Web scraping

Web scraping, also known as web harvesting or web data extraction, is a type of data scraping used to gather information from websites.

In this session, we will cover the following concepts with the help of a business use case:

- Data acquisition through Web scraping

- Warnings are given to developers to alert them about circumstances that are not necessarily exceptions. Warnings are not the same as errors. It shows some message but the program will run. The `filterwarnings()` function, defined in the `warning` module, is used to handle warnings (presented, disregarded, or raised to exceptions).

In [1]:
```
import warnings
warnings.filterwarnings("ignore")
```

Health Care Rankings for Different European Countries

Beautiful Soup is a Python package that is used for web scraping. The urllib package is used to simplify the tasks of building, loading and parsing URLs. The Python datetime module supplies classes to work with date and time.

In [2]:
```
import numpy as np
import pandas as pd
from bs4 import BeautifulSoup
import requests
import csv
import re
import urllib.request as urllib2
from datetime import datetime
import os
import sys
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

- We are going to scrape data from Wikipedia. The data indicate rankings on different health indices such as patient rights and information, accessibility (waiting time for treatment), outcomes, range, the reach of services provided, prevention, and pharmaceuticals. The data are from the Euro Health Consumer index. In the following code, we read the data and use Beautiful Soup to convert the data into **bs4.BeautifulSoup** data.

In [3]:
```
url = 'https://en.wikipedia.org/wiki/Healthcare_in_Europe'
r = requests.get(url)
HCE = BeautifulSoup(r.text)
type(HCE)
```

Out[3]:
```
bs4.BeautifulSoup
```

- First, we must choose the table that we want to scrape. As many webpages have tables, we'll retrieve the exact table names from the HTML and store them in a list called `lst.`

In [4]:
```
webpage = urllib2.urlopen(url)
htmlpage= webpage.readlines()
lst = []
for line in htmlpage:
    line = str(line).rstrip()
    if re.search('table class', line) :
        lst.append(line)
```

In [5]:
```
len(lst)
```

Out[5]:
```
5
```

- This list `lst` has a length of 5.

- Now let us display `lst`.

In [6]:
```
lst
```

Out[6]:

```
['b\'<table class="wikitable floatright sortable" style="font-size: 90%">\\n\'',
 'b\'<div class="navbox-styles nomobile"><style data-mw-deduplicate="TemplateStyles:r1061467846">.mw-parser-out
put .navbox{box-sizing:border-box;border:1px solid #a2a9b1;width:100%;clear:both;font-size:88%;text-align:cente
r;padding:1px;margin:1em auto 0}.mw-parser-output .navbox .navbox{margin-top:0}.mw-parser-output .navbox+.navbo
x,.mw-parser-output .navbox+.navbox-styles+.navbox{margin-top:-1px}.mw-parser-output .navbox-inner,.mw-parser-o
utput .navbox-subgroup{width:100%}.mw-parser-output .navbox-group,.mw-parser-output .navbox-title,.mw-parser-ou
tput .navbox-abovebelow{padding:0.25em 1em;line-height:1.5em;text-align:center}.mw-parser-output .navbox-group
{white-space:nowrap;text-align:right}.mw-parser-output .navbox,.mw-parser-output .navbox-subgroup{background-co
lor:#fdfdfd}.mw-parser-output .navbox-list{line-height:1.5em;border-color:#fdfdfd}.mw-parser-output .navbox-lis
t-with-group{text-align:left;border-left-width:2px;border-left-style:solid}.mw-parser-output tr+tr>.navbox-abov
ebelow,.mw-parser-output tr+tr>.navbox-group,.mw-parser-output tr+tr>.navbox-image,.mw-parser-output tr+tr>.nav
box-list{border-top:2px solid #fdfdfd}.mw-parser-output .navbox-title{background-color:#ccf}.mw-parser-output .
navbox-abovebelow,.mw-parser-output .navbox-group,.mw-parser-output .navbox-subgroup .navbox-title{background-c
olor:#ddf}.mw-parser-output .navbox-subgroup .navbox-group,.mw-parser-output .navbox-subgroup .navbox-abovebelo
w{background-color:#e6e6ff}.mw-parser-output .navbox-even{background-color:#f7f7f7}.mw-parser-output .navbox-od
d{background-color:transparent}.mw-parser-output .navbox .hlist td dl,.mw-parser-output .navbox .hlist td ol,.m
w-parser-output .navbox .hlist td ul,.mw-parser-output .navbox td.hlist dl,.mw-parser-output .navbox td.hlist o
l,.mw-parser-output .navbox td.hlist ul{padding:0.125em 0}.mw-parser-output .navbox .navbar{display:block;font-
size:100%}.mw-parser-output .navbox-title .navbar{float:left;text-align:left;margin-right:0.5em}</style></div><
div role="navigation" class="navbox" aria-labelledby="Healthcare_in_Europe" style="padding:3px"><table class="n
owraplinks mw-collapsible autocollapse navbox-inner" style="border-spacing:0;background:transparent;color:inher
it"><tbody><tr><th scope="col" class="navbox-title" colspan="2"><style data-mw-deduplicate="TemplateStyles:r106
3604349">.mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-co
llapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul
{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-
right:-0.125em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-par
ser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{t
ext-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-deco
ration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .nav
bar-ct-mini{font-size:114%;margin:0 4em}</style><div class="navbar plainlinks hlist navbar-mini"><ul><li class
="nv-view"><a href="/wiki/Template:Healthcare_in_Europe" title="Template:Healthcare in Europe"><abbr title="Vie
w this template" style=";;background:none transparent;border:none;box-shadow:none;padding:0;">v</abbr></a></li>
<li class="nv-talk"><a href="/wiki/Template_talk:Healthcare_in_Europe" title="Template talk:Healthcare in Europ
e"><abbr title="Discuss this template" style=";;background:none transparent;border:none;box-shadow:none;paddin
g:0;">t</abbr></a></li><li class="nv-edit"><a class="external text" href="https://en.wikipedia.org/w/index.php?
title=Template:Healthcare_in_Europe&amp;action=edit"><abbr title="Edit this template" style=";;background:none
transparent;border:none;box-shadow:none;padding:0;">e</abbr></a></li></ul></div><div id="Healthcare_in_Europe"
style="font-size:114%;margin:0 4em"><a class="mw-selflink selflink">Healthcare in Europe </a></div></th></tr><t
r><th scope="row" class="navbox-group" style="width:1%">Sovereign states</th><td class="navbox-list-with-group
navbox-list navbox-odd hlist" style="width:100%;padding:0"><div style="padding:0 0.25em">\\n\'',
 'b\'<div class="navbox-styles nomobile"><link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:
r1061467846"/></div><div role="navigation" class="navbox" aria-labelledby="Health_in_Europe" style="padding:3p
x"><table class="nowraplinks mw-collapsible autocollapse navbox-inner" style="border-spacing:0;background:trans
parent;color:inherit"><tbody><tr><th scope="col" class="navbox-title" colspan="2"><link rel="mw-deduplicated-in
line-style" href="mw-data:TemplateStyles:r1063604349"/><div class="navbar plainlinks hlist navbar-mini"><ul><li
class="nv-view"><a href="/wiki/Template:Europe_topic" title="Template:Europe topic"><abbr title="View this temp
late" style=";;background:none transparent;border:none;box-shadow:none;padding:0;">v</abbr></a></li><li class
="nv-talk"><a href="/wiki/Template_talk:Europe_topic" title="Template talk:Europe topic"><abbr title="Discuss t
his template" style=";;background:none transparent;border:none;box-shadow:none;padding:0;">t</abbr></a></li><li
class="nv-edit"><a class="external text" href="https://en.wikipedia.org/w/index.php?title=Template:Europe_topic
&amp;action=edit"><abbr title="Edit this template" style=";;background:none transparent;border:none;box-shadow:
none;padding:0;">e</abbr></a></li></ul></div><div id="Health_in_Europe" style="font-size:114%;margin:0 4em"><a
href="/wiki/Health_in_Europe" class="mw-redirect" title="Health in Europe">Health in Europe </a></div></th></tr
><tr><th scope="row" class="navbox-group" style="width:1%">Sovereign states</th><td class="navbox-list-with-gro
up navbox-list navbox-odd hlist" style="width:100%;padding:0"><div style="padding:0 0.25em">\\n\'',
 'b\'<div class="navbox-styles nomobile"><link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:
r1061467846"/></div><div role="navigation" class="navbox" aria-labelledby="List_of_hospitals_in_Europe" style
="padding:3px"><table class="nowraplinks mw-collapsible autocollapse navbox-inner" style="border-spacing:0;back
ground:transparent;color:inherit"><tbody><tr><th scope="col" class="navbox-title" colspan="2"><link rel="mw-ded
uplicated-inline-style" href="mw-data:TemplateStyles:r1063604349"/><div class="navbar plainlinks hlist navbar-m
ini"><ul><li class="nv-view"><a href="/wiki/Template:Europe_topic" title="Template:Europe topic"><abbr title="V
iew this template" style=";;background:none transparent;border:none;box-shadow:none;padding:0;">v</abbr></a></l
i><li class="nv-talk"><a href="/wiki/Template_talk:Europe_topic" title="Template talk:Europe topic"><abbr title
="Discuss this template" style=";;background:none transparent;border:none;box-shadow:none;padding:0;">t</abbr>
</a></li><li class="nv-edit"><a class="external text" href="https://en.wikipedia.org/w/index.php?title=Templat
e:Europe_topic&amp;action=edit"><abbr title="Edit this template" style=";;background:none transparent;border:no
ne;box-shadow:none;padding:0;">e</abbr></a></li></ul></div><div id="List_of_hospitals_in_Europe" style="font-si
ze:114%;margin:0 4em"><a href="/wiki/List_of_hospitals_in_Europe" class="mw-redirect" title="List of hospitals
in Europe">List of hospitals in Europe </a></div></th></tr><tr><th scope="row" class="navbox-group" style="widt
h:1%">Sovereign states</th><td class="navbox-list-with-group navbox-list navbox-odd hlist" style="width:100%;pa
dding:0"><div style="padding:0 0.25em">\\n\'',
 'b\'<div class="navbox-styles nomobile"><link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:
r1061467846"/></div><div role="navigation" class="navbox authority-control" aria-labelledby="Authority_control:
_National_libraries_frameless&amp;#124;text-top&amp;#124;10px&amp;#124;alt=Edit_this_at_Wikidata&amp;#124;link=
https&amp;#58;//www.wikidata.org/wiki/Q5691262#identifiers&amp;#124;class=noprint&amp;#124;Edit_this_at_Wikidat
a" style="padding:3px"><table class="nowraplinks hlist navbox-inner" style="border-spacing:0;background:transpa
rent;color:inherit"><tbody><tr><th id="Authority_control:_National_libraries_frameless&amp;#124;text-top&amp;#1
24;10px&amp;#124;alt=Edit_this_at_Wikidata&amp;#124;link=https&amp;#58;//www.wikidata.org/wiki/Q5691262#identif
iers&amp;#124;class=noprint&amp;#124;Edit_this_at_Wikidata" scope="row" class="navbox-group" style="width:1%"><
a href="/wiki/Help:Authority_control" title="Help:Authority control">Authority control: National libraries</a>
<a href="https://www.wikidata.org/wiki/Q5691262#identifiers" title="Edit this at Wikidata"><img alt="Edit this
at Wikidata" src="//upload.wikimedia.org/wikipedia/en/thumb/8/8a/OOjs_UI_icon_edit-ltr-progressive.svg/10px-OOj
s_UI_icon_edit-ltr-progressive.svg.png" decoding="async" width="10" height="10" style="vertical-align: text-to
p" class="noprint" srcset="//upload.wikimedia.org/wikipedia/en/thumb/8/8a/OOjs_UI_icon_edit-ltr-progressive.sv
g/15px-OOjs_UI_icon_edit-ltr-progressive.svg.png 1.5x, //upload.wikimedia.org/wikipedia/en/thumb/8/8a/OOjs_UI_i
con_edit-ltr-progressive.svg/20px-OOjs_UI_icon_edit-ltr-progressive.svg.png 2x" data-file-width="20" data-file-
height="20" /></a></th><td class="navbox-list-with-group navbox-list navbox-odd" style="width:100%;padding:0"><
div style="padding:0 0.25em">\\n\'']
```


- We will scrape the first table, and use index 0 in `lst` to capture the first table name. Now, read the table using Beautiful Soup's `find` function. A simple option is to type the table name. You can simply select the name in `lst`, which in this case is "wikitable floatright sortable".

In [7]:
```
table=HCE.find('table', {'class', 'wikitable floatright sortable'})
```

In [8]:
```
type(table)
```

Out[8]:
```
bs4.element.Tag
```

- Alternatively, there is a way to automate this step by capturing the first data from the list and then stripping off the unnecessary characters like `^ " *`.

In [9]:
```
x=lst[0]
extr=re.findall('"([^"]*)"', x)
table=HCE.find('table', {'class', extr[0]})
```

In [10]:
```
type(table)
```

Out[10]:
```
bs4.element.Tag
```

- Now, it would be good to read the header and row names separately, so later we can easily make a DataFrame.

In [11]:
```
headers= [header.text for header in table.find_all('th')]
```

In [12]:
```
headers
```

Out[12]:
```
['WorldRank\n', 'EURank\n', 'Country\n', 'Life expectancyat birth (years)\n']
```

In [13]:
```
rows = []
for row in table.find_all('tr'):
    rows.append([val.text.encode('utf8').decode() for val in row.find_all('td')])
```

- Now, all elements, rows, and headers are available to build the DataFrame, which we will call `df1`.

In [14]:
```
df1 = pd.DataFrame(rows, columns=headers)
```

- Let's display first seven rows of the `df1`

In [15]:
```
df1.head(7)
```

Out[15]:

	WorldRank\n	EURank\n	Country\n	Life expectancyat birth (years)\n
0	None	None	None	None
1	5.\n	1.\n	Spain\n	83.4\n
2	6.\n	2.\n	Italy\n	83.4\n
3	11.\n	3.\n	Sweden\n	82.7\n
4	12.\n	4.\n	France\n	82.5\n
5	13.\n	5.\n	Malta\n	82.4\n
6	16.\n	6.\n	Ireland\n	82.1\n

Health Expenditure

Let's scrape health expenditure as well. These are data per capita, which means that expenditure was corrected for the number of habitants in a country.

- Just like we did for above web page (**Health Care Rankings for Different European Countries**), we have to repeat the same steps in this web page as well (**Health Expenditure**).

- Finally, we will be directed to the first table "wikitable sortable static" in this web page as well.

In [16]:
```
url = 'https://en.wikipedia.org/wiki/List_of_countries_by_total_health_expenditure_per_capita'
r = requests.get(url)
HEE = BeautifulSoup(r.text)
webpage = urllib2.urlopen(url)
htmlpage= webpage.readlines()
lst = []
for line in htmlpage:
    line = str(line).rstrip()
    if re.search('table class', line) :
        lst.append(line)
x=lst[1]
print(x)
extr=re.findall('"([^"]*)"', x)
table=HEE.find('table', {'class', extr[0]})
headers= [header.text for header in table.find_all('th')]
rows = []
for row in table.find_all('tr'):
    rows.append([val.text.encode('utf8').decode() for val in row.find_all('td')])
headers = [i.replace("\n", "") for i in headers]
df2 = pd.DataFrame(rows, columns=headers)
```

```
b'<table class="wikitable sortable static-row-numbers plainrowheaders srn-white-background" border="1" style="t
ext-align:right;">\n'
```

- Let's display the first five rows of the table "wikitable sortable static"

In [17]:
```
df2.head()
```

Out[17]:

	Country	2017	2018	2019
0	None	None	None	None
1	Australia *\n	4,711\n	4,965\n	5,187\n
2	Austria *\n	5,360\n	5,538\n	5,851\n
3	Belgium *\n	5,014\n	5,103\n	5,428\n
4	Canada *\n	5,155\n	5,287\n	5,418\n

Additional Preprocessing Steps

If we look at the DataFrame, we can see that there are still some issues that prohibit numeric computations.

- There are undesired characters ('\n')
- The undesired decimal format (,) should be removed
- There are cells with non-numeric characters ('x') that should be NAN

In [18]:
```
def preproc(dat):
    dat.dropna(axis=0, how='all', inplace=True)
    dat.columns = dat.columns.str.replace("\n", "")
    dat.replace(["\n"], [""], regex=True, inplace=True)
    dat.replace([r"\s\*$"], [""], regex=True, inplace=True)
    dat.replace([","], [""], regex=True, inplace=True)
    dat.replace(r"\b[a-zA-Z]\b", np.nan, regex=True, inplace=True)
    dat.replace([r"^\s"], [""], regex=True, inplace=True)
    dat = dat.apply(pd.to_numeric, errors='ignore')
    return(dat)
```

In [19]:
```
df1 = preproc(df1)
df2 = preproc(df2)
```

- Apparently, after this preprocessing, there are some NANs.

In [20]:
```
print(df1.isnull().sum().sum())
print(df2.isnull().sum().sum())
```

```
0
0
```

- Now we display where the NANs occur. In fact, when we check the original table, we can see that Cyprus has values "x", which were in our preproc function changed to NANs (https://en.wikipedia.org/wiki/Healthcare_in_Europe).

In [21]:
```
df1[df1.isnull().any(axis=1)]
```

Out[21]:

	WorldRank	EURank	Country	Life expectancyat birth (years)

Now we remove the NANs.

In [22]:
```
df1.dropna(axis=0, how='any', inplace=True)
```

At this point we inspect the data types:

In [23]:
```
df1.dtypes
```

Out[23]:
```
WorldRank                             float64
EURank                                float64
Country                                object
Life expectancyat birth (years)       float64
dtype: object
```

In [24]:
```
df2.dtypes
```

Out[24]:
```
Country     object
2017         int64
2018         int64
2019         int64
dtype: object
```

The column names are a bit long, so it would be good to use shorter names.

In [25]:
```
df1.columns = ['WorldRank', 'EURank', 'Country', 'Life expectancy in (years)']
df2.columns = ['Country', '2017', '2018', '2019']
```

Analyzing Final Tables

In [26]:
```
df1.head()
```

Out[26]:

	WorldRank	EURank	Country	Life expectancy in (years)
1	5.0	1.0	Spain	83.4
2	6.0	2.0	Italy	83.4
3	11.0	3.0	Sweden	82.7
4	12.0	4.0	France	82.5
5	13.0	5.0	Malta	82.4

In [27]:
```
df2.head()
```

Out[27]:

	Country	2017	2018	2019
1	Australia	4711	4965	5187
2	Austria	5360	5538	5851
3	Belgium	5014	5103	5428
4	Canada	5155	5287	5418
5	Chile	2030	2126	2159

Merging Different Data

It should be clear from this example that web scraping can be important to quickly grasp data.

Web scraping may be particularly useful when you need to automate data processing:

- Webdata change regularly and need to be stored repeatedly.

- A large number of data sources, for example, tables, need to be loaded and merged.

Let us elaborate on the last point a bit more. If the two tables that we just scraped need to be merged, it can be done in Python. For example, if we want to merge on the column "Country", we would use the following code (we use the `.head()` function to limit the output).

In [28]:
```
pd.merge(df1, df2, how='left', on='Country').head()
```

Out[28]:

	WorldRank	EURank	Country	Life expectancy in (years)	2017	2018	2019
0	5.0	1.0	Spain	83.4	3322.0	3430.0	3616.0
1	6.0	2.0	Italy	83.4	3399.0	3485.0	3649.0
2	11.0	3.0	Sweden	82.7	5318.0	5434.0	5782.0
3	12.0	4.0	France	82.5	5057.0	5154.0	5376.0
4	13.0	5.0	Malta	82.4	NaN	NaN	NaN

Note: In this lesson, we saw the use of the data wrangling and web scraping methods, but in the next lesson we are going to use one of these methods as a sub component of "Feature Engineering".