Practice Project: GDP Data extraction and processing

Estimated time needed: **30** minutes

Introduction

In this practice project, you will put the skills acquired through the course to use. You will extract data from a website using webscraping and reqeust APIs process it using Pandas and Numpy libraries.

Project Scenario:

An international firm that is looking to expand its business in different countries across the world has recruited you. You have been hired as a junior Data Engineer and are tasked with creating a script that can extract the list of the top 10 largest economies of the world in descending order of their GDPs in Billion USD (rounded to 2 decimal places), as logged by the International Monetary Fund (IMF).

The required data seems to be available on the URL mentioned below:

URL: https://web.archive.org/web/20230902185326/https://en.wikipedia.org/wiki/List_of_countries_by_GDP_%28nominal%29

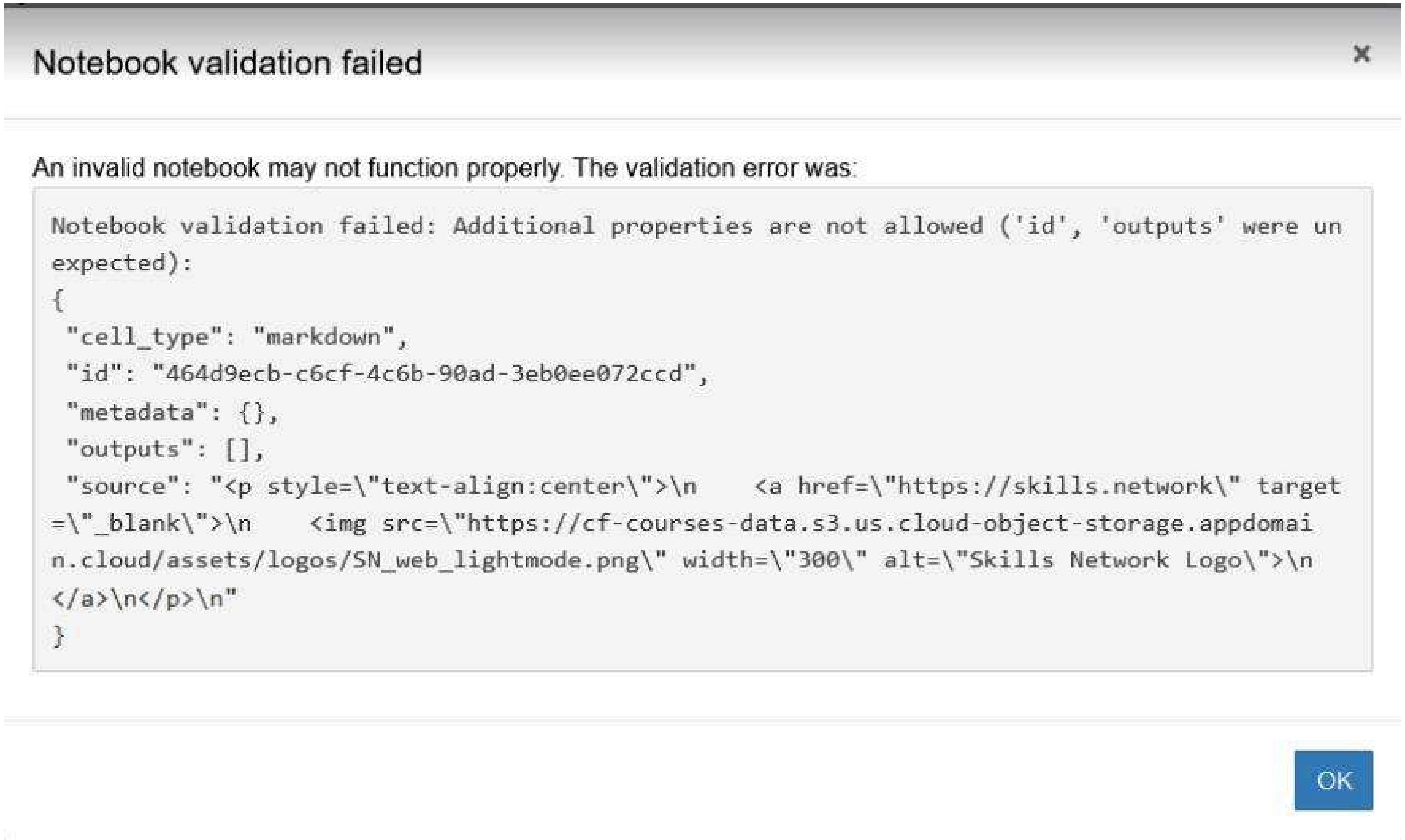
Objectives

After completing this lab you will be able to:

- Use Webscraping to extract required information from a website.
- Use Pandas to load and process the tabular data as a dataframe.
- Use Numpy to manipulate the information contatined in the dataframe.
- Load the updated dataframe to CSV file.

Dislcaimer

If you are using a downloaded version of this notebook on your local machine, you may encounter a warning message as shown in the screenshot below.



Notebook validation failed

An invalid notebook may not function properly. The validation error was:

```
Notebook validation failed: Additional properties are not allowed ('id', 'outputs' were un
expected):
{
 "cell_type": "markdown",
 "id": "464d9ecb-c6cf-4c6b-90ad-3eb0ee072ccd",
 "metadata": {},
 "outputs": [],
 "source": "<p style=\"text-align:center\">\n    <a href=\"https://skills.network\" target
=\"_blank\">\n    <img src=\"https://cf-courses-data.s3.us.cloud-object-storage.appdomai
n.cloud/assets/logos/SN_web_lightmode.png\" width=\"300\" alt=\"Skills Network Logo\">\n
</a>\n</p>\n"
}
```

This does not affect the execution of your codes in any way and can be simply ignored.

Setup

For this lab, we will be using the following libraries:

- `pandas` for managing the data.
- `numpy` for mathematical operations.

In [1]:
```
#Install required packages
!pip install pandas numpy
!pip install lxml
```

```
Collecting pandas
  Downloading pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (89 kB)
Collecting numpy
  Downloading numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (62 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.12/site-packages (from pandas) (2.9.
0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-packages (from pandas) (2024.2)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pand
as) (1.17.0)
Downloading pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.7/12.7 MB 161.7 MB/s eta 0:00:00
Downloading numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.1/16.1 MB 166.0 MB/s eta 0:00:00
Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-2.2.4 pandas-2.2.3 tzdata-2025.1
Collecting lxml
  Downloading lxml-5.3.1-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (3.7 kB)
Downloading lxml-5.3.1-cp312-cp312-manylinux_2_28_x86_64.whl (5.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.0/5.0 MB 51.9 MB/s eta 0:00:00
Installing collected packages: lxml
Successfully installed lxml-5.3.1
```

Importing Required Libraries

We recommend you import all required libraries in one place (here):

In [2]:

```
import numpy as np
import pandas as pd

# You can also use this section to suppress warnings generated by your code:
def warn(*args, **kwargs):
    pass
import warnings
warnings.warn = warn
warnings.filterwarnings('ignore')
```

Exercises

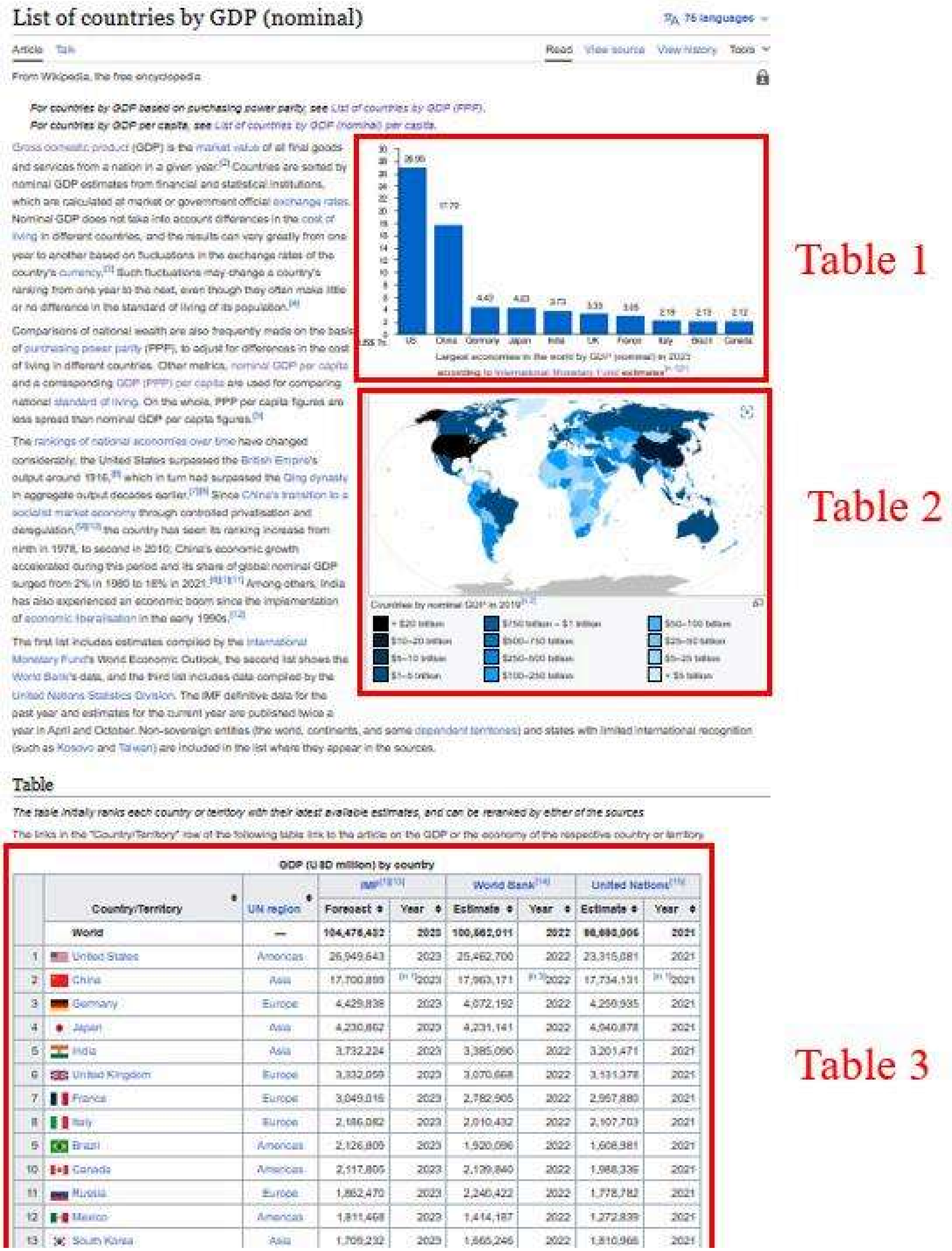
Exercise 1

Extract the required GDP data from the given URL using Web Scraping.

In [3]:

```
URL="https://web.archive.org/web/20230902185326/https://en.wikipedia.org/wiki/List_of_countries_by_GDP_%28nominal%29"
```

You can use Pandas library to extract the required table directly as a DataFrame. Note that the required table is the third one on the website, as shown in the image below.



List of countries by GDP (nominal)

From Wikipedia, the free encyclopedia

For countries by GDP based on purchasing power parity, see List of countries by GDP (PPP).
For countries by GDP per capita, see List of countries by GDP (nominal) per capita.

Gross domestic product (GDP) is the market value of all final goods and services from a nation in a given year.[2] Countries are sorted by nominal GDP estimates from financial and statistical institutions, which are calculated at market or government official exchange rates. Nominal GDP does not take into account differences in the cost of living in different countries, and the results can vary greatly from one year to another based on fluctuations in the exchange rates of the country's currency.[3] Such fluctuations may change a country's ranking from one year to the next, even though they often make little or no difference in the standard of living of its population.[4]

Comparisons of national wealth are also frequently made on the basis of purchasing power parity (PPP), to adjust for differences in the cost of living in different countries. Other metrics, nominal GDP per capita and a corresponding GDP (PPP) per capita are used for comparing national standard of living. On the whole, PPP per capita figures are less spread than nominal GDP per capita figures.[5]

The rankings of national economies over time have changed considerably; the United States surpassed the British Empire's output around 1916,[6] which in turn had surpassed the Qing dynasty in aggregate output decades earlier.[7][8] Since China's transition to a socialist market economy through controlled privatisation and deregulation,[9][10] the country has seen its ranking increase from ninth in 1978, to second in 2010; China's economic growth accelerated during this period and its share of global nominal GDP surged from 2% in 1980 to 18% in 2021.[9][11] Among others, India has also experienced an economic boom since the implementation of economic liberalisation in the early 1990s.[12]

The first list includes estimates compiled by the International Monetary Fund's World Economic Outlook, the second list shows the World Bank's data, and the third list includes data compiled by the United Nations Statistics Division. The IMF definitive data for the past year and estimates for the current year are published twice a year in April and October. Non-sovereign entities (the world, continents, and some dependent territories) and states with limited international recognition (such as Kosovo and Taiwan) are included in the list where they appear in the sources.

Largest economies in the world by GDP (nominal) in 2023 according to International Monetary Fund estimates[n 1][1]

Countries by nominal GDP in 2019[1 2]

- ■ > $20 trillion
- ■ $10–20 trillion
- ■ $5–10 trillion
- ■ $1–5 trillion
- ■ $750 billion – $1 trillion
- ■ $500–750 billion
- ■ $250–500 billion
- ■ $100–250 billion
- ■ $50–100 billion
- ■ $25–50 billion
- ■ $5–25 billion
- □ < $5 billion

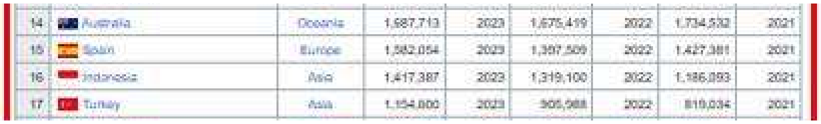
Table

The table initially ranks each country or territory with their latest available estimates, and can be reranked by either of the sources.

The links in the "Country/Territory" row of the following table link to the article on the GDP or the economy of the respective country or territory.

GDP (USD million) by country

	Country/Territory	UN region	IMF[1][13]		World Bank[14]		United Nations[15]	
			Forecast	Year	Estimate	Year	Estimate	Year
	World	—	104,476,432	2023	100,562,011	2022	96,698,005	2021
1	United States	Americas	26,949,643	2023	25,462,700	2022	23,315,081	2021
2	China	Asia	17,700,899	[n 1]2023	17,963,171	[n 3]2022	17,734,131	[n 1]2021
3	Germany	Europe	4,429,838	2023	4,072,192	2022	4,259,935	2021
4	Japan	Asia	4,230,862	2023	4,231,141	2022	4,940,878	2021
5	India	Asia	3,732,224	2023	3,385,090	2022	3,201,471	2021
6	United Kingdom	Europe	3,332,059	2023	3,070,668	2022	3,131,378	2021
7	France	Europe	3,049,016	2023	2,782,905	2022	2,957,880	2021
8	Italy	Europe	2,186,082	2023	2,010,432	2022	2,107,703	2021
9	Brazil	Americas	2,126,809	2023	1,920,096	2022	1,608,981	2021
10	Canada	Americas	2,117,805	2023	2,139,840	2022	1,988,336	2021
11	Russia	Europe	1,862,470	2023	2,240,422	2022	1,778,782	2021
12	Mexico	Americas	1,811,468	2023	1,414,187	2022	1,272,839	2021
13	South Korea	Asia	1,709,232	2023	1,665,246	2022	1,810,966	2021



14	Australia	Oceania	1,687,713	2023	1,675,419	2022	1,734,532	2021
15	Spain	Europe	1,582,054	2023	1,397,509	2022	1,427,381	2021
16	Indonesia	Asia	1,417,387	2023	1,319,100	2022	1,186,093	2021
17	Turkey	Asia	1,154,600	2023	905,988	2022	819,034	2021

In [4]:

```
# Extract tables from webpage using Pandas. Retain table number 3 as the required dataframe.

# Extract tables from webpage using Pandas. Retain table number 3 as the required dataframe.
tables = pd.read_html(URL)
df = tables[3]

# Replace the column headers with column numbers
df.columns = range(df.shape[1])

# Retain columns with index 0 and 2 (name of country and value of GDP quoted by IMF)
df = df[[0,2]]

# Retain the Rows with index 1 to 10, indicating the top 10 economies of the world.
df = df.iloc[1:11,:]

# Assign column names as "Country" and "GDP (Million USD)"
df.columns = ['Country','GDP (Million USD)']
```

▶ Click here for Solution

Exercise 2

Modify the GDP column of the DataFrame, converting the value available in Million USD to Billion USD. Use the `round()` method of Numpy library to round the value to 2 decimal places. Modify the header of the DataFrame to `GDP (Billion USD)`.

In [5]:

```
# Change the data type of the 'GDP (Million USD)' column to integer. Use astype() method.
df['GDP (Million USD)'] = df['GDP (Million USD)'].astype(int)

# Convert the GDP value in Million USD to Billion USD
df[['GDP (Million USD)']] = df[['GDP (Million USD)']]/1000

# Use numpy.round() method to round the value to 2 decimal places.
df[['GDP (Million USD)']] = np.round(df[['GDP (Million USD)']], 2)
```

```
# Rename the column header from 'GDP (Million USD)' to 'GDP (Billion USD)'
df.rename(columns = {'GDP (Million USD)' : 'GDP (Billion USD)'})
```

Out[5]:

	Country	GDP (Billion USD)
1	United States	26854.60
2	China	19373.59
3	Japan	4409.74
4	Germany	4308.85
5	India	3736.88
6	United Kingdom	3158.94
7	France	2923.49
8	Italy	2169.74
9	Canada	2089.67
10	Brazil	2081.24

▶ Click here for solution

Exercise 3

Load the DataFrame to the CSV file named "Largest_economies.csv"

In [6]:

```
# Load the DataFrame to the CSV file named "Largest_economies.csv"
# Load the DataFrame to the CSV file named "Largest_economies.csv"
df.to_csv('./Largest_economies.csv')
```

▶ Click here for Solution

Congratulations! You have completed the lab.

Authors

Abhishek Gagneja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-11-10	0.1	Abhishek Gagneja	Created initial version