Analizing webscraped data

Federico Trotta

WEB SCRAPING PROJECT

Description of the project

NECESSITY:
When we want to perform some analyses, data is often not available. A possible solution to the lack of data is to obtain data from the web, by means of the so-called 'web scraping', in order to be able to analyze them.

WHAT HAVE I DONE:
To practice with 'web scraping' I got the data from this website, and then I saved them in a CSV file which I then analyzed in Jupyter Notebook.

Website analysis

The 'worldometers' site collects world statistics, many even in real-time.
For this project, I am interested in collecting world population data for each state. By clicking on the states, you enter another page with the specific population details of the state in question. In particular, I have been interested in the values of the population over time (from the 1950s to today)

Countries in the world by population (2022)

This list includes both **countries** and **dependent territories**. Data based on the latest *United Nations Population Division* estimates.
Click on the name of the country or dependency for current estimates (live population clock), historical data, and projected figures.
See also: World Population

Search:

#	Country (or dependency)	Population (2020)	Yearly Change	Net Change	Density (P/Km²)	Land Area (Km²)	Migrants (net)	Fert. Rate	Med. Age	Urban Pop %
1	Honduras	**9,904,607**	1.63 %	158,490	89	111,890	-6,800	2.5	24	57 %
2	United Arab Emirates	**9,890,402**	1.23 %	119,873	118	83,600	40,000	1.4	33	86 %
3	Djibouti	**988,000**	1.48 %	14,440	43	23,180	900	2.8	27	79 %
4	Saint Barthelemy	**9,877**	0.30 %	30	470	21		N.A.	N.A.	0 %

Some code...

I used VS CODE to develop the web scraping part.

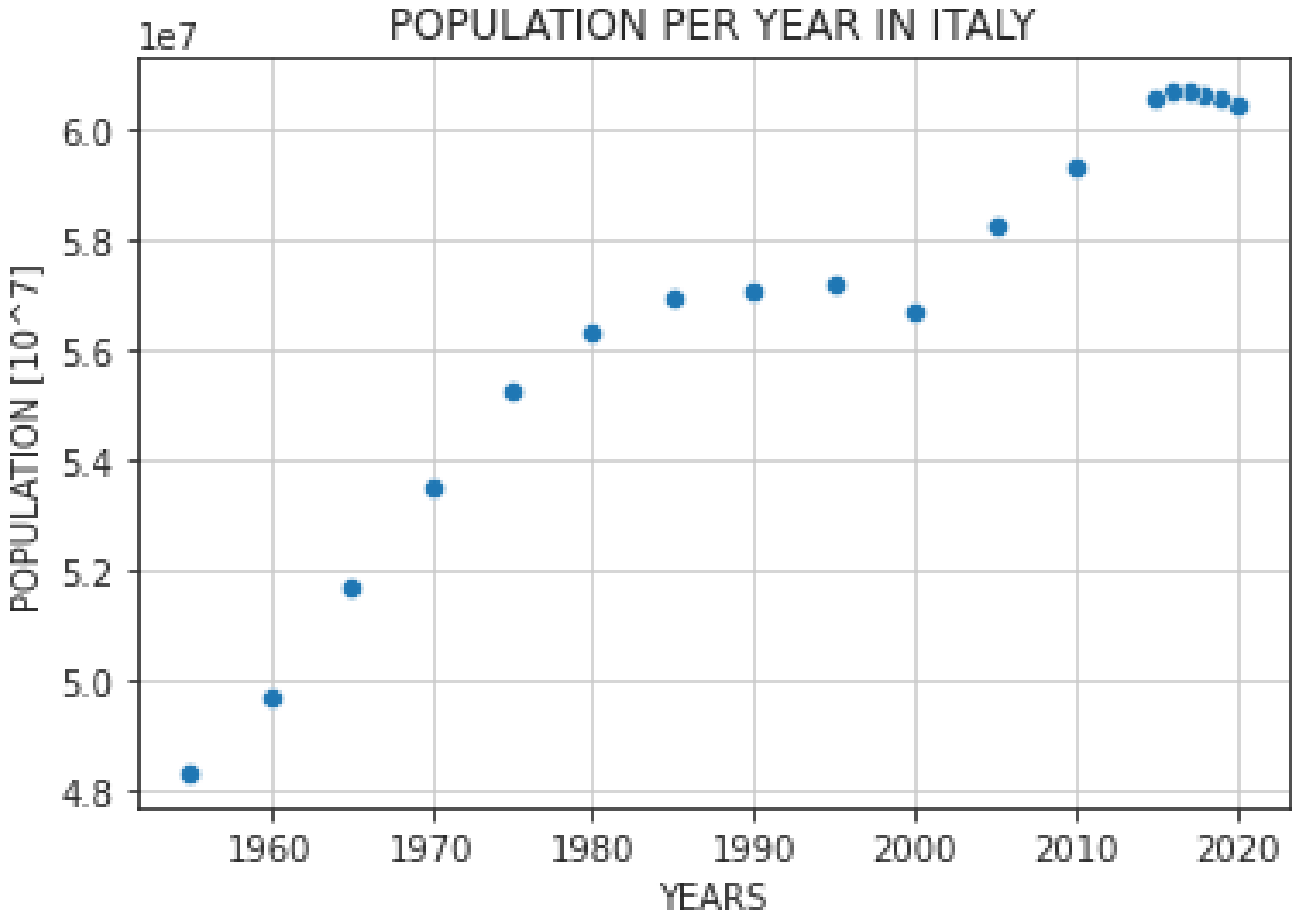
I created a CVS file in which to save the data I was interested in, that is: "Country" (the state), "Population" (the population value), "Year" (the reference year)

worldometers > spiders > countries.py > ...

```python
import scrapy
import os
import csv

#creating a csv for recap, if it does not exist
if not os.path.exists("recap.csv"):
    recap = open ("recap.csv", "w")
    writer = csv.writer(recap, delimiter=";")
    writer.writerow(["Country", "Population", "Year"]) #defining the header
else:
    recap = open ("recap.csv", "a")
    writer = csv.writer(recap, delimiter=";")

#class by which get the links to get inside and scrape the infos
class CountriesSpider(scrapy.Spider):
    name = 'countries'
    allowed_domains = ['www.worldometers.info']
    start_urls = ['https://www.worldometers.info/world-population/population-by-country/']

    def parse(self, response):
        countries = response.xpath('//td/a') #getting countries
        for country in countries:
            name = country.xpath('.//text()').get() #getting countries names
            link = country.xpath('.//@href').get() #getting link for the next function
```

Italy's data

At this point, you have the data to be analyzed. By opening the CSV in Jupyter Notebook, you can do the classic analysis. In this case, I wanted to visualize the population trend of Italy



Conclusions

A possible solution to solve the problem of lack of data, as we have seen, is web scraping which, with a certain ease, allows us to access the data available on the web.

Once the data source has been found (the site, or several sites), it is possible to save the data locally - for example, in a CSV, as in this case, in order to then be able to analyze them.