



Published in Analytics Vidhya



Vandany Lubis (Danny)

Follow

Oct 4, 2021 · 6 min read · Listen



Save



# How to Scrape Table from Website using Python

Previously, we have learned about the basics of scraping data on the sites, but did you know that we are also able to scrape data in a tabular form? If you often surfing the internet you probably notice that many websites containing data tables. For example flight schedule, product features, services differentiation, tv schedule, ranking, and many more.

Similarweb

Rankings Data Solutions Pricing Resources Explore Login GET STARTED

Analyze any Website or App

Home > Top Websites

# Top Websites Ranking

Top Websites Ranking for all categories in the world

All categories Worldwide Go

Showing up to 50 websites. Upgrade to see the full list. Last updated: August 01, 2021

Rank	Website	Category	Change	Avg. Visit Duration	Pages / Visit	Bounce Rate
1	google.com	Computers, Electronics and Technology > Search Engines	-	00:11:07	8.65	27.97%
2	youtube.com	Arts and Entertainment > TV Movies and Streaming	-	00:21:48	11.40	20.96%
3	facebook.com	Computers, Electronics and Technology > Social Networks and Online Communities	-	00:10:08	8.70	32.28%
4	twitter.com	Computers, Electronics and Technology > Social Networks and Online Communities	-	00:10:44	11.50	28.81%
5	instagram.com	Computers, Electronics and Technology > Social Networks and Online Communities	-	00:07:48	11.18	35.13%
6	baidu.com	Computers, Electronics and Technology > Search Engines	-	00:06:07	7.99	20.32%
7	wikipedia.org	Reference > Encyclopedias and Dictionaries	-	00:02:47	3.86	56.70%
8	yandex.ru	Computers, Electronics and Technology > Search Engines	-	00:11:01	8.61	22.57%
9	yahoo.com	News and Media	-	00:07:31	5.72	35.08%

Fig.1 Data Table from Similarweb.com

In some cases maybe it is easy to copy and paste the table into excel without doing coding, but as data scientists, we need to understand that in the future we will work with bigger data where copy-paste method is not really efficient. So now I will show you how to scrape a data table from a website.

## STEP 1. INSTALLING LIBRARIES

First of all, we need these required libraries installed in our environment:

1. BeautifulSoup4
2. Requests
3. pandas
4. lxml

I recommend you to read the previous tutorial about [how to scrape data from the website for beginners](#) if you having trouble in this step.

## STEP 2. IMPORT REQUIRED LIBRARIES

So after the important libraries are already installed the next thing we can do is open Spyder. The reason why I using Spyder is that it is more comfortable to use for projects, but it is okay if you have your own preferences.

Okay, once we open the Spyder the next thing we can do is importing the required library:

```
# Import libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

## STEP 3. SELECT PAGE

In this project, we will scrape the covid data table from [Worldometers](#). Same as the previous tutorial this website is also considered easier to understand for beginners since it is made with HTML.

Country, other	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious, Critical	Total Cases 1M pop	Deaths 1M pop	Total Tests	Total 1M pop	Population
World	235,715,604	+35,710	4,815,003	+329	212,521,602	+24,329	18,321,109	57,005	30,241	517.0			
1 USA	44,518,437		719,933		33,038,147		9,880,357	19,236	133,514	2,169	944,742,556	1,933,643	333,41
2 India	33,834,243		448,029		33,113,544		271,570	8,944	24,219	321	573,280,724	410,347	1,297,0
3 Brazil	21,486,121		597,986		20,442,553		427,482	8,318	100,186	2,788	83,775,168	297,388	214,41
4 UK	7,909,680		136,953		6,417,860		1,346,067	816	115,620	2,504	304,621,501	4,457,898	63,21
5 Russia	7,536,536		209,918		6,725,965		550,603	2,360	61,958	1,438	193,200,000	1,323,170	148,0
6 Turkey	7,238,267		84,661		6,686,246		487,369	633	84,681	795	87,366,730	1,022,113	85,4
7 France	7,027,559		116,798		6,750,735		116,928	1,355	107,358	1,784	143,234,831	2,188,333	66,41
8 Iran	6,624,129		121,109		6,095,627		407,392	8,758	65,903	1,419	32,306,789	378,685	85,24
9 Argentina	6,269,738		115,245		6,123,173		21,320	1,052	115,053	2,621	24,262,818	530,614	43,7
10 Colombia	4,962,554		126,401		4,804,367		31,295	542	95,233	2,461	26,670,579	467,882	51,51
11 Spain	4,981,129		86,493		4,755,738		116,927	675	106,058	1,848	54,557,887	1,380,105	46,7
12 Italy	4,692,534		131,031		4,658,036		92,967	431	77,581	2,171	93,405,967	1,547,849	60,31
13 Germany	4,285,494		94,296		4,024,300		141,893	1,338	50,648	1,121	73,348,901	871,955	84,12
14 Indonesia	4,219,284		142,173		4,044,235		32,678		15,224	513	39,888,567	143,852	277,1
15 Mexico	3,591,060	+2,980	278,601	+211	3,035,521	+6,767	387,638	4,798	28,186	2,134	10,737,104	82,194	139,81
16 Poland	2,910,866		75,695		2,864,366		170,815	168	77,016	2,503	21,015,216	556,038	37,71
17 South Africa	2,906,422		87,780		2,779,906		38,735	546	48,243	1,457	17,783,920	296,190	60,21
18 Philippines	2,693,399		38,768		2,442,623		112,008	3,170	23,278	348	21,296,983	191,167	111,41

Fig.2 COVID Data Table

## STEP 4. REQUEST PERMISSION

After we select what page we want to scrape, now we can copy the page's URL and use **requests** to ask permission from the hosting server that we want to fetch data from their site.

```
# Create an URL object
url = 'https://www.worldometers.info/coronavirus/'

# Create object page
page = requests.get(url)
```

If the output is **<Response [200]>** that means the server allows us to collect data from their website. After we have permission from the server now we need to parse the HTML code using **lxml** to make it more readable.

```
# parser-lxml = Change html to Python friendly format
# Obtain page's information
soup = BeautifulSoup(page.text, 'lxml')
soup
```

## STEP 5. INSPECT TABLE ELEMENT

In the [previous article](#), we have learned how to inspect every element in a website page so I assume you have understood how to work with it. To obtain the information we need to inspect the location of the table first.



Fig.3 Table location

As we can see from the picture above that table is located under the `<table>` tag and `id = 'main_table_countries_today'`. After we have found the location of the table now we can define the variable. In this section, we define the table as `'table1'`.

```
# Obtain information from tag <table>
table1 = soup.find('table', id='main_table_countries_today')
table1
```

## STEP 6. CREATE A COLUMN LIST

After the `table1` has been created now the next thing we can do is inspecting the location of each column. If we look into each column we notice that they have the same characteristic.



Fig.4 Columns

In the picture above we can see the characteristic of each column is they are located under the tag **<th>**.

After we found the tag of each column the next thing we can do is create a for loop to fill an empty list with each column. We define an empty list as **headers**

```
# Obtain every title of columns with tag <th>
headers = []
for i in table1.find_all('th'):
    title = i.text
    headers.append(title)
```

Output:

headers - List (22 elements)

Indi	Type	Size	
0	str	1	#
1	str	13	Country,Other
2	str	10	TotalCases
3	str	8	NewCases
4	str	11	TotalDeaths
5	str	9	NewDeaths
6	str	14	TotalRecovered
7	str	12	NewRecovered
8	str	11	ActiveCases
9	str	16	Serious,Critical
10	str	16	Tot Cases/1M pop
11	str	13	Deaths/1M pop
12	str	10	TotalTests
13	str	14	Tests/ 1M pop
14	str	10	Population
15	str	9	Continent
16	str	17	1 Caseevery X ppl
17	str	18	1 Deathevery X ppl
18	str	17	1 Testevery X ppl
19	str	16	New Cases/1M pop
20	str	17	New Deaths/1M pop

Fig.5 Headers

After the list is successfully filled with columns, now we can check again. Let's look at index 13 we find wrapped text. Wrapped text like this could be a problem when we want to make a data frame from it, so we need to convert it into one-line text.

```
# Convert wrapped text in column 13 into one line text
headers[13] = 'Tests/1M pop'
```

Output:



headers - List (22 elements)

Indi	Type	Size	
0	str	1	#
1	str	13	Country,Other
2	str	10	TotalCases
3	str	8	NewCases
4	str	11	TotalDeaths
5	str	9	NewDeaths
6	str	14	TotalRecovered
7	str	12	NewRecovered
8	str	11	ActiveCases
9	str	16	Serious,Critical
10	str	16	Tot Cases/1M pop
11	str	13	Deaths/1M pop
12	str	10	TotalTests
13	str	12	Tests/1M pop
14	str	10	Population
15	str	9	Continent
16	str	17	1 Caseevery X ppl
17	str	18	1 Deathevery X ppl
18	str	17	1 Testevery X ppl
19	str	16	New Cases/1M pop
20	str	17	New Deaths/1M pop

Fig.6 Index 13 have been fixed

## STEP 7. CREATE A DATA FRAME

After the list of columns is made the next thing we can do is create a dataframe. We define the dataframe as **mydata**.

```
# Create a dataframe
mydata = pd.DataFrame(columns = headers)
```

## STEP 8. CREATE A FOR LOOP TO FILL DATAFRAME



After the dataframe is created now we can fill it with items in each column. Before we create a for loop, we need to identify the location of the row and item column first.



Country # Other	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious/ Critical	Tot Cases/ 1M pop	Deaths/ 1M pop	Trend
World	335,827,306	+123,119	4,818,046	+2,427	212,737,701	+168,885	18,271,821	86,645	30,254	616.1	
1 USA	44,518,437	+469	719,933		33,938,147		9,860,367	19,236	133,514	2,159	6
2 India	33,834,702	+469	449,029		33,121,247	+7,603	264,426	8,944	24,219	321	5
3 Brazil	21,468,121		597,986		20,442,653		427,482	8,318	100,108	2,788	1
4 UK	7,900,880		138,953		6,417,880		1,349,067	815	115,825	2,004	31
5 Russia	7,612,317	+26,781	210,801	+883	6,740,491	+14,826	661,025	2,300	62,135	1,444	11
6 Turkey	7,238,267		54,561		5,885,246		487,360	633	84,681	736	1
7 France	7,027,059		116,796		6,730,335		119,928	1,355	107,358	1,784	1

Fig.7 Location of row



Country # Other	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious/ Critical	Tot Cases/ 1M pop	Deaths/ 1M pop	Trend
World	335,827,306	+123,119	4,818,046	+2,427	212,737,701	+168,885	18,271,821	86,645	30,254	616.1	
1 USA	44,518,437	+469	719,933		33,938,147		9,860,367	19,236	133,514	2,159	6
2 India	33,834,702	+469	449,029		33,121,247	+7,603	264,426	8,944	24,219	321	5
3 Brazil	21,468,121		597,986		20,442,653		427,482	8,318	100,108	2,788	1
4 UK	7,900,880		138,953		6,417,880		1,349,067	815	115,825	2,004	31
5 Russia	7,612,317	+26,781	210,801	+883	6,740,491	+14,826	661,025	2,300	62,135	1,444	11
6 Turkey	7,238,267		54,561		5,885,246		487,360	633	84,681	736	1
7 France	7,027,059		116,796		6,730,335		119,928	1,355	107,358	1,784	1

Fig.8 Location of item column

From the pictures above we can identify that the row is located under tag `<tr>` and items are located under tag `<td>`. This is applied to all rows and items within the table.

After we found the tags now we can create a **for loop**.

```
# Create a for loop to fill mydata
for j in table1.find_all('tr')[1:]:
    row_data = j.find_all('td')
    row = [i.text for i in row_data]
    length = len(mydata)
    mydata.loc[length] = row
```

Open in app ↗

Sign up

Sign In



	Continent	Country	GDP	Population	Area	Density	GDP per Capita	Population per Capita	Area per Capita	Density per Capita	GDP per Capita per Area	Population per Capita per Area	Area per Capita per Density	Density per Capita per Density	GDP per Capita per Density per Area
0	North Amer.	USA	21,491,000	318,900,000	9,833,517	32.5	69,300	318,900,000	9,833,517	32.5	69,300	318,900,000	9,833,517	32.5	69,300
1	Asia	China	14,700,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000
2	South Amer.	Brazil	1,740,000	215,000,000	8,511,965	25.3	20,000	215,000,000	8,511,965	25.3	20,000	215,000,000	8,511,965	25.3	20,000
3	Europe	Germany	3,840,000	83,000,000	357,021	232.0	110,000	83,000,000	357,021	232.0	110,000	83,000,000	357,021	232.0	110,000
4	Africa	Nigeria	510,000	206,000,000	923,768	223.0	2,300	206,000,000	923,768	223.0	2,300	206,000,000	923,768	223.0	2,300
5	Oceania	Australia	1,360,000	25,000,000	7,692,000	3.3	17,000	25,000,000	7,692,000	3.3	17,000	25,000,000	7,692,000	3.3	17,000
6	Asia	India	2,800,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000
7	Asia	Japan	5,400,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000
8	Asia	South Korea	1,600,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000
9	Asia	Thailand	500,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000
10	Asia	Malaysia	1,000,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000
11	Asia	Indonesia	2,000,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000
12	Asia	Philippines	350,000	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500
13	Asia	Vietnam	3,500,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000
14	Asia	Turkey	7,200,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000
15	Asia	France	2,800,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000
16	Asia	Iran	1,800,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000
17	Asia	Argentina	500,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000
18	Asia	Colombia	400,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000
19	Asia	India	2,800,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000
20	Asia	China	14,700,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000
21	Asia	Japan	5,400,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000
22	Asia	South Korea	1,600,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000
23	Asia	Thailand	500,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000
24	Asia	Malaysia	1,000,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000
25	Asia	Indonesia	2,000,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000
26	Asia	Philippines	350,000	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500
27	Asia	Vietnam	3,500,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000
28	Asia	Turkey	7,200,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000
29	Asia	France	2,800,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000
30	Asia	Iran	1,800,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000
31	Asia	Argentina	500,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000
32	Asia	Colombia	400,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000
33	Asia	India	2,800,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000
34	Asia	China	14,700,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000
35	Asia	Japan	5,400,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000
36	Asia	South Korea	1,600,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000
37	Asia	Thailand	500,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000
38	Asia	Malaysia	1,000,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000
39	Asia	Indonesia	2,000,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000
40	Asia	Philippines	350,000	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500
41	Asia	Vietnam	3,500,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000
42	Asia	Turkey	7,200,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000
43	Asia	France	2,800,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000
44	Asia	Iran	1,800,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000
45	Asia	Argentina	500,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000	45,000,000	2,780,167	16.0	12,000
46	Asia	Colombia	400,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000	50,000,000	1,104,160	45.0	8,000
47	Asia	India	2,800,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000	1,380,000,000	3,287,263	419.0	21,000
48	Asia	China	14,700,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000	1,400,000,000	9,600,000	145.0	15,000
49	Asia	Japan	5,400,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000	126,000,000	377,915	331.0	42,000
50	Asia	South Korea	1,600,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000	51,000,000	100,339	508.0	32,000
51	Asia	Thailand	500,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000	66,000,000	513,120	128.0	12,000
52	Asia	Malaysia	1,000,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000	32,000,000	330,845	96.0	30,000
53	Asia	Indonesia	2,000,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000	270,000,000	1,919,343	141.0	14,000
54	Asia	Philippines	350,000	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500	109,000,000	343,686	317.0	3,500
55	Asia	Vietnam	3,500,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000	96,000,000	331,212	289.0	35,000
56	Asia	Turkey	7,200,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000	84,000,000	783,562	107.0	80,000
57	Asia	France	2,800,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000	68,000,000	643,801	105.0	28,000
58	Asia	Iran	1,800,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,000	50.0	22,000	82,000,000	1,628,		

Index	Country/State	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	NewRecovered	ActiveCases	SeriousCritical	TotalTests/100k	Deaths/100k	TotalFacts	NewFacts	Possibilities	Continent	1.Expected 30d	1.NewDeaths 30d	1.TotalDeaths 30d	NewCases 1M
1	Mexico	137,485,...	+137,485	2,818,139	+1,878	135,750,340	+182,489	18,779,7...	85,947	36,385	833.1	...	...	...	...	...	...	...	...
2	USA	48,518,...	...	713,935	...	39,836,147	...	8,686,397	16,218	113,514	1.179	844,346	1,933,641	333,455	North America	...	...	...	...
3	India	33,841,...	+11,343	449,214	+183	33,391,303	+10,848	33,399	8,344	34,217	7.11	374,251	+11,857	1,397,6...	Asia	...	...	...	...
4	Brazil	31,408,...	...	307,881	...	18,441,353	...	427,481	8,316	189,296	2.789	81,376,...	187,308	134,454	South America	...	...	...	...
5	UK	7,089,...	...	218,431	...	6,871,838	...	1,296,267	833	113,838	6.989	89,821,...	+817,838	86,212,...	Europe	...	...	...	...
6	Russia	7,030,...	+70,783	218,881	+883	6,748,493	+58,518	681,935	2,388	91,319	1.849	281,786	1,934,994	184,811	Europe	...	...	...	...
7	Turkey	7,018,...	...	84,884	...	6,889,268	...	821,888	833	88,881	184	87,884,...	1,902,317	86,878,...	Asia	...	...	...	...
8	France	7,017,...	...	114,738	...	6,708,130	...	114,928	1,381	187,356	1.735	141,231,...	1,128,581	84,811,...	Europe	...	...	...	...
9	Iran	6,635,...	+16,887	111,347	+258	6,518,796	+18,168	481,581	5,727	86,856	1.817	81,839,...	187,218	86,348,...	Asia	...	...	...	...
10	Argentina	6,289,7...	...	115,345	...	5,113,171	...	11,318	1,852	118,855	1.812	34,131,...	558,814	47,775,...	South America	...	...	...	...
11	Colombia	6,282,...	...	111,481	...	4,968,357	...	11,298	343	36,218	1.451	25,678,...	487,852	11,382,...	South America	...	...	...	...
12	Canada	6,081,...	...	86,843	...	4,723,738	...	116,947	473	188,458	1.844	86,017,...	1,388,488	84,777,...	North America	...	...	...	...
13	Spain	6,081,...	...	111,831	...	4,968,356	...	11,297	411	77,184	1.176	82,488,...	1,547,649	80,509,...	Europe	...	...	...	...
14	Germany	6,189,...	...	84,138	...	5,812,888	+8,188	111,188	1,136	88,848	1.111	72,188,...	871,818	84,128,...	Europe	...	...	...	...
15	Indonesia	6,282,...	+833	115,345	+88	4,968,357	+8,188	11,298	343	15,217	1.11	86,889,...	184,414	77,788,...	Asia	...	...	...	...
16	Pakistan	6,081,...	+2,888	778,881	+211	5,303,121	+4,781	887,888	8,788	28,188	1.138	18,777,...	82,188	18,848,...	South Asia	...	...	...	...
17	Poland	2,931,...	+888	78,888	...	2,853,181	+727	118,711	118	77,888	2.888	21,888,...	888,888	17,788,...	Europe	...	...	...	...
18	South Africa	2,981,...	...	87,788	...	2,779,396	...	28,756	848	44,345	1.487	17,788,...	187,188	86,345,...	Africa	...	...	...	...
19	Philippines	2,644,...	+12,746	36,348	+83	2,453,851	+18,523	188,188	2,218	22,354	548	31,355,...	111,888	111,448	Asia	...	...	...	...
20	Vietnam	2,489,...	+4,881	36,348	+134	2,389,511	+4,116	136,388	117	36,637	1.111	11,888,...	188,754	41,443,...	Europe	...	...	...	...
21	Malaysia	2,177,...	...	36,348	...	2,189,188	...	218,727	848	88,187	811	36,518,...	817,818	31,888,...	Asia	...	...	...	...
22	Peru	2,131,...	...	188,888	...	1,942,188	...	188,888	848	88,888	1.848	18,888,...	111,888	11,448	South America	...	...	...	...

Fig.10 Final dataframe

## STEP 10. EXPORT TO CSV AND TRY TO RUN IT

After the dataframe is finished so the next thing we can do is export it into CSV format.

```
# Export to csv
mydata.to_csv('covid_data.csv', index=False)

# Try to read csv
mydata2 = pd.read_csv('covid_data.csv')
```

Since the dataframe has been exported into CSV now we can use the data to build a data science project, put it into machine learning, data analysis project, and so on.

That's the tutorial I gave, hopefully, it will be useful for you guys especially for you who are learning web scraping. See you again in the next project.

[Web Scraping](#)
[Data Science](#)
[Data Engineering](#)
[Beautifulsoup](#)

## Sign up for Analytics Vidhya News Bytes

By Analytics Vidhya

Latest news from Analytics Vidhya on our Hackathons and some of our best articles! [Take a look.](#)

Your email



Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

