixing projection and perturbation methods)
 - ▶ Consider MIT shocks: IRF to "parameters" using only idiosyncratic uncertainty

Wrapping up my TA Classes: What to Remember for the Profession

I hope I gave you

- ▶ A glimpse into numerical methods in Economics
- ▶ A clear understanding of our reliance on computing, as Economists
- ▶ A solid foundation into projection methods and models with heterogeneous agents
- ▶ A sense of curiosity for “robustness” exercises
- ▶ An idea of how much Python can be flexible and powerful

Tip: when you work with code, do like this guy: <https://github.com/michaelstepner/healthinequality-code/blob/master/code/readme.md>

References



Edelman, B. (2012).

Using Internet Data for Economic Research.

Journal of Economic Perspectives, 26(2):189–206.