covid19-vaccine analysis phase 3

Introduction:

The COVID-19 pandemic, also known as the coronavirus pandemic, is an ongoing global pandemic of coronavirus disease 2019 (COVID-19) caused by severe acute respiratory syndromecoronavirus 2 (SARS-CoV-2). It was first identified in December 2019 in Wuhan, China. TheWorld Health Organization declared the outbreak a Public Health Emergency of InternationalConcern on 30 January 2020, and later a pandemic on 11 March 2020. As of 8 April 2021, morethan 133 million cases have been confirmed, with more than 2.89 million deaths attributed toCOVID-19, making it one of the deadliest pandemics in history. Symptoms of COVID-19 arehighly variable, ranging from none to life-threatening illness. The virus appears to spread quicklyamong people, and more continue to be discovered over time about how it applies.

Dataset link:

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

Dataset has columns like country, iso_code, date, total_vaccinations, people_vaccinated, people_fully vaccinated, etc. An initial look at the above table shows that data has null values too. We will deal with null values later.

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
0	Albania	ALB	2021-01-10	0.0	0.0	NaN	NaN
1	Albania	ALB	2021-01-11	NaN	NaN	NaN	NaN
2	Albania	ALB	2021-01-12	128.0	128.0	NaN	NaN
3	Albania	ALB	2021-01-13	188.0	188.0	NaN	60.0
4	Albania	ALB	2021-01-14	266.0	266.0	NaN	78.0

info() function is used to get the overview of data like data type of feature, a number of null values in each column, and many more.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4568 entries, 0 to 4567
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   country                               4568 non-null   object
 1   iso_code                              4260 non-null   object
 2   date                                  4568 non-null   object
 3   total_vaccinations                    2988 non-null   float64
 4   people_vaccinated                     2541 non-null   float64
 5   people_fully_vaccinated               1702 non-null   float64
 6   daily_vaccinations_raw                2523 non-null   float64
 7   daily_vaccinations                    4409 non-null   float64
 8   total_vaccinations_per_hundred        2988 non-null   float64
 9   people_vaccinated_per_hundred         2541 non-null   float64
 10  people_fully_vaccinated_per_hundred   1702 non-null   float64
 11  daily_vaccinations_per_million        4409 non-null   float64
 12  vaccines                              4568 non-null   object
 13  source_name                           4568 non-null   object
 14  source_website                        4568 non-null   object
dtypes: float64(9), object(6)
memory usage: 535.4+ KB
```

The above picture shows that there are many null values in our dataset. We will deal with these null values later in this blog. There are two data types as seen from the table object means string and float.

The below function is used to get the total count of null values in each feature.

df.isnull().sum()

```
country                                0
iso_code                             308
date                                   0
total_vaccinations                  1580
people_vaccinated                   2027
people_fully_vaccinated             2866
daily_vaccinations_raw              2045
daily_vaccinations                   159
total_vaccinations_per_hundred      1580
people_vaccinated_per_hundred       2027
people_fully_vaccinated_per_hundred 2866
daily_vaccinations_per_million       159
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

DATA CLEANING

Dataset has many null values as we have seen before. To get rid of it we need to clean the data first, After cleaning we will perform our further analysis. For cleaning the dataset we will perform many steps. Some of these steps are shown below

1.Handling and Filling null values

2.Change the data type of features

3.Handling strings like splitting.

Check the below code for all the data cleaning that we are performing here:

```
df.fillna(value = 0, inplace = True)

df.total_vaccinations = df.total_vaccinations.astype(int)
```

```
df.people_vaccinated = df.people_vaccinated.astype(int)

df.people_fully_vaccinated = df.people_fully_vaccinated.astype(int)

df.daily_vaccinations_raw = df.daily_vaccinations_raw.astype(int)

df.daily_vaccinations = df.daily_vaccinations.astype(int)

df.total_vaccinations_per_hundred = df.total_vaccinations_per_hundred.astype(int)

df.people_fully_vaccinated_per_hundred =
df.people_fully_vaccinated_per_hundred.astype(int)

df.daily_vaccinations_per_million = df.daily_vaccinations_per_million.astype(int)

df.people_vaccinated_per_hundred = df.people_vaccinated_per_hundred.astype(int)

date = df.date.str.split('-', expand =True)

date
```

	0	1	2
0	2021	01	10
1	2021	01	11
2	2021	01	12
3	2021	01	13
4	2021	01	14
...	...	...	...
4563	2021	02	24
4564	2021	02	25
4565	2021	02	26
4566	2021	02	27
4567	2021	02	28

```
df['year'] = date[0]

df['month'] = date[1]

df['day'] = date[2]

df.year = pd.to_numeric(df.year)

df.month = pd.to_numeric(df.month)

df.day = pd.to_numeric(df.day)

df.date = pd.to_datetime(df.date)

df.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
0	Albania	ALB	2021-01-10	0	0	0	0
1	Albania	ALB	2021-01-11	0	0	0	0
2	Albania	ALB	2021-01-12	128	128	0	0
3	Albania	ALB	2021-01-13	188	188	0	60
4	Albania	ALB	2021-01-14	266	266	0	78

Data points start from 2020-12-08

Data points end at 2021-02-28

Total Number of countries in the data set = 117

Total Number of Unique Vaccines in the data set = 22

df.info()

```
RangeIndex: 4568 entries, 0 to 4567
Data columns (total 18 columns):
 #   Column                               Non-Null Count  Dtype         
---  ------                               --------------  -----         
 0   country                              4568 non-null   object        
 1   iso_code                             4568 non-null   object        
 2   date                                 4568 non-null   datetime64[ns]
 3   total_vaccinations                   4568 non-null   int64         
 4   people_vaccinated                    4568 non-null   int64         
 5   people_fully_vaccinated              4568 non-null   int64         
 6   daily_vaccinations_raw               4568 non-null   int64         
 7   daily_vaccinations                   4568 non-null   int64         
 8   total_vaccinations_per_hundred       4568 non-null   int64         
 9   people_vaccinated_per_hundred        4568 non-null   int64         
 10  people_fully_vaccinated_per_hundred  4568 non-null   int64         
 11  daily_vaccinations_per_million       4568 non-null   int64         
 12  vaccines                             4568 non-null   object        
 13  source_name                          4568 non-null   object        
 14  source_website                       4568 non-null   object        
 15  year                                 4568 non-null   int64         
 16  month                                4568 non-null   int64         
 17  day                                  4568 non-null   int64         
dtypes: datetime64[ns](1), int64(12), object(5)
memory usage: 642.5+ KB
```

Total Vaccinated Till Date

In this section, we are going to see how many total vaccines have been used in each country. Check the below code 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 includes 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
```

Total vaccinated

	total_vaccinted_till_date
United States	68767620
China	34922496
United Kingdom	19660299
England	16602591
India	13483116
...	...
Trinidad and Tobago	441
Venezuela	155
Saint Helena	0
San Marino	0
Greenland	0

For analyzing data, we need some libraries. In this section, we are importing all the

required libraries like pandas, NumPy, matplotlib, plotly, seaborn, and word cloud that are required for data analysis. Check the below code to import all the required libraries.

print('Data point starts from ',df.date.min(),'n')

print('Data point ends at ',df.date.max(),'n')

print('Total no of countries in the data set ',len(df.country.unique()),'n')

print('Total no of unique vaccines in the data set ',len(df.vaccines.unique()),'n')

features| Covid Vaccination Progress

Data points start from 2020-12-08

Data points end at 2021-02-28

Total Number of countries in the data set = 117

Total Number of Unique Vaccines in the data set = 22

df.info()

info of clean data

DATA VISUALIZATION

In this section, we are going to draw some visuals to get insights from our dataset. So let’s started.

describe() function in pandas used to get the statistics of each feature present in our dataset. Some of the information we get include count, max, min, standard deviation, median, etc.

df.describe()

Covid Vaccination Progress | describe dataset

unique() function in pandas helps to get unique values present in the feature.

df.country.unique()

Unique country values

```
def size(m,n):
    fig = plt.gcf();
    fig.set_size_inches(m,n);
```

Word Art of Countries

Word Cloud is a unique way to get information from our dataset. The words are shown in the form of art where the size proportional depends on how much the particular word repeated in the dataset. This is made by using the WordCloud library. Check the below code on how to draw word cloud

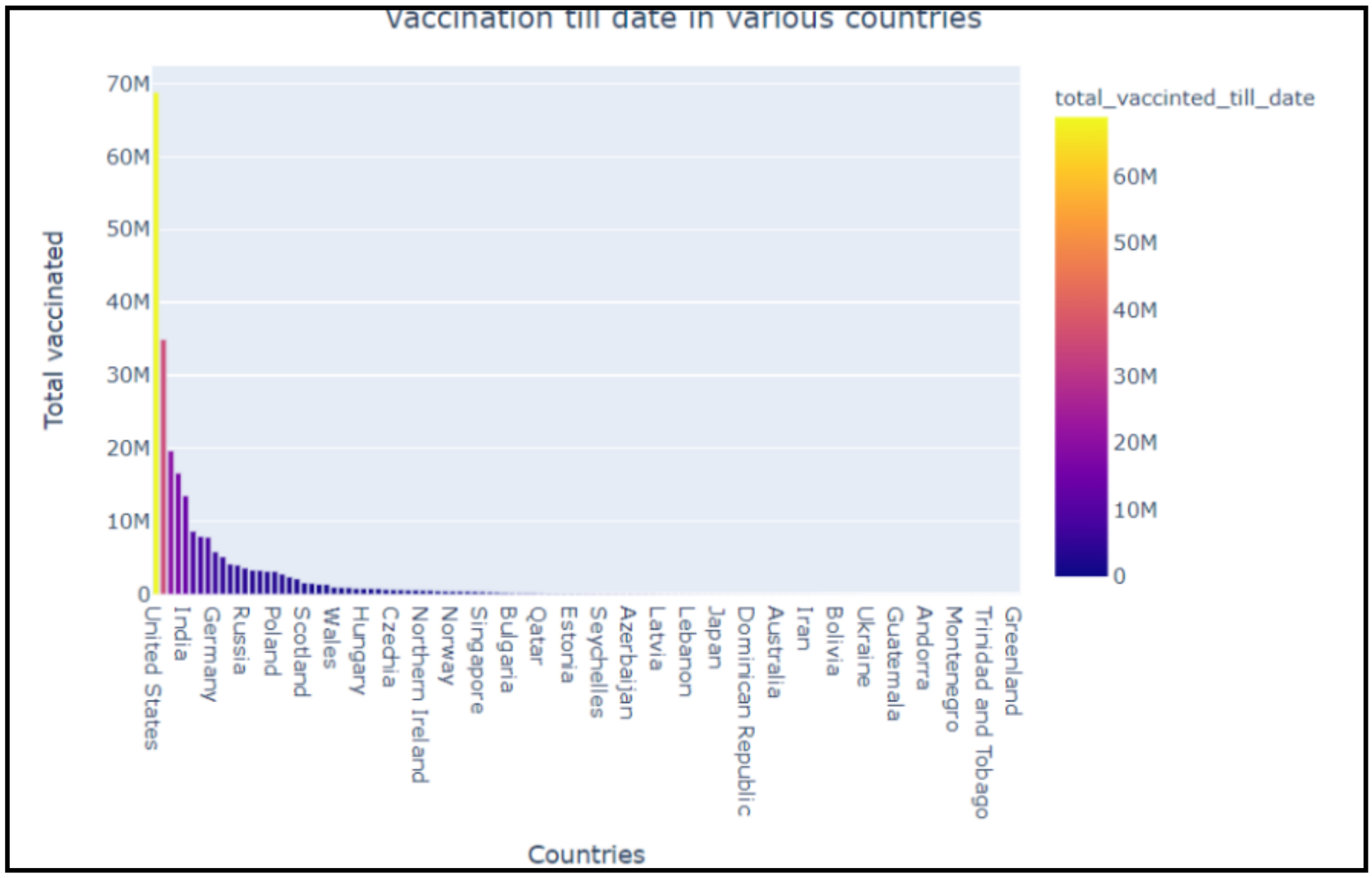
```
wordCloud = WordCloud(
    background_color='white',
    max_font_size = 50).generate(' '.join(df.country))
plt.figure(figsize=(15,7))
plt.axis('off')
plt.imshow(wordCloud)
plt.show()
```


Total Vaccinated Till Date

In this section, we are going to see how many total vaccines have been used in each country. Check the below code 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 includes 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
Total vaccinated
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()
```

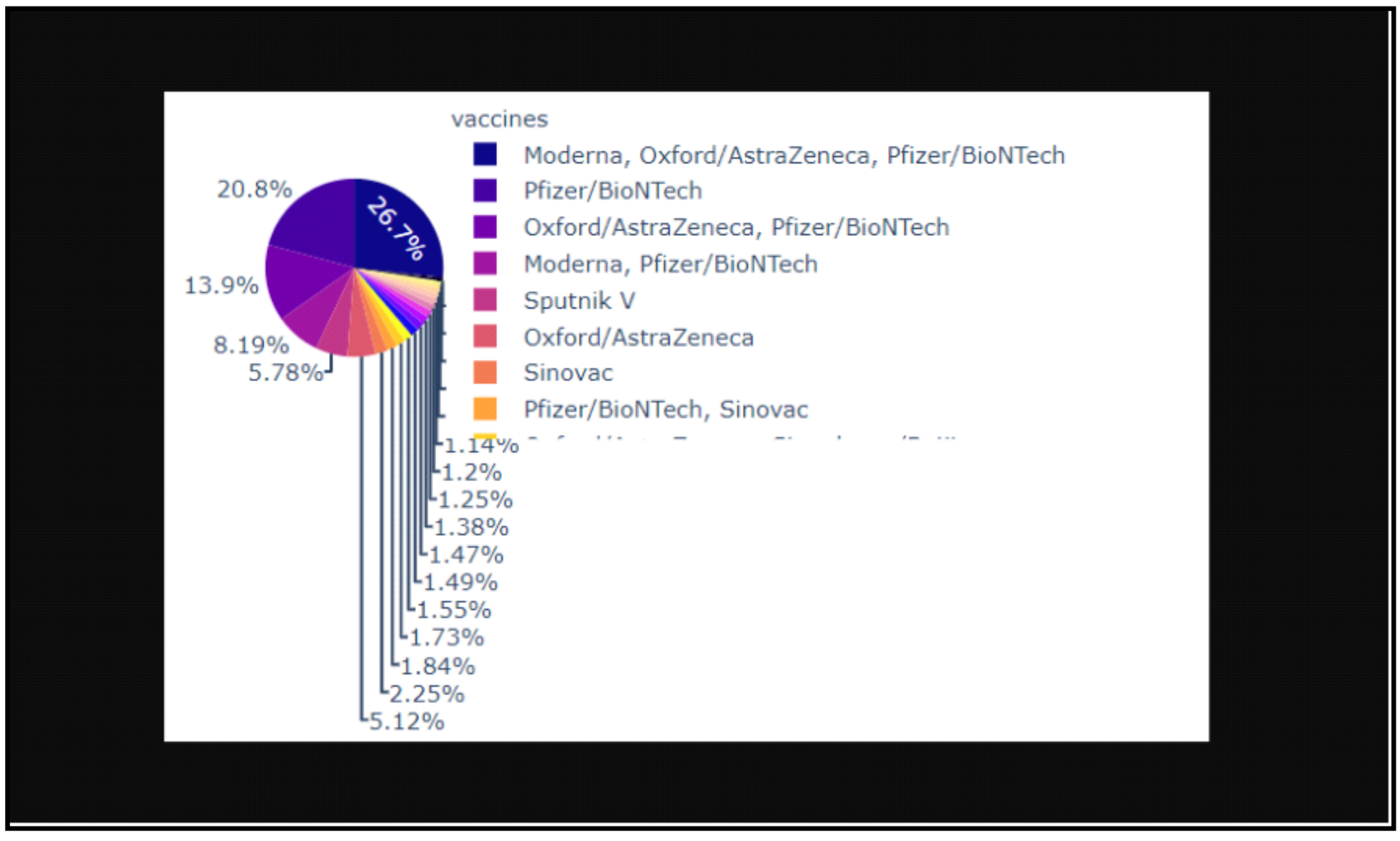


Pie-Plot

In this section, we are going to draw pip-plots. For more details check the below code:

```
def plot_pie(value, title, color) :
    new_dict = {}
    for v in df[value].unique() :
        value_count = 0
        for i in range(len(df)) :
            if df[value][i] == v :
                value_count += 1
        new_dict[v] = value_count
```

```
#     print(new_dict)
    new_df = pd.DataFrame.from_dict(new_dict, orient = 'index', columns = ['Total'])
    if color == 'plasma' :
        fig = px.pie(new_df, values= 'Total',
                     names = new_df.index,
                     title = title,
                     color_discrete_sequence=px.colors.sequential.Plasma)
elif color == 'rainbow' :
    fig = px.pie(new_df, values= 'Total',
                     names = new_df.index,
                     title = title,
                     color_discrete_sequence=px.colors.sequential.Rainbow)
else :
    fig = px.pie(new_df, values= 'Total',
                     names = new_df.index,
                     title = title)
fig.update_layout(
    title={
        'y':0.95,
        'x':0.5
    },
    legend_title = value
)
return fig.show()
plot_pie('vaccines', 'Various vaccines and their uses', 'plasma')
```



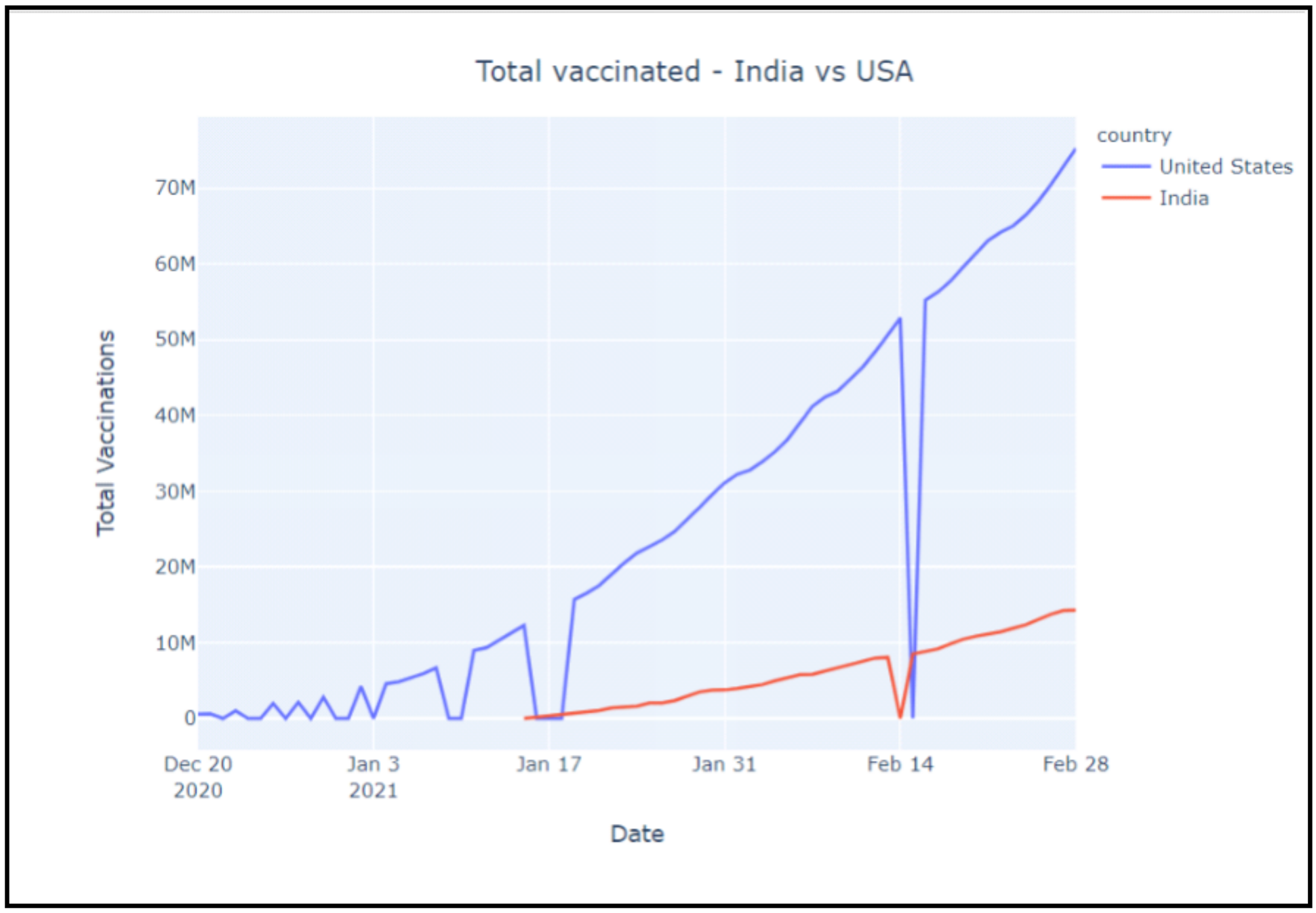
Total vaccinated – India vs the USA

In this section, we will see what is the trend of vaccination among two great countries India and the USA. We are going to draw a line plot where the x-axis is Date and the y-axis is daily vaccination. Check the below code for more information:

```
india_usa = [df[df.country == 'United States'], df[df.country == 'India']]

result = pd.concat(india_usa)

fig = px.line(result, x = 'date', y ='total_vaccinations', color = 'country')

fig.update_layout(
    title={
        'text' : "Total vaccinated - India vs USA",
        'y':0.95,
        'x':0.5
    },
xaxis_title="Date",
yaxis_title="Total Vaccinations"
```

)

fig.show()

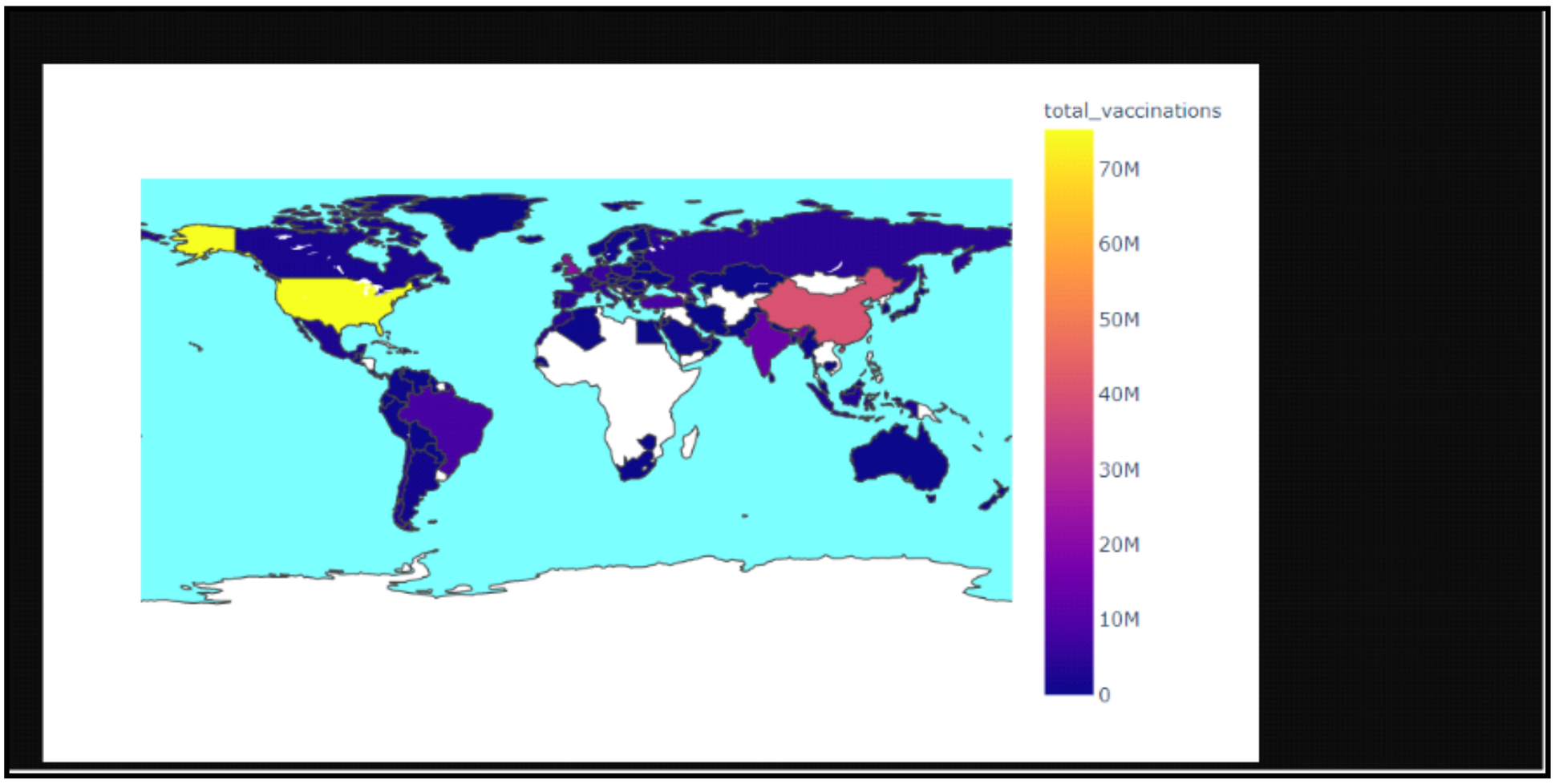


MAPS

In this section, we are going to see how vaccinations are going in different countries using maps. The colour signifies how many people have been vaccinated. Check the below maps for more details.

Most vaccinated country

plot_map('total_vaccinations','Most vaccinated country', None)


total_vaccinations
70M
60M
50M
40M
30M
20M
10M
0