COVID 19 VACCINE ANALYSIS

Phase 2

**2.1 Short explanation:**

Covid vaccine analysis informs vaccine distribution strategies,addressing logistical challenges and promoting equitable access. It plays a cruical role in managing vaccine hesitancy by providing data on a vaccine safety and efficacy. It guides decisions on potential booster doses and adaptation to combat emerging variants of the virus.

**2.2 Data set link:**

[**https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress**](https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress)

The data (country vaccinations) contains the following information:

* **Country**- this is the country for which the vaccination information is provided;
* **Country ISO Code** - ISO code for the country;
* **Date** - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;
* **Total number of vaccinations** - this is the absolute number of total immunizations in the country;
* **Total number of people vaccinated** - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccinations might be larger than the number of people;
* **Total number of people fully vaccinated** - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;
* **Daily vaccinations (raw)** - for a certain data entry, the number of vaccinations for that date/country;
* **Daily vaccinations** - for a certain data entry, the number of vaccinations for that date/country;
* **Total vaccinations per hundred** - ratio (in percent) between vaccination number and total population up to the date in the country;
* **Total number of people vaccinated per hundred** - ratio (in percent) between population immunized and total population up to the date in the country;
* **Total number of people fully vaccinated per hundred** - ratio (in percent) between population fully immunized and total population up to the date in the country;
* **Number of vaccinations per day** - number of daily vaccinations for that day and country;
* **Daily vaccinations per million** - ratio (in ppm) between vaccination number and total population for the current date in the country;
* **Vaccines used in the country** - total number of vaccines used in the country (up to date);
* **Source name** - source of the information (national authority, international organization, local organization etc.);

**2.3 Columns to be used :**

* Total vaccination
* People vaccinated
* People fully vaccinated
* Country

**2.4 Libraries :**

NUMPY:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

* To download : Pip install numpy
* To import: import numpy as np

PANDAS:

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

* To download : Pip install pandas
* To import: import pandas as pd

MATPLOTLIB:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

* To download : Pip install matplotlib
* To import: import matplotlib.pyplot as plt

SEABORN:

Seaborn library is a widely popular data visualization library that is commonly used for data science and machine learning tasks

* To download : Pip install seaborn
* To import: import seaborn as sns

**2.5 TEST AND TRAIN:**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

df=pd.read\_csv("country\_vaccinations.csv")

print(df.info()) #TO FIND BASIC INFORMATIION ABOUT DATASET

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 86512 entries, 0 to 86511

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 country 86512 non-null object

1 iso\_code 86512 non-null object

2 date 86512 non-null object

3 total\_vaccinations 43607 non-null float64

4 people\_vaccinated 41294 non-null float64

5 people\_fully\_vaccinated 38802 non-null float64

6 daily\_vaccinations\_raw 35362 non-null float64

7 daily\_vaccinations 86213 non-null float64

8 total\_vaccinations\_per\_hundred 43607 non-null float64

9 people\_vaccinated\_per\_hundred 41294 non-null float64

10 people\_fully\_vaccinated\_per\_hundred 38802 non-null float64

11 daily\_vaccinations\_per\_million 86213 non-null float64

12 vaccines 86512 non-null object

13 source\_name 86512 non-null object

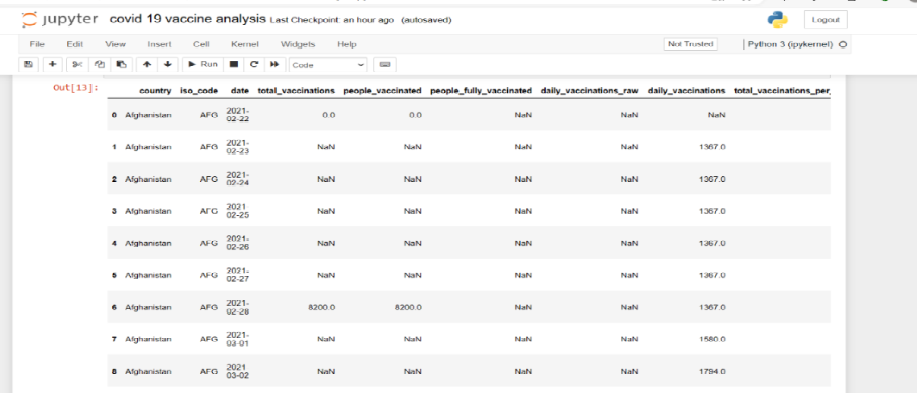
14 source\_website 86512 non-null object

dtypes: float64(9), object(6)

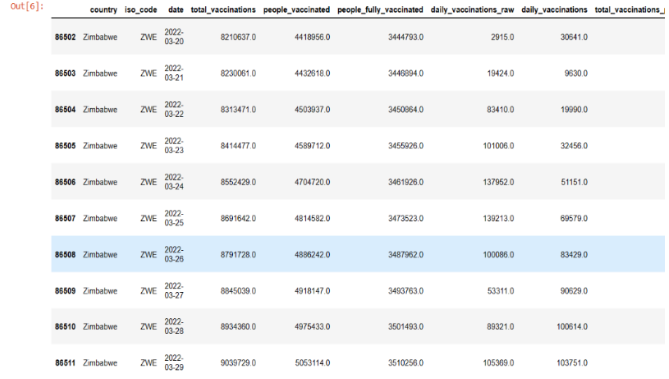
memory usage: 9.9+ MB

None

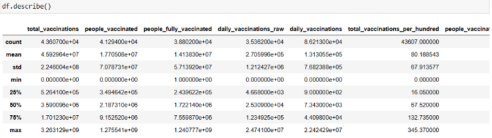
df.head(10) #DISPLAY FIRST 10 DATA IN DATASET



df.tail(10)



df.describe()



df.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

vaccines

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech 7608

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech 6263

df.value\_counts("vaccines")

vaccines

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech 7608

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech 6263

Oxford/AstraZeneca 6022

Oxford/AstraZeneca, Pfizer/BioNTech 4629

Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech 3564

...

Johnson&Johnson, Oxford/AstraZeneca, Sinovac 312

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V 311

Johnson&Johnson, Moderna 251

Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing 228

EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001 190

Length: 84, dtype: int64

df.hist(figsize=(12,12),layout=(5,3))

array([[<AxesSubplot:title={'center':'total\_vaccinations'}>,

<AxesSubplot:title={'center':'people\_vaccinated'}>,

<AxesSubplot:title={'center':'people\_fully\_vaccinated'}>],

[<AxesSubplot:title={'center':'daily\_vaccinations\_raw'}>,

<AxesSubplot:title={'center':'daily\_vaccinations'}>,

<AxesSubplot:title={'center':'total\_vaccinations\_per\_hundred'}>],

[<AxesSubplot:title={'center':'people\_vaccinated\_per\_hundred'}>,

<AxesSubplot:title={'center':'people\_fully\_vaccinated\_per\_hundred'}>,

<AxesSubplot:title={'center':'daily\_vaccinations\_per\_million'}>],

[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>],

[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]], dtype=object)

**2.6 EXPLANATION:**

### **Total Vaccinated till Date**

In this section, we are going to see how many total vaccines have been used in each country. Check the code below for more information. The data shows the United States has administrated most vaccines in the world followed by China, United Kingdom, England, India and at the last some countries include Saint Helena, San Marino has 0 vaccination.

country\_wise\_total\_vaccinated = {}  
for country in df.country.unique() :   
 vaccinated = 0  
 for i in range(len(df)) :   
 if df.country[i] == country :   
 vaccinated += df.daily\_vaccinations[i]  
 country\_wise\_total\_vaccinated[country] = vaccinated   
# made a seperate dict from the df   
 country\_wise\_total\_vaccinated\_df = pd.DataFrame.from\_dict(country\_wise\_total\_vaccinated,  
 orient='index',  
 columns = ['total\_vaccinted\_till\_date'])  
# converted dict to df   
country\_wise\_total\_vaccinated\_df.sort\_values(by = 'total\_vaccinted\_till\_date', ascending = False, inplace = True)  
country\_wise\_total\_vaccinated\_df

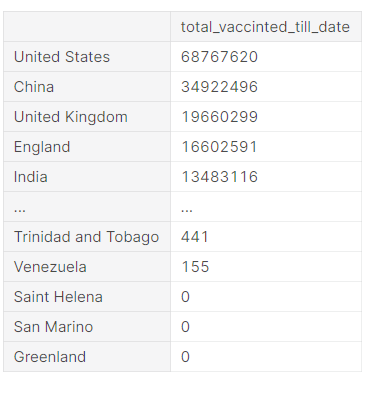
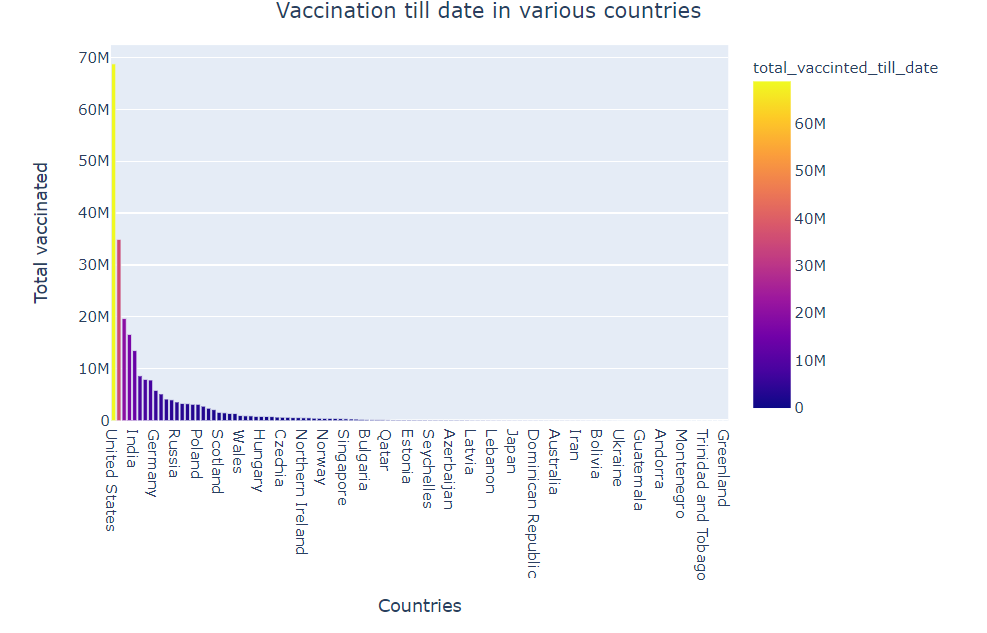


fig = px.bar(country\_wise\_total\_vaccinated\_df,   
 y = 'total\_vaccinted\_till\_date',  
 x = country\_wise\_total\_vaccinated\_df.index,  
 color = 'total\_vaccinted\_till\_date',  
 color\_discrete\_sequence= px.colors.sequential.Viridis\_r  
 )  
fig.update\_layout(  
 title={  
 'text' : "Vaccination till date in various countries",  
 'y':0.95,  
 'x':0.5  
 },  
 xaxis\_title="Countries",  
 yaxis\_title="Total vaccinated",  
 legend\_title="Total vaccinated"  
)  
fig.show()



### **Country Wise Daily Vaccination**

To check what is the vaccination trend in each country, check the below code. We are drawing the line plot where the x-axis is the date and the y-axis is the count of daily vaccination, Colours Is set to be the **country**.

fig = px.line(df, x = 'date', y ='daily\_vaccinations', color = 'country')  
fig.update\_layout(  
 title={  
 'text' : "Daily vaccination trend",  
 'y':0.95,  
 'x':0.5  
 },  
 xaxis\_title="Date",  
 yaxis\_title="Daily Vaccinations"  
)  
fig.show()

**2.7 METRICS USED FOR ACCURACY:**

Precision is used for accuracy checks. Precision is a measure of a model's performance that tells you how many of the positive predictions made by the model are actually correct. It is calculated as the number of true positive predictions divided by the number of true positive and false positive predictions.