# Naan Mudhalvan IBM project

## Applied DataScience(Phase 5)

## Topic- covid 19 Vaccine Analysis

**By: Sushthi. R(au411521104115)**

## Problem Definition:

This Project mainly aims to find out the trend of the vaccinations around the world for the prevention of the Covid 19 pandemic and how much has been achieved so far. It also aims to convey the analysis of different ongoing vaccination programs around the globe by using the inferences discovered from the scraped data from the internet. The python libraries used in the exploratory data analysis include *NumPy, Pandas, Matplotlib, Seaborn, and Plotly*.

## Design thinking:

## Data Preparation & Cleaning:

We read the data file and aggregate the data on a few fields (country, iso_code, and vaccines — that is the vaccination scheme used in a certain country). Data Cleaning is the most crucial step towards a successful data analysis project. In most of the cases, the dataset has few “NaN”(not a number) values, some empty rows(having value 0) as well as redundant columns which could be removed using and configuring drop function and changing NaN values to 0 or removing the entire row as per need.

## Exploratory Data Analysis and Visualization:

We will initialize the Python packages, that we are going to use for data ingestion and visualization. We will configure the environment by setting the font size, figure size, face color, etc. Also, we would mostly use seaborn for our visualization.

## Statistical analysis:

Statistical hypothesis testing, apply estimation statistics and interpret the results. We will also validate this with the findings from part one. We will apply both parametric and non-parametric tests.

## Insights:

Here we analyzed the top 10 fully vaccinated countries in which India tops the list which indicates that people in the country where showing lots of interests to get vaccinated.And also analyzed top 5 vaccinated countries here also India tops the list.And then analyzed top 5 daily vaccinating countries and here China tops the list.And also we analyse the sum of daily vaccinating details, fully vaccinating and vaccinating people details.And our year wise analyse shows that 2021 was the peak year for every vaccination details.

## Recommendations:

We should collect day to day reports and we should update our records daily to get more accurate details.So that we can move forward with more vaccination to the right country which needs the most.

## Phase of Development:

**Dataset link: https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress**

## Dataset description:

Data is collected daily from kaggle GitHub repository for covid-19, merged and uploaded. Country level vaccination data is gathered and assembled in one single file. Then, this data file is merged with locations data file to include vaccination sources information.

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 vaccination 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 vaccination for that date/country;
- **Daily vaccinations -** for a certain data entry, the number of vaccination 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 vaccination 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.);
- **Source website -** website of the source of information;

## Importing the libraries

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

import seaborn as sns

import plotly.express as px

import plotly.graph_objects as go

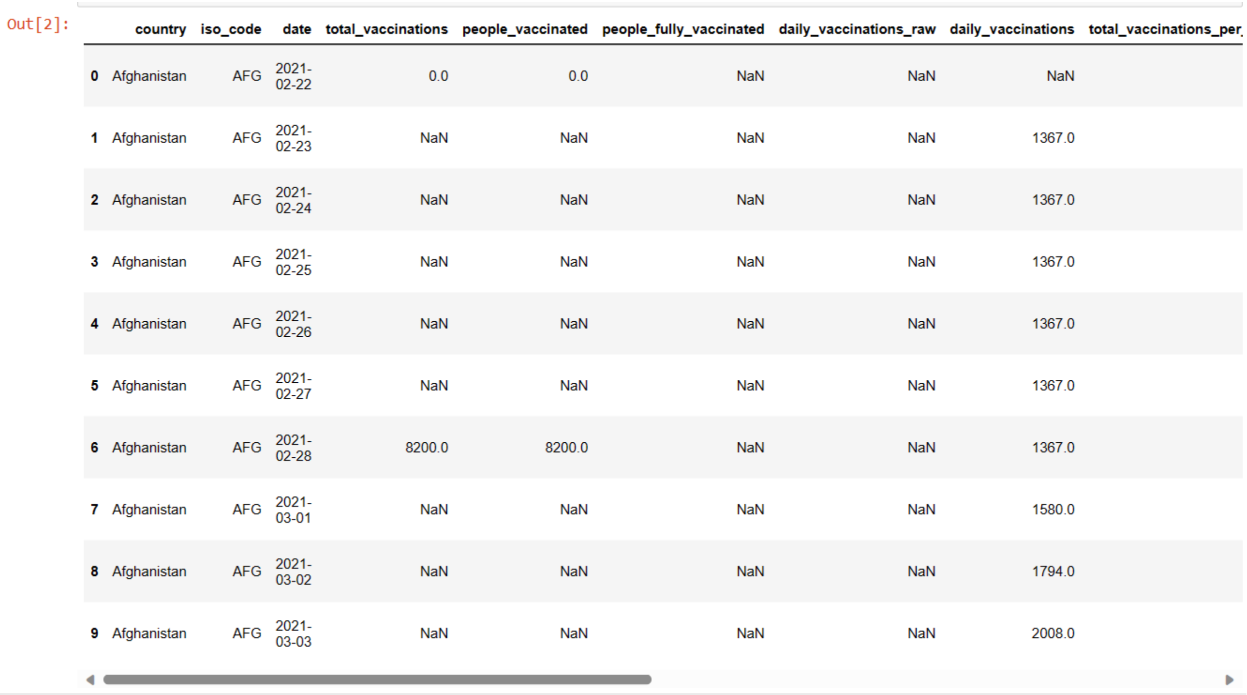
import warnings

warnings.filterwarnings('ignore')

## Importing the data

dataset = pd.read_csv("country_vaccinations.csv")

dataset.head(10) # we check the first 10 rows of our dataset



Out[2]:

|   | country | iso_code | date | total_vaccinations | people_vaccinated | people_fully_vaccinated | daily_vaccinations_raw | daily_vaccinations | total_vaccinations_per_ |
|---|---|---|---|---|---|---|---|---|---|
| 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 | |
| 5 | Afghanistan | AFG | 2021-02-27 | NaN | NaN | NaN | NaN | 1367.0 | |
| 6 | Afghanistan | AFG | 2021-02-28 | 8200.0 | 8200.0 | NaN | NaN | 1367.0 | |
| 7 | Afghanistan | AFG | 2021-03-01 | NaN | NaN | NaN | NaN | 1580.0 | |
| 8 | Afghanistan | AFG | 2021-03-02 | NaN | NaN | NaN | NaN | 1794.0 | |
| 9 | Afghanistan | AFG | 2021-03-03 | NaN | NaN | NaN | NaN | 2008.0 | |

## Finding null values present

df.isna().sum().any()

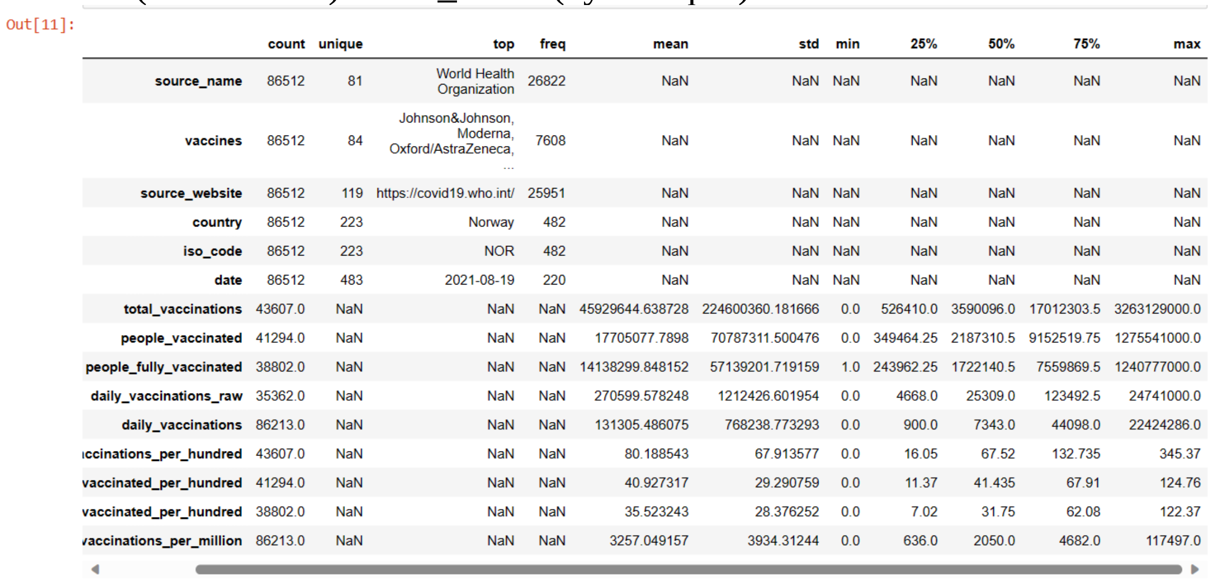
```
True
```

df.isna().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
```

df.describe(include='all').T.sort_values(by='unique')

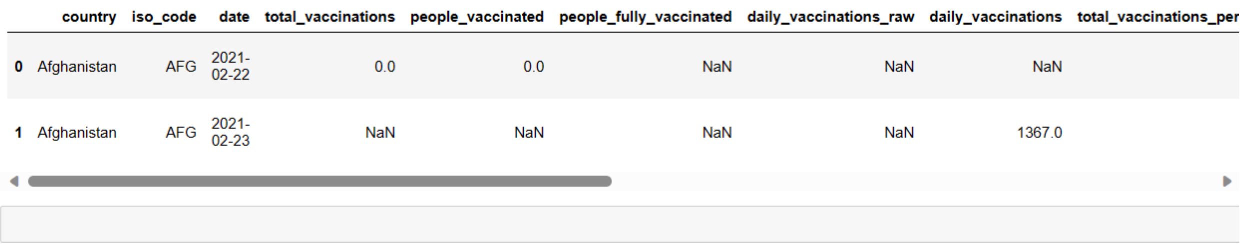


Out[11]:

| | count | unique | top | freq | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **source_name** | 86512 | 81 | World Health Organization | 26822 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **vaccines** | 86512 | 84 | Johnson&Johnson, Moderna, Oxford/AstraZeneca, ... | 7608 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **source_website** | 86512 | 119 | https://covid19.who.int/ | 25951 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **country** | 86512 | 223 | Norway | 482 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **iso_code** | 86512 | 223 | NOR | 482 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **date** | 86512 | 483 | 2021-08-19 | 220 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **total_vaccinations** | 43607.0 | NaN | NaN | NaN | 45929644.638728 | 224600360.181666 | 0.0 | 526410.0 | 3590096.0 | 17012303.5 | 3263129000.0 |
| **people_vaccinated** | 41294.0 | NaN | NaN | NaN | 17705077.7898 | 70787311.500476 | 0.0 | 349464.25 | 2187310.5 | 9152519.75 | 1275541000.0 |
| **people_fully_vaccinated** | 38802.0 | NaN | NaN | NaN | 14138299.848152 | 57139201.719159 | 1.0 | 243962.25 | 1722140.5 | 7559869.5 | 1240777000.0 |
| **daily_vaccinations_raw** | 35362.0 | NaN | NaN | NaN | 270599.578248 | 1212426.601954 | 0.0 | 4668.0 | 25309.0 | 123492.5 | 24741000.0 |
| **daily_vaccinations** | 86213.0 | NaN | NaN | NaN | 131305.486075 | 768238.773293 | 0.0 | 900.0 | 7343.0 | 44098.0 | 22424286.0 |
| **ıccinations_per_hundred** | 43607.0 | NaN | NaN | NaN | 80.188543 | 67.913577 | 0.0 | 16.05 | 67.52 | 132.735 | 345.37 |
| **vaccinated_per_hundred** | 41294.0 | NaN | NaN | NaN | 40.927317 | 29.290759 | 0.0 | 11.37 | 41.435 | 67.91 | 124.76 |
| **vaccinated_per_hundred** | 38802.0 | NaN | NaN | NaN | 35.523243 | 28.376252 | 0.0 | 7.02 | 31.75 | 62.08 | 122.37 |
| **vaccinations_per_million** | 86213.0 | NaN | NaN | NaN | 3257.049157 | 3934.31244 | 0.0 | 636.0 | 2050.0 | 4682.0 | 117497.0 |

df1 = df.copy() // copy of original file

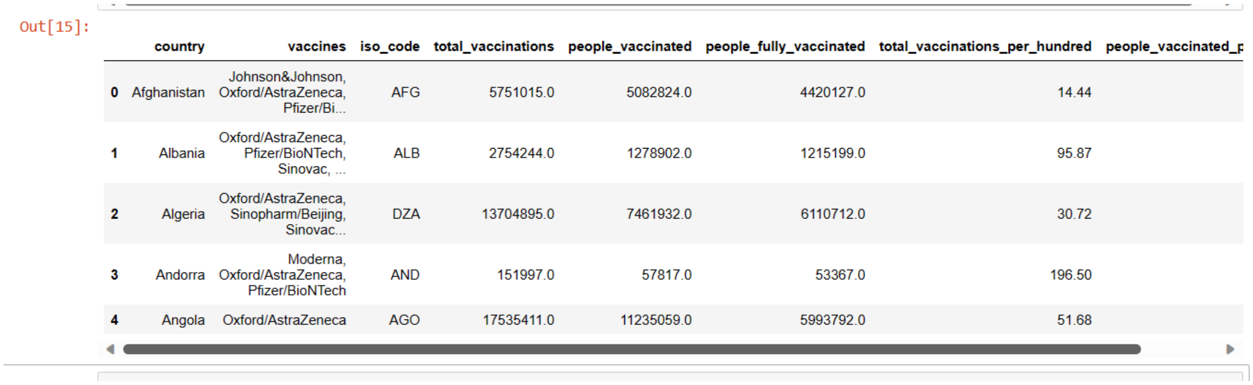
df1.head(2) //first two data in df1



| | country | iso_code | date | total_vaccinations | people_vaccinated | people_fully_vaccinated | daily_vaccinations_raw | daily_vaccinations | total_vaccinations_per |
|---|---|---|---|---|---|---|---|---|---|
| **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 | |

```
vaccine = df1.groupby(['country','vaccines','iso_code'])['total_vaccinations','people_vaccinated','people_fully_vaccinated','total_vaccinations_per_hundred','people_vaccinated_per_hundred'].max().reset_index()
```

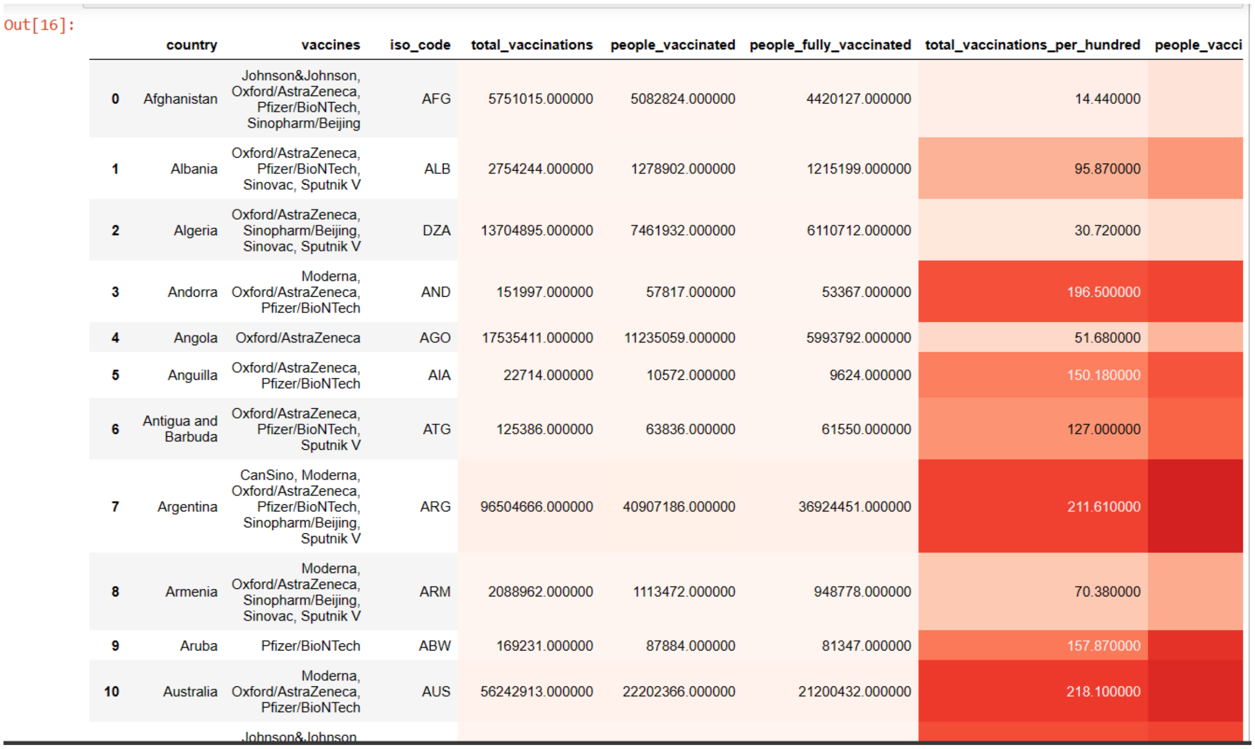
```
vaccine.head()
```



Out[15]:

| | country | vaccines | iso_code | total_vaccinations | people_vaccinated | people_fully_vaccinated | total_vaccinations_per_hundred | people_vaccinated_p |
|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... | AFG | 5751015.0 | 5082824.0 | 4420127.0 | 14.44 | |
| **1** | Albania | Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ... | ALB | 2754244.0 | 1278902.0 | 1215199.0 | 95.87 | |
| **2** | Algeria | Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac... | DZA | 13704895.0 | 7461932.0 | 6110712.0 | 30.72 | |
| **3** | Andorra | Moderna, Oxford/AstraZeneca, Pfizer/BioNTech | AND | 151997.0 | 57817.0 | 53367.0 | 196.50 | |
| **4** | Angola | Oxford/AstraZeneca | AGO | 17535411.0 | 11235059.0 | 5993792.0 | 51.68 | |

```
# Here Red color indicates the maximum number of data entries

vaccine.style.background_gradient(cmap='Reds')
```

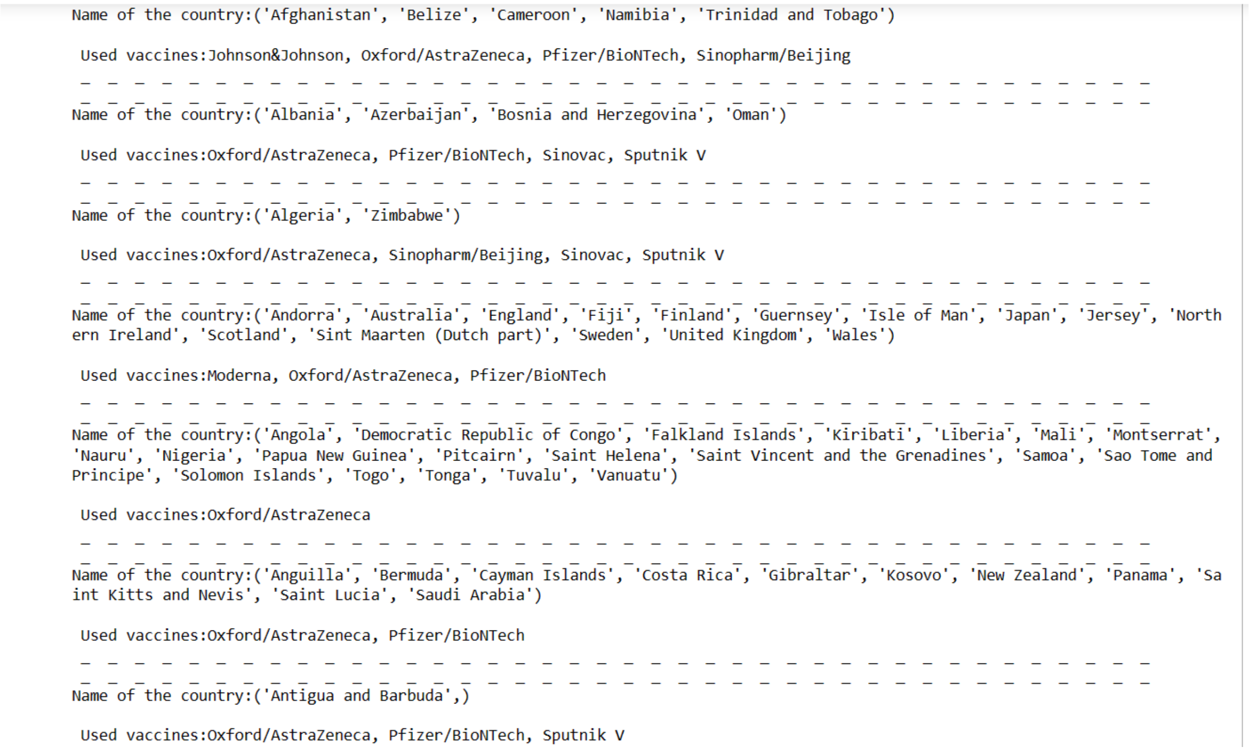


Out[16]:

| | country | vaccines | iso_code | total_vaccinations | people_vaccinated | people_fully_vaccinated | total_vaccinations_per_hundred | people_vacci |
|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing | AFG | 5751015.000000 | 5082824.000000 | 4420127.000000 | 14.440000 | |
| **1** | Albania | Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V | ALB | 2754244.000000 | 1278902.000000 | 1215199.000000 | 95.870000 | |
| **2** | Algeria | Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V | DZA | 13704895.000000 | 7461932.000000 | 6110712.000000 | 30.720000 | |
| **3** | Andorra | Moderna, Oxford/AstraZeneca, Pfizer/BioNTech | AND | 151997.000000 | 57817.000000 | 53367.000000 | 196.500000 | |
| **4** | Angola | Oxford/AstraZeneca | AGO | 17535411.000000 | 11235059.000000 | 5993792.000000 | 51.680000 | |
| **5** | Anguilla | Oxford/AstraZeneca, Pfizer/BioNTech | AIA | 22714.000000 | 10572.000000 | 9624.000000 | 150.180000 | |
| **6** | Antigua and Barbuda | Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V | ATG | 125386.000000 | 63836.000000 | 61550.000000 | 127.000000 | |
| **7** | Argentina | CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V | ARG | 96504666.000000 | 40907186.000000 | 36924451.000000 | 211.610000 | |
| **8** | Armenia | Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V | ARM | 2088962.000000 | 1113472.000000 | 948778.000000 | 70.380000 | |
| **9** | Aruba | Pfizer/BioNTech | ABW | 169231.000000 | 87884.000000 | 81347.000000 | 157.870000 | |
| **10** | Australia | Moderna, Oxford/AstraZeneca, Pfizer/BioNTech | AUS | 56242913.000000 | 22202366.000000 | 21200432.000000 | 218.100000 | |
| | | Johnson&Johnson | | | | | | |

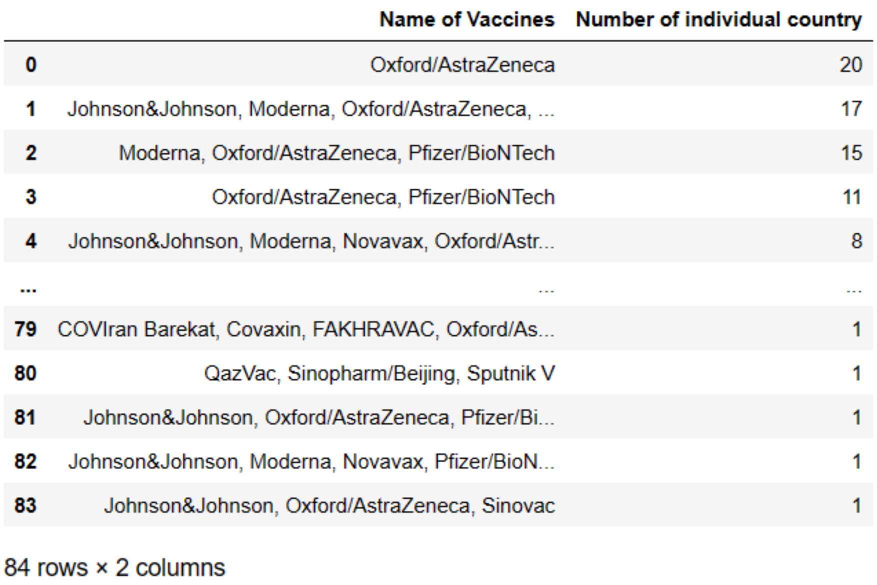
```
#which country used which vaccines to fight against COVID-19

vaccines_list = list(vaccine['vaccines'].unique())
```

```
for i in vaccines_list:
   country = tuple(vaccine[vaccine['vaccines']==i]['country'])
   print(f"Name of the country:{country}\n\n Used vaccines:{i}")
   print(' _ '*40)
   print(' _ '*40)
```



```
Name of the country:('Afghanistan', 'Belize', 'Cameroon', 'Namibia', 'Trinidad and Tobago')

 Used vaccines:Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Albania', 'Azerbaijan', 'Bosnia and Herzegovina', 'Oman')

 Used vaccines:Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Algeria', 'Zimbabwe')

 Used vaccines:Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Andorra', 'Australia', 'England', 'Fiji', 'Finland', 'Guernsey', 'Isle of Man', 'Japan', 'Jersey', 'North
ern Ireland', 'Scotland', 'Sint Maarten (Dutch part)', 'Sweden', 'United Kingdom', 'Wales')

 Used vaccines:Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Angola', 'Democratic Republic of Congo', 'Falkland Islands', 'Kiribati', 'Liberia', 'Mali', 'Montserrat',
'Nauru', 'Nigeria', 'Papua New Guinea', 'Pitcairn', 'Saint Helena', 'Saint Vincent and the Grenadines', 'Samoa', 'Sao Tome and
Principe', 'Solomon Islands', 'Togo', 'Tonga', 'Tuvalu', 'Vanuatu')

 Used vaccines:Oxford/AstraZeneca
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Anguilla', 'Bermuda', 'Cayman Islands', 'Costa Rica', 'Gibraltar', 'Kosovo', 'New Zealand', 'Panama', 'Sa
int Kitts and Nevis', 'Saint Lucia', 'Saudi Arabia')

 Used vaccines:Oxford/AstraZeneca, Pfizer/BioNTech
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _ 
Name of the country:('Antigua and Barbuda',)

 Used vaccines:Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
```

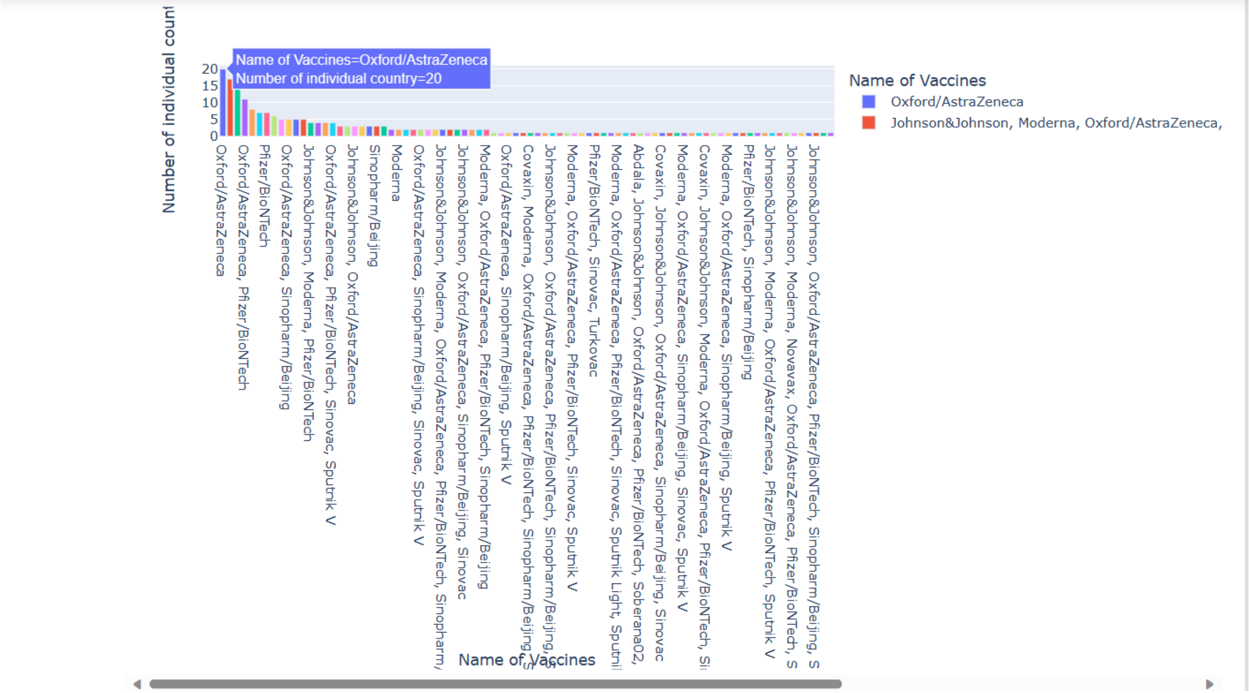
```
vaccines_used = vaccine['vaccines'].value_counts().reset_index()
vaccines_used.columns = ['Name of Vaccines','Number of individual country']
vaccines_used
```



| | Name of Vaccines | Number of individual country |
|---|---|---|
| **0** | Oxford/AstraZeneca | 20 |
| **1** | Johnson&Johnson, Moderna, Oxford/AstraZeneca, ... | 17 |
| **2** | Moderna, Oxford/AstraZeneca, Pfizer/BioNTech | 15 |
| **3** | Oxford/AstraZeneca, Pfizer/BioNTech | 11 |
| **4** | Johnson&Johnson, Moderna, Novavax, Oxford/Astr... | 8 |
| **...** | ... | ... |
| **79** | COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/As... | 1 |
| **80** | QazVac, Sinopharm/Beijing, Sputnik V | 1 |
| **81** | Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... | 1 |
| **82** | Johnson&Johnson, Moderna, Novavax, Pfizer/BioN... | 1 |
| **83** | Johnson&Johnson, Oxford/AstraZeneca, Sinovac | 1 |

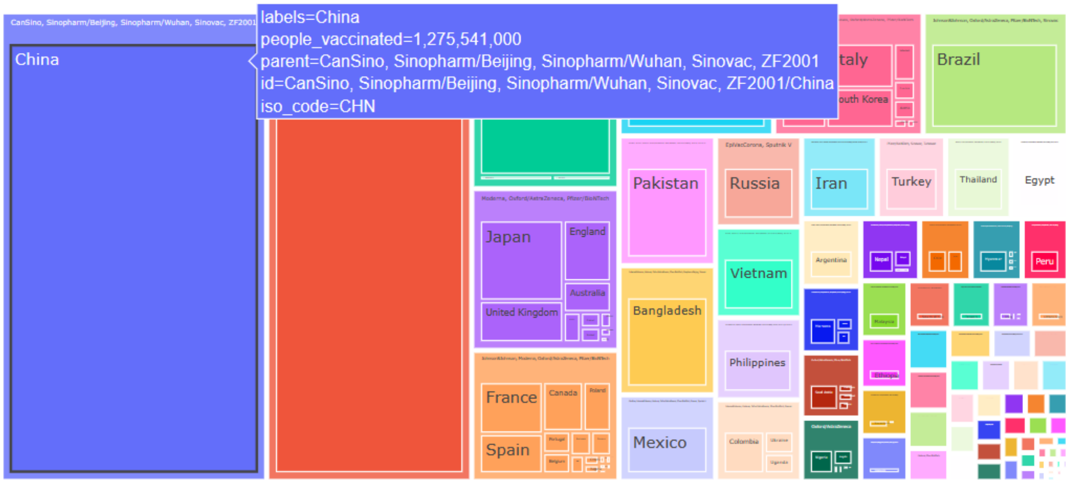
84 rows × 2 columns

fig = px.bar(vaccines_used,x='Name of Vaccines',y='Number of individual cry',color='Name of Vaccines',height=600,width=150)
fig.show()

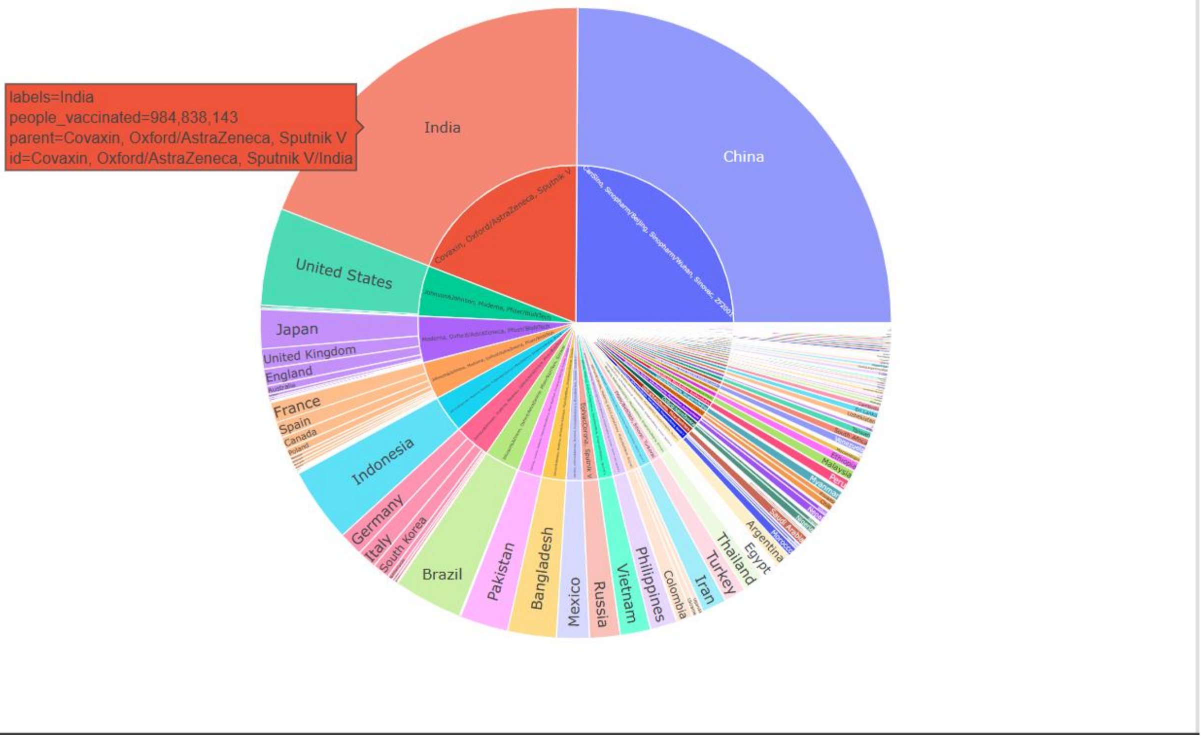


which country using which vaccines in figure and this can be visualized easily with tree map and sunburst

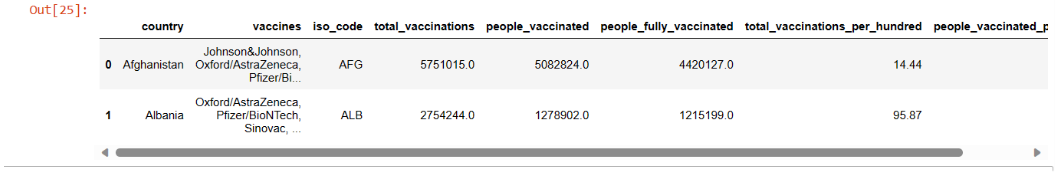
fig = px.treemap(vaccine,names='country',values='people_vaccinated',path=['vaccines','country'],hover_data=['iso_code])
fig.show()



```
fig = px.sunburst(vaccine,names='country',values='people_vaccinated',path=['vaccines','country'],
                  width=1000,height=700,title='Name of vaccines per Country'
                 )
fig.show()
```



vaccine.head(2)



Out[25]:

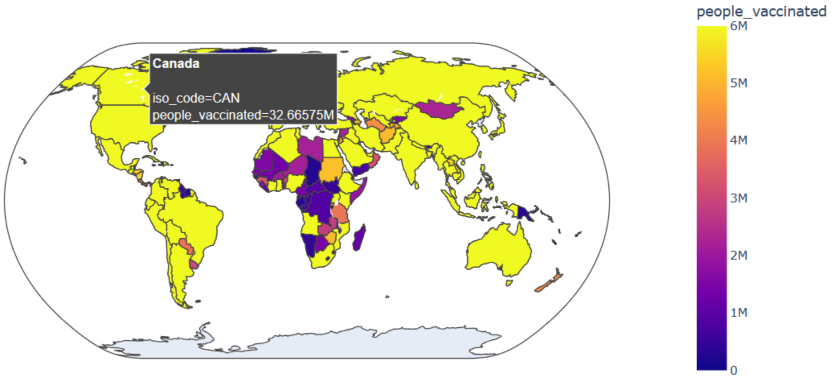
|   | country | vaccines | iso_code | total_vaccinations | people_vaccinated | people_fully_vaccinated | total_vaccinations_per_hundred | people_vaccinated_p |
|---|---|---|---|---|---|---|---|---|
| 0 | Afghanistan | Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... | AFG | 5751015.0 | 5082824.0 | 4420127.0 | 14.44 |  |
| 1 | Albania | Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ... | ALB | 2754244.0 | 1278902.0 | 1215199.0 | 95.87 |  |

fig =px.choropleth(vaccine,locations='iso_code',color='vaccines',projection='natural earth', hover_name='country',height=None )

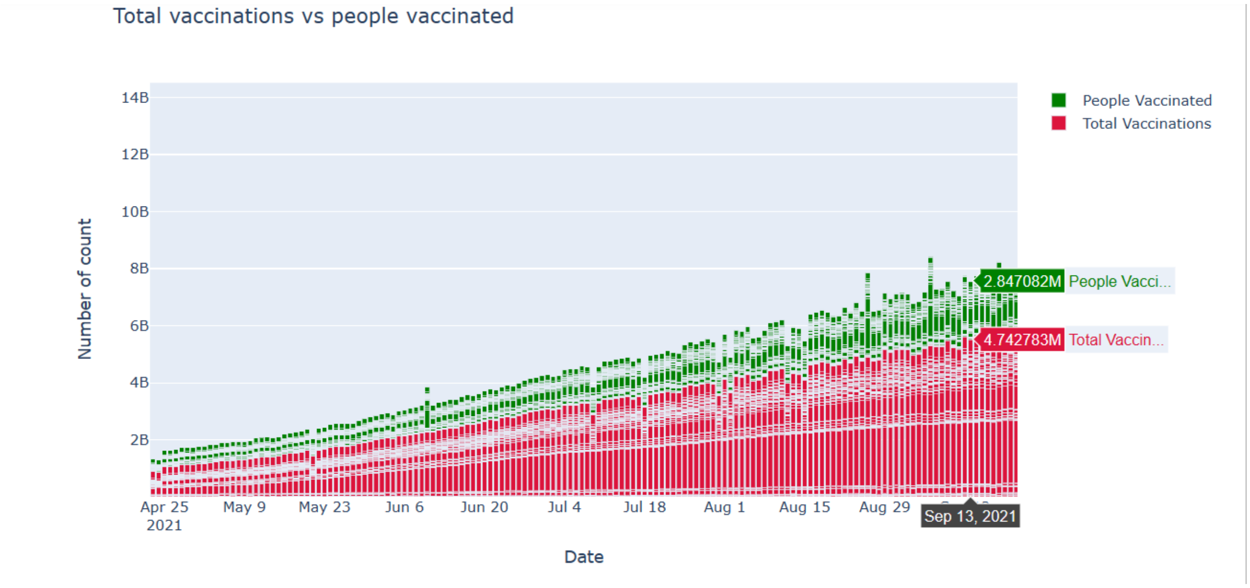
fig.show()

Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Oxford/AstraZeneca, Sputnik V
Moderna
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
Oxford/AstraZeneca, Sinopharm/Beijing
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Moderna
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
Pfizer/BioNTech, Sinovac
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Covaxin, Oxford/AstraZeneca, Sputnik V
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
QazVac, Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V
Johnson&Johnson, Moderna, Novavax, Pfizer/BioNTech
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
Pfizer/BioNTech, Sinopharm/Beijing
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac

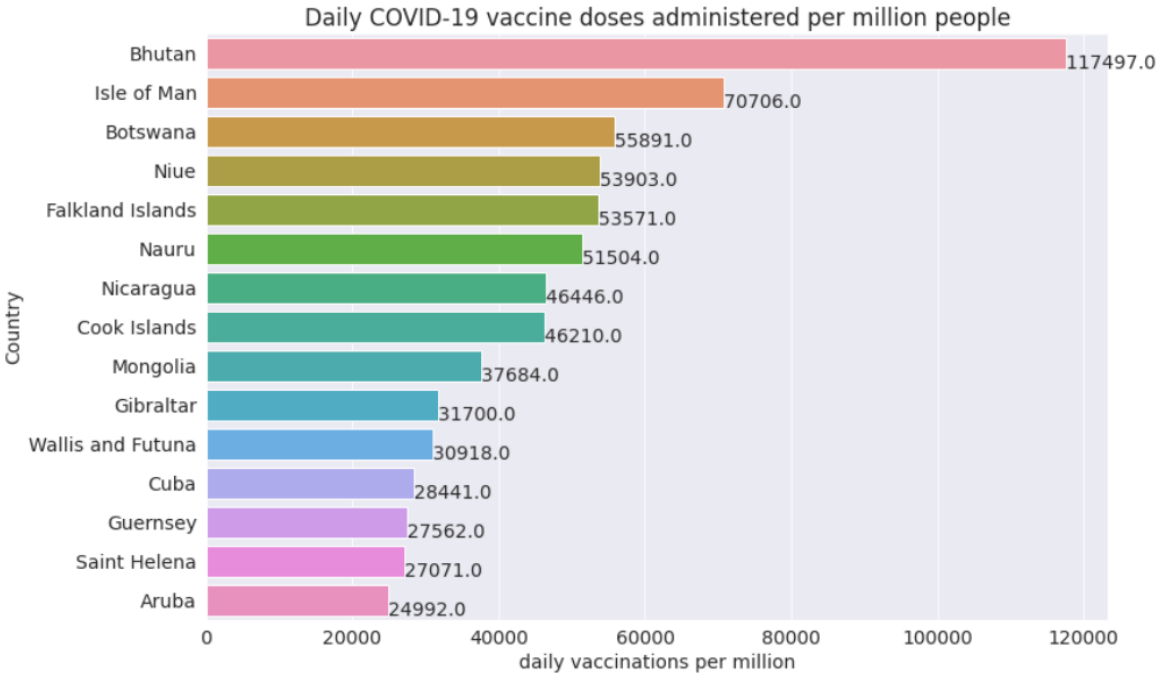
fig = px.choropleth(vaccine,locations='iso_code',color='people_vaccinated',projection='natural earth',
                    hover_name='country',height=None,range_color=[0,6000000],
                   )
fig.show()



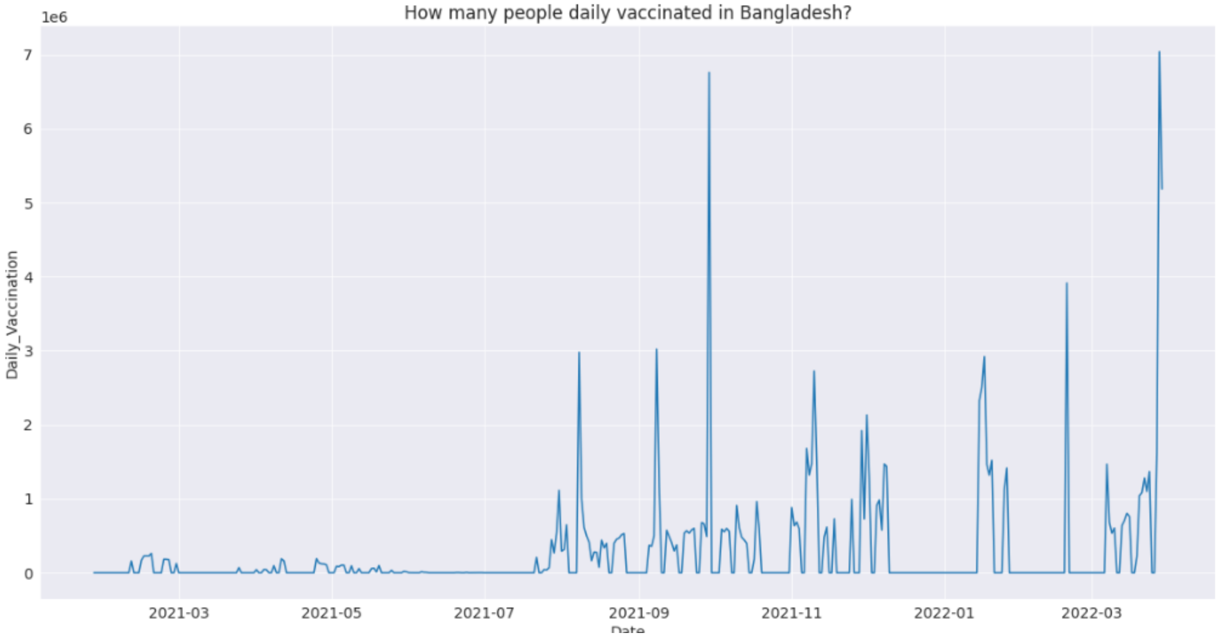
```
df2 = df.copy()

fig= go.Figure(data=[

    go.Bar(

        name='Total Vaccinations',

        x=df2['date'],

        y=df2['total_vaccinations'],

        marker_color = 'crimson  ),

 go.Bar(name='People Vaccinated',

        x=df2['date'],

        y=df2['people_vaccinated'],

        marker_color = 'green),])

fig.update_layout( title="Total vaccinations vs people vaccinated",
    xaxis_title = 'Date',

    yaxis_title = 'Number of count',

    barmode='stack',

    hovermode='x')fig.show()
```



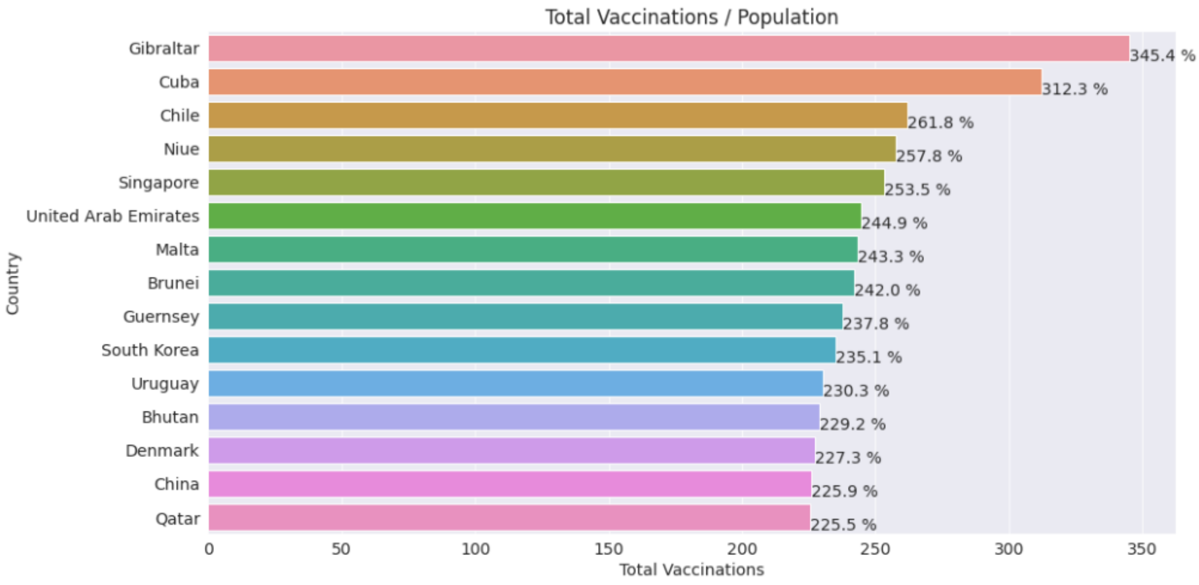
```
plt.figure(figsize=(12,8))

ax = sns.barplot(x=daily_vaccinations_per_million,
y=daily_vaccinations_per_million.index )

plt.xlabel("daily vaccinations per million")

plt.ylabel("Country")

plt.title("Daily COVID-19 vaccine doses administered per million people");



for patch in ax.patches:

    width = patch.get_width()
    height = patch.get_height()

    x = patch.get_x()

    y = patch.get_y()

      plt.text(width + x, height + y, '{:.1f} '.format(width))
```



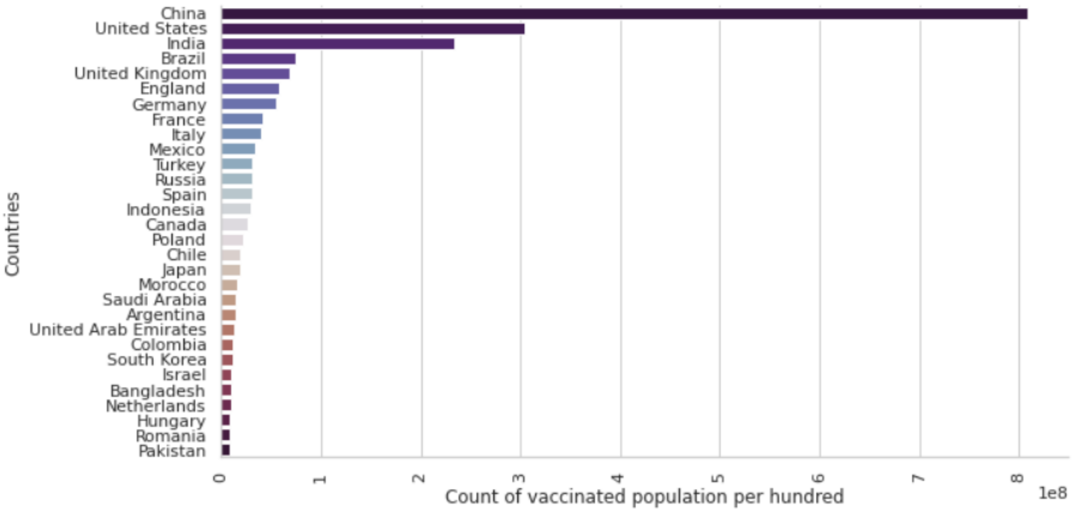
```
plt.figure(figsize=(20,10))
sns.lineplot(x=bangladesh_df.date, y=bangladesh_df.daily_vaccinations_raw)
plt.xlabel("Date")
plt.ylabel("Daily_Vaccination")
plt.title('How many people daily vaccinated in Bangladesh?');
```



```
plt.figure(figsize= (15, 8))
ax = sns.barplot(x=population_country, y=population_country.index)
plt.title('Total Vaccinations / Population')
plt.xlabel('Total Vaccinations')
plt.ylabel('Country')

for patch in ax.patches:
    width = patch.get_width()
    height = patch.get_height()
    x = patch.get_x()
    y = patch.get_y()

    plt.text(width + x, height + y, '{:.1f} %'.format(width))
```

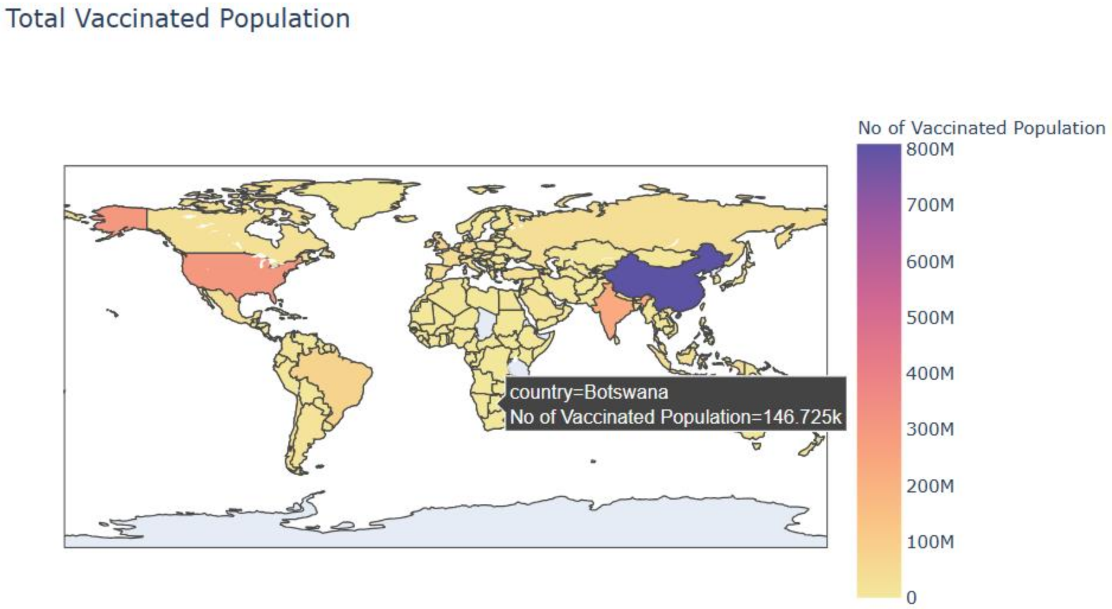


```
sns.catplot( x='total_vaccinations',  y=vacc_data30.country ,data=vacc_data30,kind='bar',ci=None,palette='twilight_shifted', legend_out=False,aspect=2, orient='h')
plt.xlabel('Count of vaccinated population per hundred')
plt.ylabel('Countries')
plt.xticks(rotation=90)
plt.show()
```

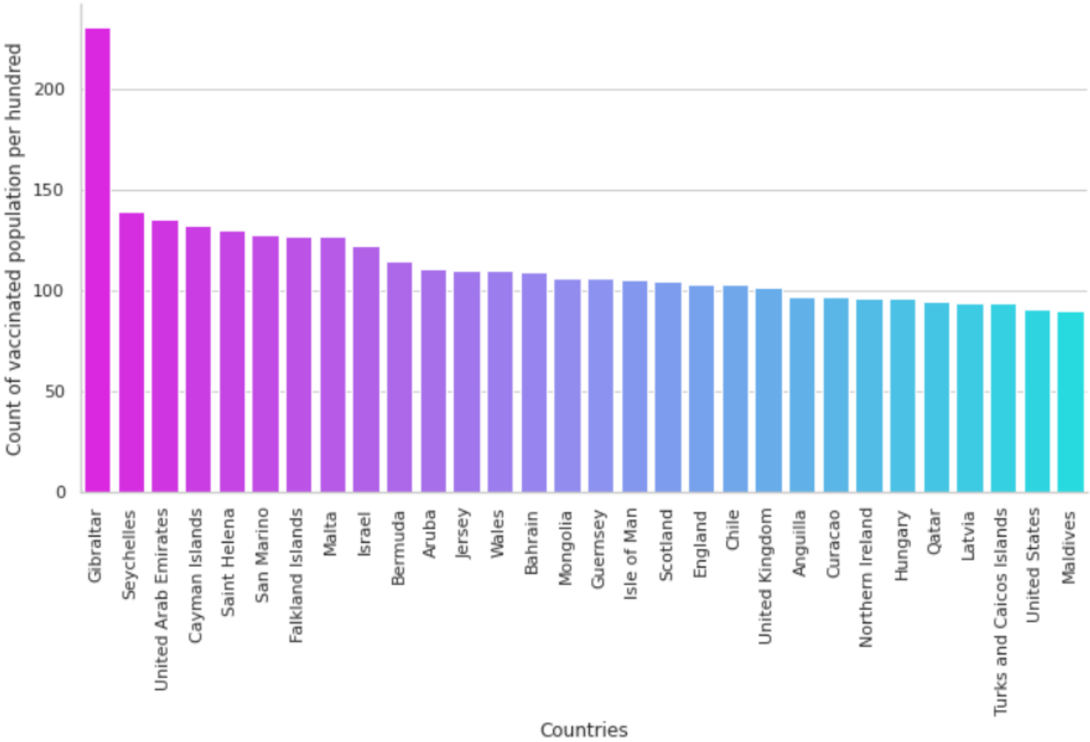


```
vaccince=vaccince_df[cols].groupby('country').max().sort_values('total_vaccinations', ascending=False)
```

```
fig = px.choropleth(locations=vacc_data.index, locationmode='country names' ,
                    data_frame=vaccince_data,

                    color='total_vaccinations', title='Total Vaccinated Population',

                    labels={'total_vaccinations':"No of Vaccinated Population"},color_continuous_scale='sunset'

                   )

fig.show('notebook')
```


Total Vaccinated Population
No of Vaccinated Population
800M
700M
600M
500M
400M
300M
200M
100M
0
country=Botswana
No of Vaccinated Population=146.725k

```
cols = ['country','total_vaccinations_per_hundred']

vacc_per_hund30 =vacc_df[cols].groupby('country').max()

vacc_per_hund30=vacc_per_hund30.sort_values('total_vaccinations_per_hundred', ascending=False).head(30).reset_index()



sns.catplot(data=vacc_per_hund30, x=vacc_per_hund30.country, y='total_vaccinations_per_hundred',kind='bar',palette='cool_r' ,ci=None, legend_out=False,aspect =2)

plt.ylabel('Count of vaccinated population per hundred')
plt.xlabel('Countries')

plt.xticks(rotation=90)

plt.show()
```

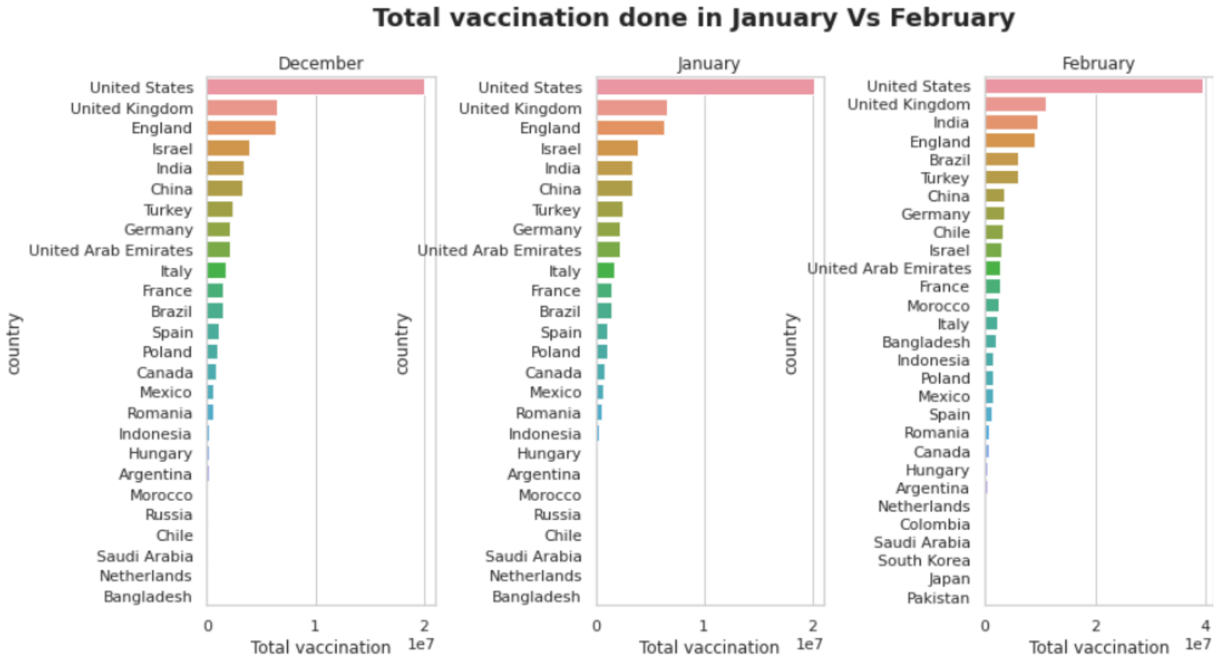


```
cols= ['country','daily_vaccinations','date']

start_date = '2020-01-01'

end_date = '2020-01-31'

vacc_daily_dec=vacc_df[vacc_df['country'].isin(country)] # get data for only those country that are present in top 30

mask = (vacc_daily_dec['date'] > start_date) & (vacc_daily_dec['date'] <= end_date)

vacc_daily_dec=vacc_daily_dec[mask] # filter data according to date

vacc_daily_dec=vacc_daily_dec[cols].groupby('country').sum()
```

```
vacc_daily_dec=vacc_daily_dec.sort_values('daily_vaccinations', ascending = False ).reset_index()

#Countrywise sum all daily vaccinations done in month of January

country=vacc_data30.country  # get top 30 countries from data set

cols = ['country','daily_vaccinations','date']

start_date = '2021-01-01'

end_date = '2021-01-31'

vacc_daily_jan=vacc_df[vacc_df['country'].isin(country)] # get data for only those country that are present in top 30

mask = (vacc_daily_jan['date'] > start_date) & (vacc_daily_jan['date'] <= end_date)

vacc_daily_jan=vacc_daily_jan[mask] # filter data according to date

vacc_daily_jan=vacc_daily_jan[cols].groupby('country').sum()

vacc_daily_jan=vacc_daily_jan.sort_values('daily_vaccinations', ascending = False ).reset_index()


#Countrywise sum all daily vaccinations done in month of February

cols = ['country','daily_vaccinations','date']

start_date = '2021-02-01'

end_date = '2021-02-28'



vacc_daily_feb=vacc_df[vacc_df['country'].isin(country)]
```

```
mask = (vacc_daily_feb['date'] > start_date) & (vacc_daily_feb['date'] <=
end_date)

vacc_daily_feb=vacc_daily_feb[mask]

vacc_daily_feb=vacc_daily_feb[cols].groupby('country').sum()

vacc_daily_feb=vacc_daily_feb.sort_values('daily_vaccinations', ascending =
False ).reset_index()
```

```
fig1, axes1 =plt.subplots(1,3,figsize=(13, 7))

fig1.suptitle('Total vaccination done in January Vs February', fontsize=18,
fontweight='bold')

plt.subplots_adjust(wspace=0.7)

ax=sns.barplot(data=vacc_daily_jan, x='daily_vaccinations',
y='country',ax=axes1[0],orient='h').set(

   title='December',xlabel='Total vaccination')

ax1=sns.barplot(data=vacc_daily_jan, x='daily_vaccinations',
y='country',ax=axes1[1],orient='h').set(

   title='January',xlabel='Total vaccination')

ax2=sns.barplot(data=vacc_daily_feb, x='daily_vaccinations',
y='country',ax=axes1[2],orient='h').set(title='February',xlabel='Total vaccination')

plt.show()
```



## Conclusion:

In Conclusion, we can take look at the Dashboard for further Analysis.

In China and India in these two countries, most people are Vaccinated.

In 2021 60.79% of people are fully Vaccinated and in 2020 only 39.2 % of people are fully Vaccinated.

China, India, the United States, Brazil, Indonesia, Germany, the United States, Turkey, France, and England There are the top 10 countries is completed the full Vaccinations.