Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Engineering

Name of the Student: _DIKSHA JADHAV_____ Roll Number: _C206___

SAP ID: _60004240043_____ Class: _C3_____ Division:_C3_____

Batch:__DV2_____

Subject: Data Visualization____

EXPERIMENT NO: 04

AIM: Data Collection and Wrangling in Python. (CO4)

IMPLEMENTATION/SOURCE CODE WITH OUTPUT:

```
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
```

[1] ✓ 5.9s — Python

Generate | + Code | + Markdown

(A) Loading a CSV File

```
df = pd.read_csv("dataset_new.csv")
df.head()
```

[ ] — Python

	Country	Cases	Deaths	Recovered	Date
0	France	282587	32532	173231	2024-10-20
1	China	257357	5504	490323	2024-10-02
2	Indonesia	665376	13709	462288	2024-04-21
3	UK	176639	12831	55408	2024-10-25
4	Turkey	427238	43009	806343	2024-05-21

```
df.describe()
```

Launchpad | ⊗ 0 ⚠ 7 | Live Share | Ln 1, Col 14 | CRLF | Cell 18 of 18 | Go Live

```
df.describe()
```

[ ] — Python

	Cases	Deaths	Recovered
count	200.000000	200.000000	200.000000
mean	532050.365000	25256.935000	448650.015000
std	266370.531774	13575.763109	250875.513199
min	61111.000000	1524.000000	47718.000000
25%	308932.750000	14148.750000	245365.250000
50%	542032.000000	23521.000000	434767.500000
75%	762503.750000	36682.000000	644089.000000
max	993999.000000	49939.000000	899804.000000

(B) Fetching Live Data from an API

```
api_key = "7a9ad05412f1f2af9ab39b67fa9388c0"
city = "Mumbai"
url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"

response = requests.get(url)
```

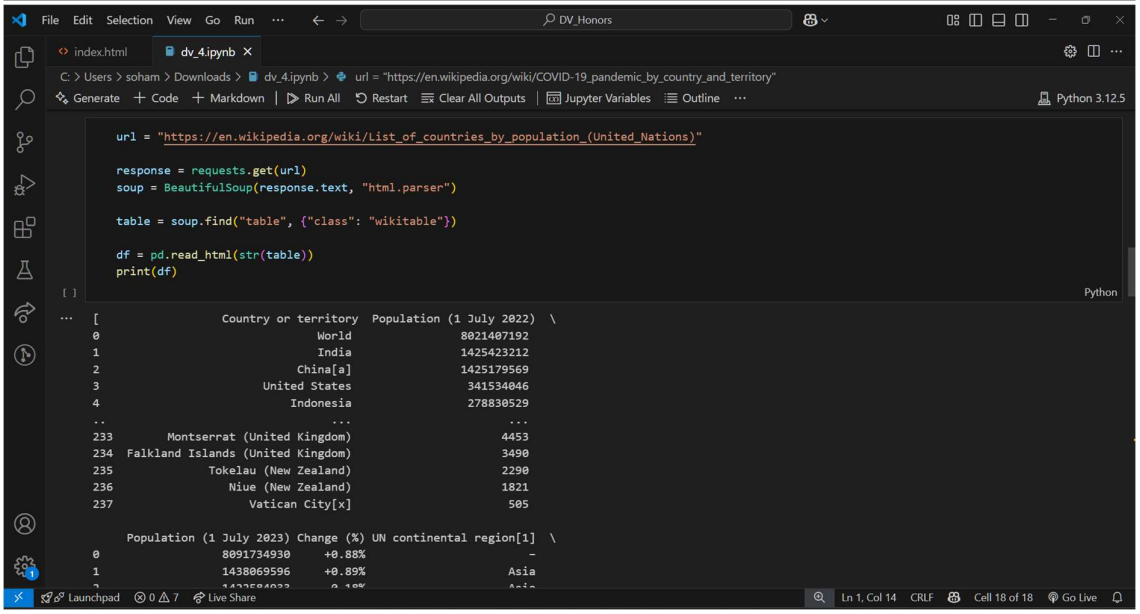
[6]

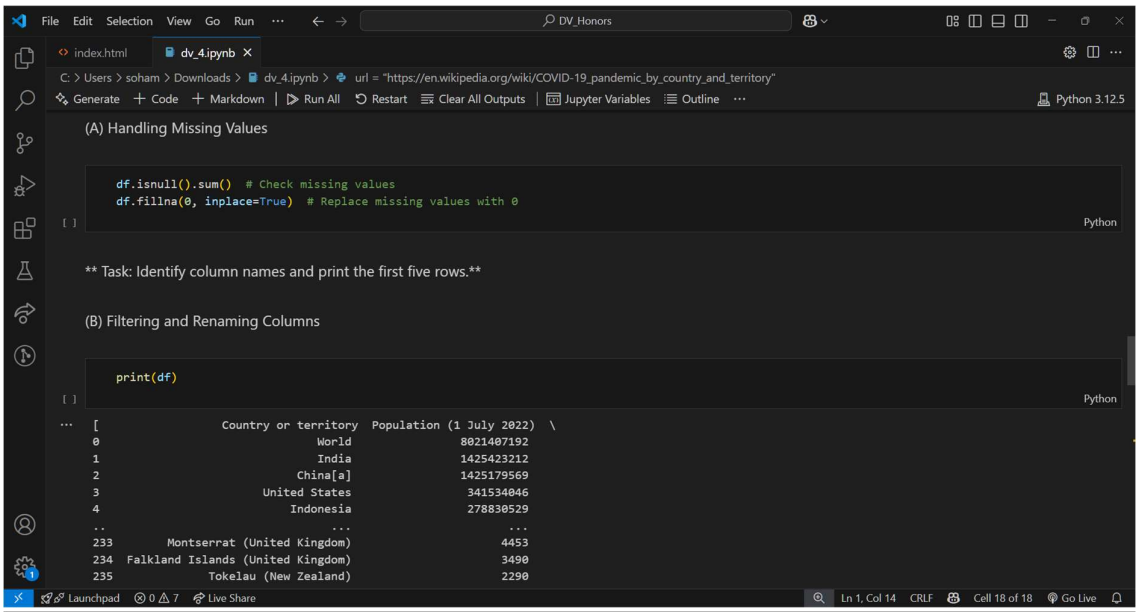
(B) Fetching Live Data from an API

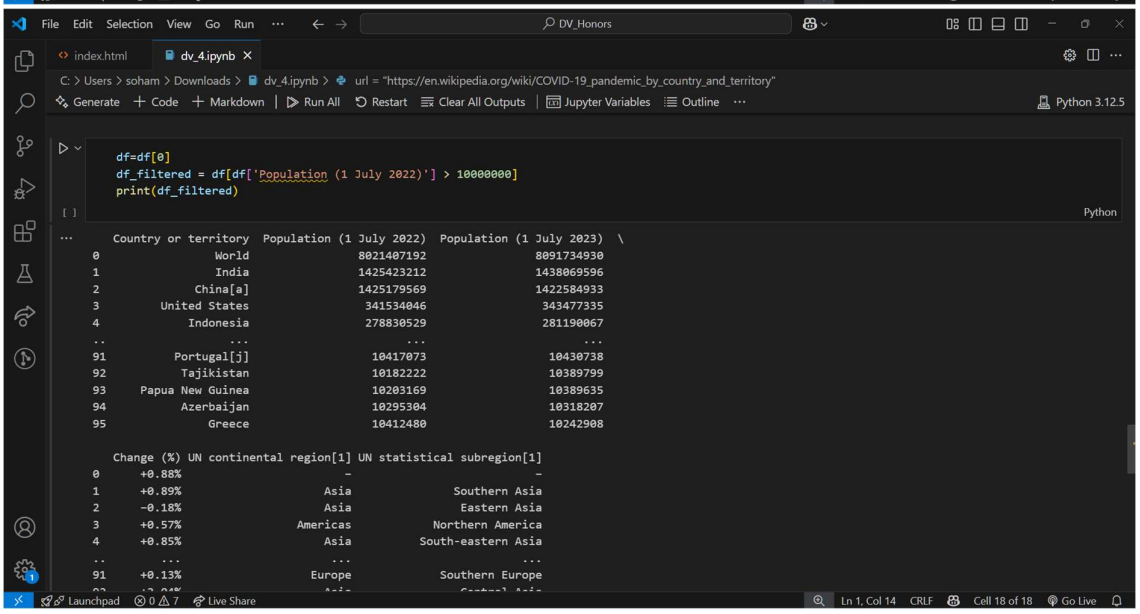
```python
api_key = "7a9ad05412f1f2af9ab39b67fa9388c0"
city = "Mumbai"
url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"

response = requests.get(url)
data = response.json()
print(*data.items(), sep='\n')
```

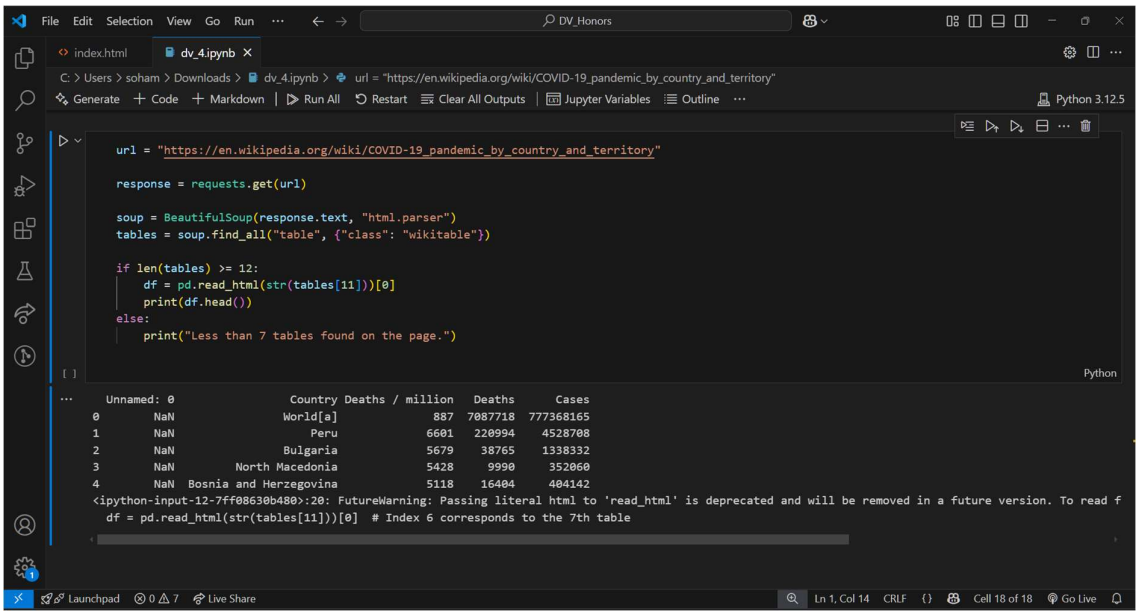
[6] ✓ 0.1s Python

```
('coord', {'lon': 72.8479, 'lat': 19.0144})
('weather', [{'id': 800, 'main': 'Clear', 'description': 'clear sky', 'icon': '01d'}])
('base', 'stations')
('main', {'temp': 301.85, 'feels_like': 305.07, 'temp_min': 301.85, 'temp_max': 301.85, 'pressure': 1008, 'humidity': 69, 'sea_level': 1008, 'grnd_leve
('visibility', 10000)
('wind', {'speed': 4.89, 'deg': 284, 'gust': 5.35})
('clouds', {'all': 5})
('dt', 1744459814)
('sys', {'country': 'IN', 'sunrise': 1744419215, 'sunset': 1744464289})
('timezone', 19800)
('id', 1275339)
('name', 'Mumbai')
('cod', 200)
```




File Edit Selection View Go Run
DV_Honors
index.html
dv_4.ipynb
C: > Users > soham > Downloads > dv_4.ipynb > url = "https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory"
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline
Python 3.12.5
(A) Handling Missing Values
df.isnull().sum() # Check missing values
df.fillna(0, inplace=True) # Replace missing values with 0
Python
** Task: Identify column names and print the first five rows.**
(B) Filtering and Renaming Columns
print(df)
Python
[Country or territory Population (1 July 2022) \
0 World 8021407192
1 India 1425423212
2 China[a] 1425179569
3 United States 341534046
4 Indonesia 278830529
.. ..
233 Montserrat (United Kingdom) 4453
234 Falkland Islands (United Kingdom) 3490
235 Tokelau (New Zealand) 2290
Launchpad 0 7 Live Share
Ln 1, Col 14 CRLF Cell 18 of 18 Go Live


File Edit Selection View Go Run
DV_Honors
index.html
dv_4.ipynb
C: > Users > soham > Downloads > dv_4.ipynb > url = "https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory"
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline
Python 3.12.5
df=df[0]
df_filtered = df[df['Population (1 July 2022)'] > 10000000]
print(df_filtered)
Python
Country or territory Population (1 July 2022) Population (1 July 2023) \
0 World 8021407192 8091734930
1 India 1425423212 1438069596
2 China[a] 1425179569 1422584933
3 United States 341534046 343477335
4 Indonesia 278830529 281190067
.. ..
91 Portugal[j] 10417073 10430738
92 Tajikistan 10182222 10389799
93 Papua New Guinea 10203169 10389635
94 Azerbaijan 10295304 10318207
95 Greece 10412480 10242908
Change (%) UN continental region[1] UN statistical subregion[1]
0 +0.88% - -
1 +0.89% Asia Southern Asia
2 −0.18% Asia Eastern Asia
3 +0.57% Americas Northern America
4 +0.85% Asia South-eastern Asia
.. ..
91 +0.13% Europe Southern Europe
Launchpad 0 7 Live Share
Ln 1, Col 14 CRLF Cell 18 of 18 Go Live



CONCLUSION:

Data Collection offers the raw material—timely and accurate data—from various sources. Data Wrangling molds this raw material into a clean, consistent, and usable form for analysis or machine learning. These are the key steps in ensuring the quality and integrity of any data-driven project. Clean, properly prepared data frequently determines the success of analysis or model performance.