Lab Assignment 5: Web Scraping

DS 6001: Practice and Application of Data Science

Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

For the following problems, you will be scraping http://books.toscrape.com/. This website is a fake book retailer, designed to mimic the design of many retail websites. It exists solely to help students practice web-scraping, so there aren't going to be any ethical concerns with this particular exercise, and there shouldn't be any issues with rate limits or other gates that could prevent web-scraping. Take a moment and look at this website, so that you know what you will be working with.

Your goal is to generate a dataframe with four columns: one for the title, one for the price, one for the star-rating, and one for the book cover JPEG's URL. The dataframe will also 1000 rows, one for each of the 1000 books listed on the 50 pages of this website.

Problem 0

Import the following libraries:

In [1]:
```
import numpy as np
import pandas as pd
import requests
import json
from bs4 import BeautifulSoup
import sys
sys.tracebacklimit = 0 # turn off the error tracebacks
```

In [2]:
```
url = "https://httpbin.org/user-agent"
r = requests.get(url)
myjson = json.loads(r.text)
useragent = myjson["user-agent"]
headers = {"User-Agent": useragent, "From": "mdg7wj@virginia.edu"}
```

Problem 1

Pull the HTML code from http://books.toscrape.com/. Make sure you provide a user

agent string. Then parse this HTML code and save the parsed code as a separate Python variable. [3 points]

In [3]:
```
r = requests.get("http://books.toscrape.com/", headers=headers)
soup = BeautifulSoup(r.text, "html.parser")
```

Problem 2

Extract all 20 of the book titles and save them in a list. [2 points]

In [4]:
```
titles = [x.h3.a["title"] for x in soup.find_all("article", "product_pod")]
titles
```

Out[4]:
```
['A Light in the Attic',
 'Tipping the Velvet',
 'Soumission',
 'Sharp Objects',
 'Sapiens: A Brief History of Humankind',
 'The Requiem Red',
 'The Dirty Little Secrets of Getting Your Dream Job',
 'The Coming Woman: A Novel Based on the Life of the Infamous Feminist, Victoria Woodhull',
 'The Boys in the Boat: Nine Americans and Their Epic Quest for Gold at the 1936 Berlin Olympics',
 'The Black Maria',
 'Starving Hearts (Triangular Trade Trilogy, #1)',
 "Shakespeare's Sonnets",
 'Set Me Free',
 "Scott Pilgrim's Precious Little Life (Scott Pilgrim #1)",
 'Rip it Up and Start Again',
 'Our Band Could Be Your Life: Scenes from the American Indie Underground, 1981-1991',
 'Olio',
 'Mesaerion: The Best Science Fiction Stories 1800-1849',
 'Libertarianism for Beginners',
 "It's Only the Himalayas"]
```

Problem 3

Extract the price of each of the 20 books and save these prices in a list. (The prices are listed in British pounds, and include the £ symbol. Remove the £ symbols: if you've saved the prices in a list named `prices`, then the following code should work: `prices = [s.replace('Â£', '') for s in prices]`.) [2 points]

In [5]:
```
prices = [x.text for x in soup.find_all("p", "price_color")]
prices = [s.replace('Â£', '') for s in prices]
prices
```

Out[5]:
```
['51.77',
 '53.74',
 '50.10',
 '47.82',
 '54.23',
 '22.65',
 '33.34',
 '17.93',
 '22.60',
 '52.15',
 '13.99',
 '20.66',
 '17.46',
 '52.29',
 '35.02',
 '57.25',
 '23.88',
 '37.59',
 '51.33',
 '45.17']
```

Problem 4

Extract the star level ratings for the 20 books. [Hint: for tags such as `<p class="star-rating One">` in which the class has a space, the class is actually a list in which the first item in the list is `"star-rating"` and the second item in the list is `"One"` . It's possible to search on either item in this list.] [3 points]

In [6]:
```
star_ratings = [x["class"][1] for x in soup.find_all("p", "star-rating")]
star_ratings
```

Out[6]:
```
['Three',
 'One',
 'One',
 'Four',
 'Five',
 'One',
 'Four',
 'Three',
 'Four',
 'One',
 'Two',
 'Four',
 'Five',
 'Five',
 'Five',
 'Three',
 'One',
 'One',
 'Two',
 'Two']
```

Problem 5

Extract the URLs for the JPEG thumbnail images that show the covers of the 20 books. (Maybe we want to mine the images to build models that predict the star level, literally judging books by their covers.) [2 points]

In [7]:
```
jpegs = [x["src"] for x in soup.find_all("img", "thumbnail")]
jpegs
```

Out[7]:
```
['media/cache/2c/da/2cdad67c44b002e7ead0cc35693c0e8b.jpg',
 'media/cache/26/0c/260c6ae16bce31c8f8c95daddd9f4a1c.jpg',
 'media/cache/3e/ef/3eef99c9d9adef34639f510662022830.jpg',
 'media/cache/32/51/3251cf3a3412f53f339e42cac2134093.jpg',
 'media/cache/be/a5/bea5697f2534a2f86a3ef27b5a8c12a6.jpg',
 'media/cache/68/33/68339b4c9bc034267e1da611ab3b34f8.jpg',
 'media/cache/92/27/92274a95b7c251fea59a2b8a78275ab4.jpg',
 'media/cache/3d/54/3d54940e57e662c4dd1f3ff00c78cc64.jpg',
 'media/cache/66/88/66883b91f6804b2323c8369331cb7dd1.jpg',
 'media/cache/58/46/5846057e28022268153beff6d352b06c.jpg',
 'media/cache/be/f4/bef44da28c98f905a3ebec0b87be8530.jpg',
 'media/cache/10/48/1048f63d3b5061cd2f424d20b3f9b666.jpg',
 'media/cache/5b/88/5b88c52633f53cacf162c15f4f823153.jpg',
 'media/cache/94/b1/94b1b8b244bce9677c2f29ccc890d4d2.jpg',
 'media/cache/81/c4/81c4a973364e17d01f217e1188253d5e.jpg',
 'media/cache/54/60/54607fe8945897cdcced0044103b10b6.jpg',
 'media/cache/55/33/553310a7162dfbc2c6d19a84da0df9e1.jpg',
 'media/cache/09/a3/09a3aef48557576e1a85ba7efea8ecb7.jpg',
 'media/cache/0b/bc/0bbcd0a6f4bcd81ccb1049a52736406e.jpg',
 'media/cache/27/a5/27a53d0bb95bdd88288eaf66c9230d7e.jpg']
```

Problem 6

Create a dataframe with one row for each of the 20 books, and the book titles, prices, star ratings, and cover JPEG URLs as the four columns. [2 points]

In [9]:
```
pd.DataFrame({"title": titles, "price": prices, "star_rating": star_ratings,
```

Out[9]:

	title	price	star_rating	cover_jpeg
0	A Light in the Attic	51.77	Three	media/cache/2c/da/2cdad67c44b002e7ead0cc35693c...
1	Tipping the Velvet	53.74	One	media/cache/26/0c/260c6ae16bce31c8f8c95daddd9f...
2	Soumission	50.10	One	media/cache/3e/ef/3eef99c9d9adef34639f51066202...
3	Sharp Objects	47.82	Four	media/cache/32/51/3251cf3a3412f53f339e42cac213...
4	Sapiens: A Brief History of Humankind	54.23	Five	media/cache/be/a5/bea5697f2534a2f86a3ef27b5a8c...
5	The Requiem Red	22.65	One	media/cache/68/33/68339b4c9bc034267e1da611ab3b...
6	The Dirty Little Secrets of Getting Your Dream...	33.34	Four	media/cache/92/27/92274a95b7c251fea59a2b8a7827...
7	The Coming Woman: A Novel Based on the Life of...	17.93	Three	media/cache/3d/54/3d54940e57e662c4dd1f3ff00c78...
8	The Boys in the Boat: Nine Americans and Their...	22.60	Four	media/cache/66/88/66883b91f6804b2323c8369331cb...
9	The Black Maria	52.15	One	media/cache/58/46/5846057e28022268153beff6d352...
10	Starving Hearts (Triangular Trade Trilogy, #1)	13.99	Two	media/cache/be/f4/bef44da28c98f905a3ebec0b87be...
11	Shakespeare's Sonnets	20.66	Four	media/cache/10/48/1048f63d3b5061cd2f424d20b3f9...
12	Set Me Free	17.46	Five	media/cache/5b/88/5b88c52633f53cacf162c15f4f82...
13	Scott Pilgrim's Precious Little Life (Scott Pi...	52.29	Five	media/cache/94/b1/94b1b8b244bce9677c2f29ccc890...
14	Rip it Up and Start Again	35.02	Five	media/cache/81/c4/81c4a973364e17d01f217e118825...

	title	price	star_rating	cover_jpeg
15	Our Band Could Be Your Life: Scenes from the A...	57.25	Three	media/cache/54/60/54607fe8945897cdcced0044103b...
16	Olio	23.88	One	media/cache/55/33/553310a7162dfbc2c6d19a84da0d...
17	Mesaerion: The Best Science Fiction Stories 18...	37.59	One	media/cache/09/a3/09a3aef48557576e1a85ba7efea8...
18	Libertarianism for Beginners	51.33	Two	media/cache/0b/bc/0bbcd0a6f4bcd81ccb1049a52736...
19	It's Only the Himalayas	45.17	Two	media/cache/27/a5/27a53d0bb95bdd88288eaf66c923...

Problem 7

Create a function that takes the URL of the webpage to scrape as an input, applies the code you wrote for questions 1 through 6, and generates the dataframe from question 6 as the output. [3 points]

In [10]:
```
def bookscrape(url):
    r = requests.get(url, headers=headers)
    soup = BeautifulSoup(r.text, "html.parser")
    titles = [x.h3.a["title"] for x in soup.find_all("article", "product_pod
    prices = [x.text for x in soup.find_all("p", "price_color")]
    prices = [s.replace('Â£', '') for s in prices]
    star_ratings = [x["class"][1] for x in soup.find_all("p", "star-rating")
    jpegs = [x["src"] for x in soup.find_all("img", "thumbnail")]
    df = pd.DataFrame({"title": titles, "price": prices, "star_rating": star
    return df
```

Problem 8

Notice that there are many pages to http://books.toscrape.com/. When you click on "Next" in the bottom-right corner of the screen, it takes you to http://books.toscrape.com/catalogue/page-2.html. The front page is the same as http://books.toscrape.com/catalogue/page-1.html, and there are 50 total pages.

Write a loop that uses the function you wrote in question 7 to scrape each of the 50 pages, and append each of these data frames together. If you write this loop correctly, your dataframe will have 1000 rows (20 books on each of the 50 pages).

Some hints:

- Typing `new_df = pd.DataFrame()` with nothing in the parentheses will create an empty data frame on which new data can be appended.
- There are many loops you can use, but the most straightforward one is a for-values loop that counts from 1 to 50. In Python, you can initialize such a loop with for i in range(1, 51):, and indenting every line below it that belongs inside the loop. Inside the loop, the letter i is now a stand-in for the number currently being considered.
- You will need to figure out how to replace the number in URLs like http://books.toscrape.com/catalogue/page-2.html with the number currently under consideration in the loop. You might need the `str()` function, which turns numeric values into strings.

[3 points]

In [11]:
```
bookscrape_df = pd.DataFrame()
for i in range(1,51):
    bookscrape_df = pd.concat([bookscrape_df, bookscrape("http://books.toscr
bookscrape_df
```

Out[11]:

	title	price	star_rating	cover_jpeg
0	A Light in the Attic	51.77	Three	../media/cache/2c/da/2cdad67c44b002e7ead0cc356...
1	Tipping the Velvet	53.74	One	../media/cache/26/0c/260c6ae16bce31c8f8c95dadd...
2	Soumission	50.10	One	../media/cache/3e/ef/3eef99c9d9adef34639f51066...
3	Sharp Objects	47.82	Four	../media/cache/32/51/3251cf3a3412f53f339e42cac...
4	Sapiens: A Brief History of Humankind	54.23	Five	../media/cache/be/a5/bea5697f2534a2f86a3ef27b5...
...	...	...	...	...
15	Alice in Wonderland (Alice's Adventures in Won...	55.53	One	../media/cache/96/ee/96ee77d71a31b7694dac6855f...
16	Ajin: Demi-Human, Volume 1 (Ajin: Demi-Human #1)	57.06	Four	../media/cache/09/7c/097cb5ecc6fb3fbe1690cf0cb...
17	A Spy's Devotion (The Regency Spies of London #1)	16.97	Five	../media/cache/1b/5f/1b5ff86f3c75e51e24c573d3f...
18	1st to Die (Women's Murder Club #1)	53.98	One	../media/cache/2b/41/2b4161c5b72a4ae386b644682...
19	1,000 Places to See Before You Die	26.08	Five	../media/cache/d7/0f/d70f7edd92705c45a82118c3f...

1000 rows × 4 columns

In [ ]: