15 Web Scraping

Web Scraping 101 - Reading HTML With rvest

1. Purpose Of rvest
    - rvest helps you scrape data from websites
    - it makes html extraction feel similar to using dplyr
    - the goal is to easily pull structured data out of unstructured web pages
2. Reading And Parsing HTML
    - read_html() downloads and parses the html of a webpage
    - parsed html becomes an xml-like document you can query
    - selecting elements requires using css or xpath selectors
3. Selecting Elements With CSS Or XPath
    - html_elements() pulls all matching nodes
    - html_element() pulls the first matching node
    - css selectors like ".class" "#id" and "tag" help find content
    - xpath selectors allow more complex queries when needed
4. Extracting Text And Attributes
    - html_text() retrieves the human-readable text
    - html_attr() gets attribute values such as href or src
    - attribute extraction is essential for scraping links and images
5. Scraping Tables
    - html_table() converts html tables into data frames
    - it automatically detects header rows and cell content
    - cleaning may be required if tables are irregular
6. Navigating HTML Structure
    - web pages may require drilling down multiple layers of tags
    - combining html_elements() with html_attr() or html_text() refines extraction
    - understanding nested html helps target the right content
7. Scraping Multiple Pages
    - purrr::map() can iterate over lists of urls
    - looping through pages allows scraping across categories or results pages
    - always check that selectors behave consistently across pages
8. Ethical And Practical Considerations
    - scraping should follow robots.txt and site terms of service
    - adding delays between requests avoids overloading servers
    - unstable or changing html may break scraping code