Web scraping in R or Python

Presented as part of the 2023 Digital Scholarship Series Showcase
February 24, 2023

Kristin Yeager *Head, Statistical Consulting*

Moira O'Neill *Graduate Assistant, Statistical Consulting*

Motivation

Having good manners + staying out of trouble

Reading HTML & CSS

Defining “web scraping”

To scrape or not to scrape

Demonstration

https://github.com/statconsultingKSU/webscraping-workshop

Code from today's workshop will be available at:

https://github.com/statconsultingKSU/webscraping-workshop

When and why we'd want to scrape the web

- The web is a rich source of quantitative and qualitative data
  - Examples: Online newspapers, posts on social media, government databases
  - We may be interested in the text of the content, or the images linked on the page, or the metadata (things like titles and authors).
- Often, the data we're interested in is bound up in a website's HTML structure
- We could copy/paste, or print to PDF… but the results are often unsatisfactory, and the process is time consuming for larger datasets

Defining “web scraping”

*What web scraping **is**:*

An automated process in which your computer

(1) contacts a website or web resource

(2) makes a copy of that website’s HTML or XML

(3) extracts the content of that HTML or XML as data

*What web scraping **is not**:*

Connecting to an application programming interface (API). APIs also contact the web and retrieve data, but they do not involve making a copy of the website’s document structure and pulling content out of that structure – the data returned is completely different in structure and format.

https://github.com/statconsultingKSU/webscraping-workshop

Differentiating between web scraping and APIs

Web scraping: Works with most publicly accessible web content. You don't have to rely on an API because technology exists to scrape any viewable webpage. You can "take what you need". However, the data that results is often messy and disorganized and requires additional processing before it can be considered "usable" for analysis.

API: Data is neatly organized and maintained for easy access. However, APIs require significant tech infrastructure and personnel investments on the part of the data owners/distributors.

https://github.com/statconsultingKSU/webscraping-workshop

Differentiating between web scraping and APIs: **A drinking water analogy**

Problem: You're thirsty and you want a glass of water.

Thirst: research problem/question

Water: the data you need to answer your question or solve your problem

https://github.com/statconsultingKSU/webscraping-workshop

Differentiating between web scraping and APIs: **A drinking water analogy**

Web scraping

(1) Get a vessel suitable to transport water that you can comfortably carry

(2) Go to a freshwater lake or river and fill the vessel with the “raw water”

(3) Carry the vessel back to your house

(4) Boil the water to make it safe for drinking

(5) You can now use the water you’ve collected to quench your thirst

https://github.com/statconsultingKSU/webscraping-workshop

Differentiating between web scraping and APIs: **A drinking water analogy**

APIs

(1) Go to a place where there is working water treatment and delivery infrastructure

(2) Get a cup from the cupboard

(3) Go to the nearest sink and turn on the tap

(4) Fill the cup with as much water as it will hold, then turn off the tap

(5) You can now use the water you've collected to quench your thirst

https://github.com/statconsultingKSU/webscraping-workshop

Having good manners + Staying out of trouble when scraping

Acceptability, legal issues, technical issues

Staying out of trouble: Acceptability of scraped data for research

- Signs of growing acceptance of scraped data for research: we see federal research funding from the NSF for projects that use web scraping and publications in top journals

NSF_ORG	AWARDS_WS
Office of the Director	26
Mathematical & Physical Sciences	200
Social, Behavioral, & Econ Sciences	95
Computer Info Sci & Engineering	487
Geosciences	437
Engineering	133
Biological Sciences	414
STEM Education	177
Tech, Innovation, & Partnerships	48

Implicit: Federal Research Funding

Explicit: Pubs in Top Journals

Staying out of trouble: Acceptability of scraped data

- But also: Not everyone will find scraped data acceptable in all situations
- For research: Scraped data has not always been accepted as being "research-quality"
  - Ethics (do individuals have a reasonable expectation of privacy)
  - Provenance (has data been edited)
  - Reproducibility (can others repeat what you did and get the same results)
- For commercial use: This is quite risky – we are not endorsing this
- For personal use: In general, as long as you are working with publicly available data + abiding by the site's rules + not overloading the site with requests, you are probably OK

https://github.com/statconsultingKSU/webscraping-workshop

Staying out of trouble: Copyright

- Original work is automatically protected by the specific terms of its ***copyright***, and this includes work created in or disseminated through the digital realm.
- Emphasis on **original**: online content or databases must demonstrate creativity to qualify.
- Exceptions for ***"fair use"*** of portions of copyrighted material are usually limited to non-commercial activities (e.g., teaching, criticism, school presentations, and as quoted sources in research papers).

https://github.com/statconsultingKSU/webscraping-workshop

Staying out of trouble: Copyright vs. Terms of Use

Some content may not be considered “original” or “creative” work for the purpose of copyright, BUT: Private firms with public-facing websites may attempt to protect non-copyrighted web material through a ***Terms of Use*** statement.

The extent to which these statements are defensible in court depends on the legal system with jurisdiction.

United States:

In ***Linkedin v. hiQ****,* the U.S. Supreme Court explicitly tasked the 9th Circuit Court of Appeals with determining whether web scraping was a criminal offense under the Computer Fraud and Abuse Act (CFAA). In what’s been widely characterized as a “pro-scraping” ruling, the judges determined “the concept of ‘without authorization’ under the CFAA does not apply to public websites” (Neuberger 2022, p. 342).

https://github.com/statconsultingKSU/webscraping-workshop

Having good manners: Limiting your impact while scraping

- Websites aren't free to run – there's technically a "cost" every time a computer visits a website
  - This includes visits from robots and scraping scripts!
- The servers that host websites can also be "overwhelmed" when there are many visitors/requests – when this happens, it slows down the site for everyone, and can even make the site inaccessible
- Check the site's **robots.txt** file (if it has one) to see if it expressly disallows scraping

https://github.com/statconsultingKSU/webscraping-workshop

Final note: Having good manners and staying out of trouble when scraping

Don’t overload the site’s server!

For large scraping projects, avoid biting the hand that feeds you by inserting a time delay between scraping tasks.

https://github.com/statconsultingKSU/webscraping-workshop

To scrape or not to scrape

Think of scraping as a “last resort” if no other options to acquire your data exist. Here’s what you should check for first

Selecting the right data collection tools

Does the dataset already exist as secondary data (such as a CSV file or an embedded file with a direct link)

- Data repositories – browse https://www.re3data.org/
- Library's subscription databases – for example:
  - WRDS (stock market data)
  - ICPSR (social sciences data)
  - Factiva (text and metadata of national and international news articles)
- If you're in a class that uses a textbook and it has datasets used in its homework problems: check the publisher's textbook website – most of the datasets can be downloaded directly

https://github.com/statconsultingKSU/webscraping-workshop

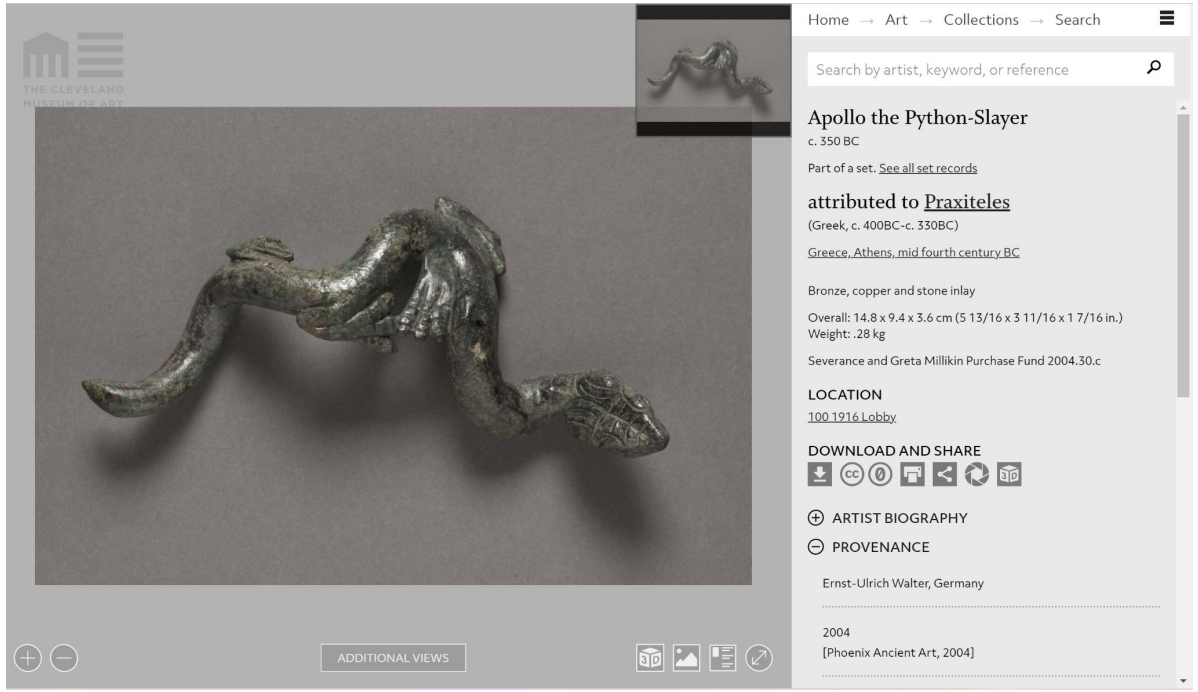
Selecting the right data collection tools

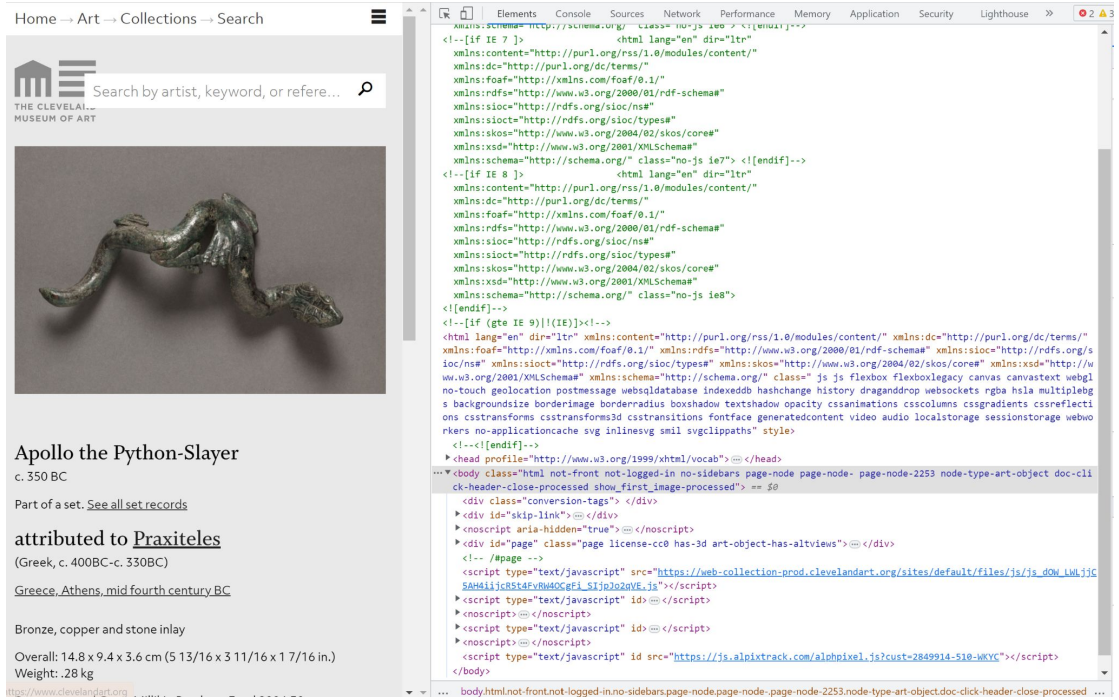
Next check if an API exists to "formally" extract data from a system. A good rule of thumb is: Companies whose business is data are more likely to have APIs.

- Social media firms
  - e.g. Twitter API can return tweet data in JSON format – but they just got rid of their free tier
- Government agencies
  - e.g. US Census has an API for its primary and American Community Survey data
- Companies and organizations
  - e.g. New York Times
- Cloud-based software and services
  - Google Drive
  - Qualtrics (online surveys): The KSU Qualtrics subscription has API access!

https://github.com/statconsultingKSU/webscraping-workshop

Case study:
The Cleveland Museum of Art's Digital Collection





URL: https://www.clevelandart.org/art/2004.30.c

https://github.com/statconsultingKSU/webscraping-workshop

Step 1: Looking at the HTML and CSS to find out what to target

There's a whole world under the hood of a website.

"Markup languages" like HyperText Markup Language (***HTML***), Extensible Markup Language (***XML***) define the content and structure of a document (such as links, text, lists, tables, and images), while Cascading Style Sheets (***CSS***) control the aesthetics (such as font families, colors, line spacing, and more).

https://www.clevelandart.org/art/2004.30.c

https://github.com/statconsultingKSU/webscraping-workshop

Step 1: Looking at the HTML and CSS to find out what to target

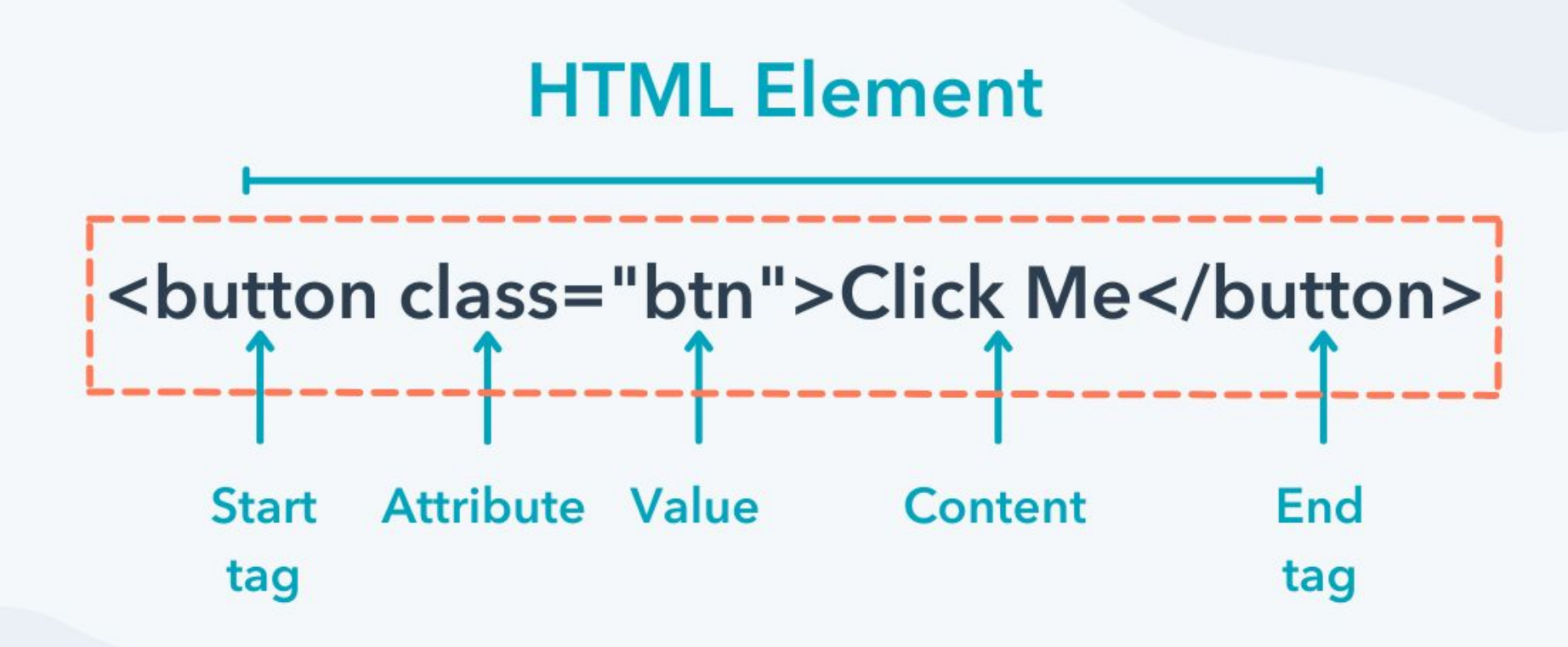


Image source: https://blog.hubspot.com/website/html-elements

HTML elements:
Text & Images

Text:

```
<p></p> Paragraphs
<h1></h1>, <h2></h2>, …, <h6>></h6> Headings
<ol></ol> Ordered lists
  <li></li> List items
<ul></ul> Unordered lists
  <li></li> List items
```

Images:

```
<img src=”image-url.png”/>
```


https://github.com/statconsultingKSU/webscraping-workshop

HTML elements:
Links & Tables

Links:

```
<a href="https://google.com">Link destination</a>
```

Tables:

```
<table></table>
   <thead></thead> Table header row(s)
   <tr></tr> Table rows
      <td></td> Table cells
```


https://github.com/statconsultingKSU/webscraping-workshop

HTML elements:
Generic Containers and Markup

Div: Acts like a “box” – often invisible – often used for arrangement of content or to denote sections of the page

```
<div></div>
```

```
<div class="pane-content">
  <div  about="/art/1947.209" typeof="sioc:Item foaf:Document" class="ds-1col node nod
```

Span: Used for inline markup to apply aesthetic changes, or paired with classes or ids to denote the variety of data

```
<span></span>
```

```
<span class="fieldset-legend">Artist biography</span>
```


https://github.com/statconsultingKSU/webscraping-workshop

CSS rules:
Classes and IDs

Classes:

Can be applied to more than one element
Names begin with “.” in <style> section
Applied using class argument

IDs:

Are unique – can only apply an ID to one and only one element on a page
Names begin with “#” in <style> section

https://github.com/statconsultingKSU/webscraping-workshop

Step 2: Picking a scraping tool

- In this workshop, we're going to be showing mostly Python and/or R, but there are *a lot* of other options for scraping
- Why did we choose Python and R? → Because we come from disciplines where these are the tools of choice, and because we are approaching this as a data science task; both of these softwares are good options in this case.

- Choose the tool you're most comfortable using!

Python in Google Colab

- Python and R can be installed as programs on your computer
- Alternatively, you can use Python on any computer with an internet connection from any browser using Google Colab (https://colab.research.google.com/)
  - You can create “notebooks” in Colab that contain executable Python code and narrative
    - You can view these notebooks without logging in
    - If you want to run the code in the notebooks, you’ll need a Google account

Python vs. R

Python 

- Free
- Open source
- Windows, Mac, Linux
- Birthdate: 1991
- General-purpose programming language
  - Really good at doing a whole lot of things

Download:

https://www.python.org/

R 

- Free
- Open source
- Windows, Mac, Linux
- Birthdate: 1993
- Statistical software and statistical programming language
  - Really good at doing one very specific thing (stats)

Download:

https://cran.r-project.org/

https://colab.research.google.com/drive/1ABCyGzk7XTvEj8swJvKuIJDdReH0XO0L?usp=sharing

Kristin Yeager *Head, Statistical Consulting*
Moira O'Neill *Graduate Assistant, Statistical Consulting*

References - Python

Python libraries used:

- **Requests**
- **BeautifulSoup**

You might also be interested in:

- **Selenium** Library to use with webdriver (e.g., Chromedriver) for dynamic scraping and crawling tasks
- **json** Python library for collecting and storing data

References - R

R packages used:

- rvest - for “harvesting” data from the web
- xml2 - for parsing HTML/XML
- Base R function `download.file()`

You might also be interested in these R packages:

- httr – implements http verbs
- jsonlite - package for working with JSON data
- polite – newer package that attempts to automate “polite” scraping and checking robots.txt

References

Cleveland Museum of Art Digital Collections

https://www.clevelandart.org/art/collection/search

Cleveland Museum of Art Open Access API

https://openaccess-api.clevelandart.org/