

# Phase 3

## COVID Vaccine Analysis

To load and preprocess a COVID Vaccine dataset for analysis, you can follow these general steps using Python and Pandas. Make sure you have a COVID Vaccine dataset in a suitable format available.

**1. Import Libraries:** Start by importing the necessary Python libraries, including Pandas, to load and preprocess the dataset.

Import pandas as pd

**2. Load the COVID-19 Dataset:** Load the COVID Vaccine dataset into a Pandas DataFrame. You can use `pd.read_csv()` for CSV files, but the method may vary depending on the file format.

In [1]: `import pandas as pd`

In [2]: `cv = pd.read_csv(r"C:\Users\MUHILAN\OneDrive\Documents\country_vaccinations.csv")`  
`cv`

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	...
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows x 15 columns

**3. Data Inspection:** Before preprocessing, inspect the data to understand its structure and identify any potential issues.

In [3]: `print(cv.head())`

```

country iso_code date total_vaccinations people_vaccinated \
0 Afghanistan AFG 2021-02-22 0.0 0.0
1 Afghanistan AFG 2021-02-23 NaN NaN
2 Afghanistan AFG 2021-02-24 NaN NaN
3 Afghanistan AFG 2021-02-25 NaN NaN
4 Afghanistan AFG 2021-02-26 NaN NaN

people_fully_vaccinated daily_vaccinations_raw daily_vaccinations \
0 NaN NaN NaN
1 NaN NaN 1367.0
2 NaN NaN 1367.0
3 NaN NaN 1367.0
4 NaN NaN 1367.0

total_vaccinations_per_hundred people_vaccinated_per_hundred \
0 0.0 0.0
1 NaN NaN
2 NaN NaN
3 NaN NaN
4 NaN NaN

people_fully_vaccinated_per_hundred daily_vaccinations_per_million \
0 NaN NaN
1 NaN 34.0
2 NaN 34.0
3 NaN 34.0
4 NaN 34.0

vaccines \
0 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
1 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
2 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
3 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
4 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

source_name source_website
0 World Health Organization https://covid19.who.int/
1 World Health Organization https://covid19.who.int/
2 World Health Organization https://covid19.who.int/
3 World Health Organization https://covid19.who.int/
4 World Health Organization https://covid19.who.int/

```

In [5]: `print(cv.isnull().sum())`

country	0
iso_code	0
date	0
total_vaccinations	42905
people_vaccinated	45218
people_fully_vaccinated	47710
daily_vaccinations_raw	51150
daily_vaccinations	299
total_vaccinations_per_hundred	42905
people_vaccinated_per_hundred	45218
people_fully_vaccinated_per_hundred	47710
daily_vaccinations_per_million	299
vaccines	0
source_name	0
source_website	0
dtype: int64	

In [6]: `print(cv.dtypes)`

country	object
iso_code	object
date	object
total_vaccinations	float64
people_vaccinated	float64
people_fully_vaccinated	float64
daily_vaccinations_raw	float64
daily_vaccinations	float64
total_vaccinations_per_hundred	float64
people_vaccinated_per_hundred	float64
people_fully_vaccinated_per_hundred	float64
daily_vaccinations_per_million	float64
vaccines	object
source_name	object
source_website	object
dtype: object	

## 4. Data Preprocessing:

### a. Data Cleaning:

- Handle missing values by either imputing them or removing rows with missing data.

```
In [7]: cv.fillna(0, inplace=True)
cv.dropna(inplace=True)
cv
```

Out[7]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	0.0	0.0	0.0	
1	Afghanistan	AFG	2021-02-23	0.0	0.0	0.0	0.0	1367.0	
2	Afghanistan	AFG	2021-02-24	0.0	0.0	0.0	0.0	1367.0	
3	Afghanistan	AFG	2021-02-25	0.0	0.0	0.0	0.0	1367.0	
4	Afghanistan	AFG	2021-02-26	0.0	0.0	0.0	0.0	1367.0	
...	...	...	...	...	...	...	...	...	...
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows x 15 columns

## b. Data Transformation:

- If necessary, transform the data to suit your analysis objectives. For instance, you may want to aggregate data by date or region.

```
In [11]: cv['date'] = pd.to_datetime(cv['date'])
cv = cv.groupby('date').agg({'country': 'sum', 'total_vaccinations': 'sum'}).reset_index()
cv
```

Out[11]:

	date	country	total_vaccinations
0	2020-12-02	Norway	0.000000e+00
1	2020-12-03	Norway	0.000000e+00
2	2020-12-04	LatviaNorway	1.000000e+00
3	2020-12-05	LatviaNorway	0.000000e+00
4	2020-12-06	LatviaNorway	0.000000e+00
...	...	...	...
478	2022-03-25	AngolaAnguillaArgentinaArubaAustraliaAustriaAz...	8.803269e+09
479	2022-03-26	ArgentinaArubaAustraliaAzerbaijanBangladeshBar...	8.785869e+09
480	2022-03-27	ArgentinaArubaAustraliaAzerbaijanBangladeshBar...	9.136042e+09
481	2022-03-28	ArgentinaArubaAustraliaAzerbaijanBangladeshBar...	8.982040e+09
482	2022-03-29	ArgentinaArubaAustraliaAzerbaijanBangladeshBra...	8.343523e+09

483 rows × 3 columns

### c. Data Filtering:

- Filter the data to focus on a specific time frame or specific regions of interest.

```
In [13]: start_date = '2021-06-01'
end_date = '2021-12-31'
cv = cv[(cv['date'] >= start_date) & (cv['date'] <= end_date)]
cv
```

Out[13]:

	date	country	total_vaccinations
181	2021-06-01	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	2.016435e+09
182	2021-06-02	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	2.045128e+09
183	2021-06-03	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	2.081324e+09
184	2021-06-04	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	2.041895e+09
185	2021-06-05	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	2.142456e+09
...	...	...	...
390	2021-12-27	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	8.012818e+09
391	2021-12-28	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	8.308256e+09
392	2021-12-29	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	7.585792e+09
393	2021-12-30	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	8.070097e+09
394	2021-12-31	AfghanistanAlbaniaAlgeriaAndorraAngolaAnguilla...	7.627917e+09

214 rows × 3 columns

**5. Save the Preprocessed Data (Optional):** If you want to save the preprocessed data for future analysis, you can use Pandas to save it to a new CSV file.

```
In [14]: cv.to_csv('preprocessed_cv_data.csv', index=False)
```

These are the general steps to load and preprocess a COVID Vaccine dataset using Python and Pandas. Remember that the specific preprocessing steps and operations may vary depending on the structure of your dataset and your analysis objectives.