SCRAPING SOME NUMBERS FROM NUMBEO

Rafał Misiórski, 372727

Kamil Golis, 437959

PROJECT PURPOSE

The aim of the project is to gather data using web scraping tools used throughout the course (including Beautiful Soup, Scrapy and Selenium). Three scrapers were constructed and applied in order to collect data on prices of products registered on Numbeo all around the world. The level of prices aggregation chosen for the project is the country level.

Our motivation to scrape the website is the abundance of information and potential uses of the collected data on prices (Three use cases are presented later in this document).

WHAT IS NUMBEO?

Numbeo is a database aggregating costs of living across the globe. What is interesting, all of the data are collected from volunteers all around the world – anyone may enter appropriate data concerning costs of living in a given country/city. Apart from prices for different categories of products, the website presents indicators for quality of life, e.g. housing, crime, healthcare indicators. What may be interesting, the website aggregates also job posts (remote work/ work with relocation).

SCRAPER DESCRIPTION

Short description of scraper mechanics

Beautiful Soup

The beautiful soup scraper is divided into several parts. Below find a description for each part

1. Import appropriate libraries.
2. Set a boolean variable *limit_scraper*, which determines number of scraped pages (default set to True = scrap first 100 pages, False = scrap all pages).
3. Prepare list of links to scrap data from (in fact, prepare list of countries to get information about, there is a separate link for each country).
4. Prepare list of countries (formatted appropriately to present them in a more readable form) – items from this list are going to be included in the final dataset.
5. Prepare empty lists for all of the columns of the final dataset.
6. Run a for loop – scrape the information needed from each link (or from first 100 links) – next points are in fact subparts of this for loop.
7. Open the appropriate link.
8. Scrape three types of information from the link: whole table with all of the data (basing on that we are going to extract product names), elements containing information about prices (class = currency) – these elements are a base for extracting prices and elements of class priceBarTd, which constitute a base for getting information on price ranges.
9. Use appropriate regular expressions, set conditions concerning found content, format the output and append to appropriate lists (for countries, products, prices and price ranges). For price ranges additional part -> divide the range character column into two new numerical columns: min and max price.
10. Construct the final data frame using the lists with output and save it as a csv file.
11. Get a final message informing that scraping is done – and we are done with the beautiful scraper as well.

Scrapy

The process of web scraping by using Scrapy is divided into 2 separate parts. The first part relies on setting up a spider that gathers all links related to costs from the Numbeo.com site for each country. The second spider goes into each link, parses the content of a site (by Beautiful Soup library) and returns output in a CSV file as a data frame.

Spider: S1

1. Import of Scrapy library.
2. Run of parse function which collects names of countries.
3. Concatenation of links to obtain a separate link for each country.

Spider: S2 (Steps 2-10 are the same as per Beautiful Soup approach).

1. Import of Scrapy, Beautiful Soup and Pandas library.
2. Set a boolean variable *limit_scraper*, which determines the number of scraped pages (default set to True = scrap first 100 pages, False = scrap all pages).
3. Prepare a list of links to scrap data from (in fact, prepare a list of countries to get information about, there is a separate link for each country).
4. Prepare a list of countries (formatted appropriately to present them in a more readable form) – items from this list are going to be included in the final dataset.
5. Prepare empty lists for all of the columns of the final dataset.
6. Run a for a loop – scrape the information needed from each link (or from the first 100 links) – next points are in fact subparts of this for a loop.
7. Open the appropriate link.
8. Scrape three types of information from the link: the whole table with all of the data (based on that we are going to extract product names), elements containing information about prices (class = currency) – these elements are a base for extracting prices and elements of class priceBarTd, which constitute a base for getting information on price ranges.
9. Use appropriate regular expressions, set conditions concerning found content, format the output and append to appropriate lists (for countries, products, prices and price ranges). For price ranges, additional part -> divide the range character column into two new numerical columns: min and max price.
10. Construct the final data frame using the lists with output and save it as a CSV file.
11. Yield data frame to CSV file.

Selenium

The Script Selenium library gather

1. Import of Selenium, Pandas and Re library.
2. Opening the Numbeo website that consists of all links to the cost of products for a given country
3. Collecting all links (to scrape data through each of them).
4. Run for a loop – Scrape data from each link.
5. Opening a link from the list in a new browser window.
6. Scrape the content of a site by using Xpath response .
7. Extract three types of elements: products, prices and ranges by using the regex function.
8. Insert elements into separate dataframes.
9. Close the web browser.
10. Merge each element into the final data frame by using lists.
11. Construction of CSV file filled with final data frame.

SCRAPERS PERFORMANCE COMPARISON

We measured performace of the scrapers (scraping first 100 links). For each scraper we set a delay between scraping links as two seconds (in order to make our scrapers more servers-friendly). The best scraper in terms of performance is Scrapy (no surprise here). Detailed results present as follows:

Scraper	Run Time
Beautiful Soup	297.8 s
Scrapy	246 s
Selenium	1384.52 s

THE OUTPUT DATA

The final dataset contains information on scraped country, product name, mean price from all questionnaires, and price ranges in two views: whole range in one cell and then division of the range for minimum and maximum price.

The final dataset consists of 12815 records (excluding the headers) and 7 columns (6 of them are informative, as presented in the sample, the seventh is index column). There are 233 scraped pages (which translates into data on 233 countries) – for each country we observe prices of 55 items. Please note, that all prices are standardized in terms of currency – we decided to present all of the prices in USD (Numbeo allows to choose from variety of currencies, it is also possible to view data in domestic currency). Empty cells indicate missing data.

Find below a sample of the final scraped dataset:

country	product	price	range	min	max
Afghanistan	Meal, Inexpensive Restaurant	1.74	1.74-2.91	1.74	2.91
Afghanistan	Meal for 2 People, Mid-range Restaurant, Three-course	8.47	5.81-14.53	5.81	14.53
Afghanistan	McMeal at McDonalds (or Equivalent Combo Meal)	3.28	2.99-3.49	2.99	3.49
Afghanistan	Domestic Non-Alcoholic Beer (0.5 liter draught)	3.50	2.33-4.67	2.33	4.67
Afghanistan	Imported Non-Alcoholic Beer (0.33 liter bottle)	4.17			
Afghanistan	Cappuccino (regular)	1.09	0.35-2.91	0.35	2.91
Afghanistan	Coke/Pepsi (0.33 liter bottle)	0.39	0.23-0.69	0.23	0.69
Afghanistan	Water (0.33 liter bottle)	0.18	0.12-0.35	0.12	0.35
Afghanistan	Milk (regular), (1 liter)	0.61	0.35-0.76	0.35	0.76
Afghanistan	Loaf of Fresh White Bread (500g)	0.32	0.12-0.58	0.12	0.58

To make the described data perfectly clear, below definition of columns in the dataset:

No.	Column name	Definition / Comment
1	Country	Country for which the data was collected (note: collected prices are aggregated from all cities in given countries – people insert data on the city level)
2	Product	Name of the product (“product” is a general name used here, the scraped items include mortgage interest rates, average salaries etc.)
3	Price	Average price from the data inserted by the users
4	Range	Price range (in the form min-max). Due to the “-“ sign, it is a string type column
5	Min	Minimum price for a given country extracted from the range column
6	Max	Maximum price for a given country extracted from the range column

OUTPUT DATA ANALYSIS

The output data may be used to get information about prices of different products for many countries. We decided to build an analytical dashboard using Power Bi (attached as an extra appendix in appendix folder). Below find presented three use cases of the output data.

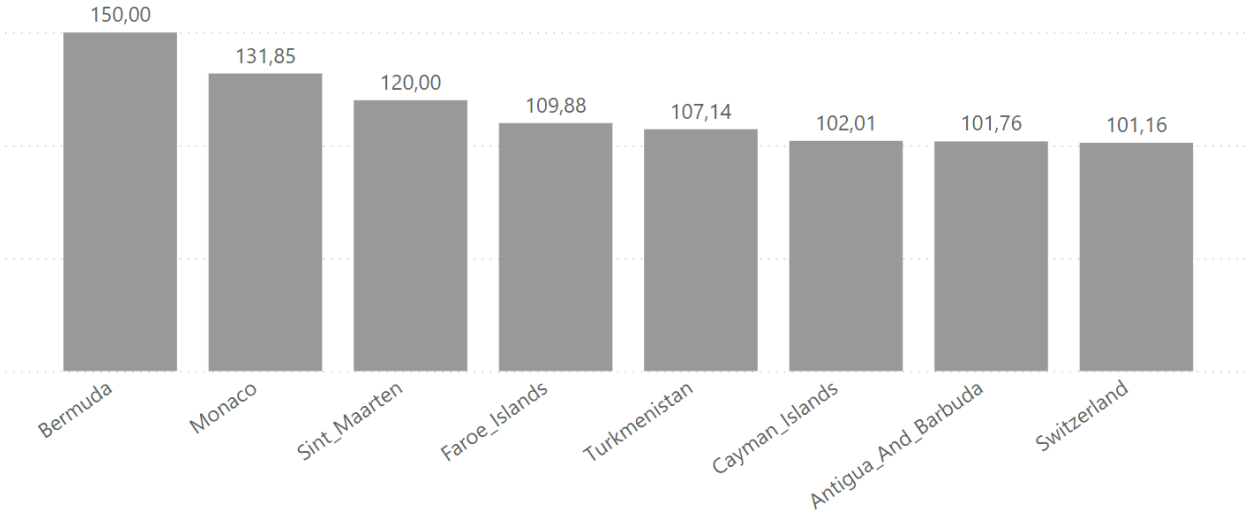
#1 Let's compare two different countries according to chosen products (useful e.g. before moving to another country) – remainder, units presented in USD currency:

Country	Poland			Malta		
Product	min_price	avg_price	max_price	min_price	avg_price	max_price
1 Pair of Jeans (Levis 501 Or Similar)	26.84	65.5	89.48	26.37	72.9	116.03
Apartment (1 bedroom) in City Centre	392.35	514.58	671.07	685.64	868.53	1107.57
Average Monthly Net Salary (After Tax)		1000			1208	
Bottle of Wine (Mid-Range)	3.47	5.59	7.83	4.22	6.33	10.55
Cinema, International Release, 1 Seat	4.47	5.59	7.38	7.91	9.49	13.71
Gasoline (1 liter)	1.12	1.36	1.53	1.37	1.44	1.59
Loaf of Fresh White Bread (500g)	0.47	0.81	1.34	0.84	1.17	2.11
Meal, Inexpensive Restaurant	4.47	6.71	10.07	9.49	15.82	26.37
Taxi 1km (Normal Tariff)	0.45	0.56	0.84	1.05	2.11	3.6

#2 Let's calculate summary statistics of prices of chosen products for ten countries that joined EU in 2004: Czechia, Cyprus, Estonia, Latvia, Lithuania, Hungary, Malta, Poland, Slovenia, Slovakia:

product	mean_price	median_price	min_price	max_price	std	skewness	kurtosis
Average Monthly Net Salary (After Tax)	1119.26	1138.5	836.23	1421	191.14	0.01	-1.55
Cappuccino (regular)	2.36	2.34	1.52	3.36	0.55	0.11	-0.86
Price per Square Meter to Buy Apartment in City Centre	2924.9	2957	1664	4677	847.99	0.48	-0.48
Water (1.5 liter bottle)	0.68	0.69	0.34	0.97	0.2	-0.18	-1.3

#3 Let's plot prices of meal consisting of three courses for two people in a mid-range restaurant (assuming that price is above $100):



DISTRIBUTION OF WORK AMONG TEAM MEMBERS

No.	Type	Task	Person
1	Soup Scraper	Prepare list of links	Kamil
2		Prepare list of countries	Kamil
3		Regex patterns preparation	Rafał
4		Extract information about countries	Kamil
5		Extract information about prices	Kamil
6		Extract information about price ranges	Kamil
7		Construction of final dataset	Kamil
8		Benchmarking and tests	Kamil
9		Code documentation	Kamil
10	Selenium Scraper	Prepare list of links	Kamil
11		Prepare list of countries	Kamil
12		Regex patterns preparation	Rafał
13		Extract information about countries	Rafał
14		Extract information about prices	Rafał
15		Extract information about price ranges	Rafał
16		Construction of final dataset	Rafał
17		Benchmarking and tests	Rafał
18		Code documentation	Rafał
19	Scrapy Scraper	Prepare list of links	Kamil
20		Prepare list of countries	Kamil
21		Regex patterns preparation	Rafał
22		Extract information about countries	Rafał
23		Extract information about prices	Rafał
24		Extract information about price ranges	Rafał
25		Construction of final dataset	Rafał
26		Benchmarking and tests	Rafał
27		Code documentation	Rafał
28	Data analysis & Dashboard	Final output transformations	Kamil
29		Preparation of additional external table on product categories	Kamil
30		Preparation of Power Bi dashboard	Kamil
31	Project Description	Project purpose / What is Numbeo	Kamil
32		What is Numbeo	Rafał
33		Beautiful Soup Scraper Description	Kamil
34		Selenium and Scrapy Description	Rafał
35		Scrapers Performance Comparison	Kamil
36		Output Data Description	Kamil
37		Output Data Analysis	Kamil