BCCP web scraping course

Day 1

Table of Contents

Day 1

very short intro to Python

Very short intro to Python

Day 1

Intro to Webscraping

Introduction to Webscraping

- Basic idea: Turn information on website to structured data
- Typical workflow:
  1. Look at website to decide best approach
     - Is an Application Programming Interface (API) available?
     - Do the HTML elements have fixed names?
     - Does the page load statically or dynamically?
  2. Download information from URL
  3. Turn information into structured data and save

Some concepts

- APIs
- HTML parsing vs text matching
- Static vs dynamic websites

APIs

- If available, a convenient way to get pre-structured data (usually JSON or XML).
- Example: See 0_*intro_webscraping.ipynb*

HTML parsing

- Use structure of HTML code to find needed information.
- Works best if the code is well-structured and element names are fixed.
- Example: See 0_*intro_webscraping.ipynb*

Text pattern matching

- If the HTML code is not well-structured or names change, text pattern matching is an alternative.
- Idea: Take text from (parts of) a page and find needed information by matching a regular expression
- Example: See 0_*intro_webscraping.ipynb*

Static vs dynamic websites

- On static websites, the entire content is loaded immediately.
- On dynamic websites, content may load subsequently or after user action, making them usually more complicated to scrape.
- Example: See 0_*intro_webscraping.ipynb*

Important Python packages

- `requests`: To load URL and recover source code (for static web pages)
- `beautifulsoup4`: To turn HTML code to navigable Python object
- `selenium`: To automate browsers
- `pandas`: To create DataFrames

Day 1

APIs

Application Programming Interface

Twitter API

- "Conduct historical research and search from Twitter's massive archive of publicly-available Tweets posted since March 2006?"
- "Listen in real-time for Tweets of interest?"

Day 2

Table of Contents

Day 2

HTML parsing

Basic principle

1. Load URL and save HTML source code: `requests` or `urllib2`
2. Convert source code to Python object: `beautifulsoup4`
3. Navigate the source code and save needed elements

Day 2

Text pattern matching

Day 3

Table of Contents

Day 3

Browser automation

Day 3

Own script