COVID 19 EXPLORATORY DATA ANALYSIS

Covid-19 Cases and Vaccinations

Image Source

Introduction

This is an Exploratory Data Analysis project on Covid-19 Cases and Vaccination.

The Aim of the Project is to extract the characteristic results from the Data set and conduct comparative analysis of cases and vaccination among the countries.

The essential Questions formulated are related to Set of Vaccines are preferred by Countries, daily vaccinations, positive rate, total confirmed, active and death cases. There are several Visualizations regarding Leading Countries in highest number of total confirmed cases, serious critical cases and total Covid-19 deaths

TABLE OF CONTENTS

1. READING THE DATASET

Importing all the Packages required for the 'Exploratory Data Analysis'.

In [15]:
```
import os
import numpy as np
import pandas as pd
```

```
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px
```

Loading the Datasets for the Analysis:-

Loading the required Dataset from my github which is downloaded from https://www.kaggle.com/josephassaker/covid19-global-dataset and https://www.kaggle.com/gpreda/covid-world-vaccination-progress

With the help of pandas function `pd.read_csv()` , the Dataset csv file is opened and read.

In [6]:
```
vaccine_data_df= pd.read_csv('https://raw.githubusercontent.com/MrinalAnand227/COVI
```

In [7]:
```
covid_data_df = pd.read_csv('https://raw.githubusercontent.com/MrinalAnand227/COVID
```

In [8]:
```
#Extremely Important for further analysis and cleaning
summary_data_df = pd.read_csv('https://raw.githubusercontent.com/MrinalAnand227/COV
```

In [16]:
```
type(vaccine_data_df)
```

Out[16]:
```
pandas.core.frame.DataFrame
```

In [10]:
```
type(covid_data_df)
```

Out[10]:
```
pandas.core.frame.DataFrame
```

In [17]:
```
type(summary_data_df)
```

Out[17]:
```
pandas.core.frame.DataFrame
```

Thus the dataset in csv format is now stored in the form of `DataFrame`

In [18]:
```
vaccine_data_df
```

Out[18]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN
...	...	...	...	...	...	...
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0

86512 rows × 15 columns

In [19]:
```
covid_data_df
```

Out[19]:

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_
0	2020-2-15	Afghanistan	0.0	NaN	0.0	
1	2020-2-16	Afghanistan	0.0	NaN	0.0	
2	2020-2-17	Afghanistan	0.0	NaN	0.0	
3	2020-2-18	Afghanistan	0.0	NaN	0.0	
4	2020-2-19	Afghanistan	0.0	NaN	0.0	
...	...	...	...	...	...	
184782	2022-5-10	Zimbabwe	248642.0	106.0	963.0	
184783	2022-5-11	Zimbabwe	248778.0	136.0	1039.0	
184784	2022-5-12	Zimbabwe	248943.0	165.0	1158.0	
184785	2022-5-13	Zimbabwe	249131.0	188.0	1283.0	
184786	2022-5-14	Zimbabwe	249206.0	75.0	1307.0	

184787 rows × 7 columns

In [20]:
```
summary_data_df
```

Out[20]:

	country	continent	total_confirmed	total_deaths	total_recovered	active_cases	ser
0	Afghanistan	Asia	179267	7690.0	162202.0	9375.0	
1	Albania	Europe	275574	3497.0	271826.0	251.0	
2	Algeria	Africa	265816	6875.0	178371.0	80570.0	
3	Andorra	Europe	42156	153.0	41021.0	982.0	
4	Angola	Africa	99194	1900.0	97149.0	145.0	
...	...	...	...	...	...	...	
221	Wallis And Futuna Islands	Australia/Oceania	454	7.0	438.0	9.0	
222	Western Sahara	Africa	10	1.0	9.0	0.0	
223	Yemen	Asia	11819	2149.0	9009.0	661.0	
224	Zambia	Africa	320591	3983.0	315997.0	611.0	
225	Zimbabwe	Africa	249206	5482.0	242417.0	1307.0	

226 rows × 12 columns

2. DATA TREATMENT: CLEANING AND FORMATTING

Data cleaning routines work to "clean" the data vy filling in missing vakues,smoothing noise data,identifying or removing outliers, and resolving inconsistencies.

Real world Data tend to be incomplete, noisy and inconsistent. Data cleaning(or Data cleanzing) routines attempt to fill in missing values smooth out noise while while identifying outliers and correct inconsistencies in the Data.

2.1 Lets Analyse the Shape and of the DataFrame i.e. Columns and Rows

In [21]:
```
print('Vaccination Dataset')
print('Rows: ',vaccine_data_df.shape[0])
print('Column: ',vaccine_data_df.shape[1])
print('Total size: ',vaccine_data_df.size)
```

```
Vaccination Dataset
Rows:  86512
Column:  15
Total size:  1297680
```

In [22]:
```
print('World Covid-19 Dataset')
print('Rows: ',covid_data_df.shape[0])
print('Column: ', covid_data_df.shape[1])
print('Total size: ',covid_data_df.size)
```

```
World Covid-19 Dataset
Rows:  184787
Column:  7
Total size:  1293509
```

In [23]:
```
print('Summary Covid-19 Dataset')
print('Rows: ',summary_data_df.shape[0])
print('Column: ',summary_data_df.shape[1])
print('Total size: ',summary_data_df.size)
```

```
Summary Covid-19 Dataset
Rows:  226
Column:  12
Total size:  2712
```

2.2 Adding new columns of Year, Month , Day and Weekday

In [24]:
```
covid_data_df['year'] = pd.DatetimeIndex(covid_data_df.date).year
covid_data_df['month'] = pd.DatetimeIndex(covid_data_df.date).month
covid_data_df['day'] = pd.DatetimeIndex(covid_data_df.date).day
covid_data_df['weekday'] = pd.DatetimeIndex(covid_data_df.date).weekday
```

In [25]:
```
vaccine_data_df['year'] = pd.DatetimeIndex(vaccine_data_df.date).year
vaccine_data_df['month'] = pd.DatetimeIndex(vaccine_data_df.date).month
vaccine_data_df['day'] = pd.DatetimeIndex(vaccine_data_df.date).day
vaccine_data_df['weekday'] = pd.DatetimeIndex(vaccine_data_df.date).weekday
```

2.3 Searching those countries in Covid-19 Vaccination Dataset which are not present in the Summary Dataset

In [26]:
```
print("Countries in Covid-19 Vaccination Dataset which are not present in the Summa
print([x for x in vaccine_data_df.country.unique() if x not in summary_data_df.cou
```

```
Countries in Covid-19 Vaccination Dataset which are not present in the Summary Dat
aset
['Antigua and Barbuda', 'Bonaire Sint Eustatius and Saba', 'Bosnia and Herzegovin
a', 'Brunei', 'Cape Verde', "Cote d'Ivoire", 'Czechia', 'Democratic Republic of Co
ngo', 'England', 'Eswatini', 'Falkland Islands', 'Guernsey', 'Guinea-Bissau', 'Hon
g Kong', 'Isle of Man', 'Jersey', 'Kosovo', 'Macao', 'North Macedonia', 'Northern
Cyprus', 'Northern Ireland', 'Palestine', 'Pitcairn', 'Saint Kitts and Nevis', 'Sa
int Vincent and the Grenadines', 'Sao Tome and Principe', 'Scotland', 'Sint Maarte
n (Dutch part)', 'Timor', 'Tokelau', 'Trinidad and Tobago', 'Turkmenistan', 'Turks
and Caicos Islands', 'Tuvalu', 'United Kingdom', 'United States', 'Vietnam', 'Wale
s', 'Wallis and Futuna']
```

Analyzing the above list there are some countries whoes names are needed to be replaced like

United States = USA

United Kingdom = UK

North Cyprus = Cyprus

Czechia = Czech Republic

and The United Kingdom (UK) is made up of England, Scotland, Wales and Northern Ireland.

Thus we have to drop those countries for appropriate analysis

In [27]:
```
vaccine_data_df.country = vaccine_data_df.country.replace().replace({
    "United States": "USA",
    "United Kingdom": "UK",
    "Republic of Ireland": "Ireland",
```

```
    "Northern Cyprus" : "Cyprus",
    "Czechia": "Czech Republic"
})

vaccine_data_df = vaccine_data_df[vaccine_data_df.country.apply(lambda x: x not in
```

2.4 Adding Vaccine Dataset summarized columns in the summary Dataset

In [28]:
```
def aggregate(df: pd.Series, agg_col: str) -> pd.DataFrame:
    data = df.groupby("country")[agg_col].max()
    data = pd.DataFrame(data)
    return data

columns_tobe_added = ['total_vaccinations','people_vaccinated','people_fully_vacci
final_summary_data_df = summary_data_df.set_index("country")
covid_data_df.set_index("country")

type_vaccine_data_df = vaccine_data_df[['country', 'vaccines']].drop_duplicates().
final_summary_data_df = summary_data_df.join(type_vaccine_data_df)
for column in columns_tobe_added:
    final_summary_data_df = final_summary_data_df.join(aggregate(vaccine_data_df,
```

2.5 Adding new columns of `Percentage Vaccinated` and `Positive Rate` in the Summary Dataset

In [29]:
```
positive_rate = final_summary_data_df.total_confirmed / final_summary_data_df.tota
final_summary_data_df['positive_rate'] = positive_rate
percentage_vaccinated = final_summary_data_df.total_vaccinations/ final_summary_da
final_summary_data_df['pecentage_vaccinated'] = percentage_vaccinated
```

2.6 Missing values (NaN values)

In order to avoid misleading calculation, the missing values need to be resolved.

Let's check the NaN values in the Data

In [33]:
```
final_summary_data_df.isna().sum().sort_values(ascending=False)
```

Out[33]:
```
pecentage_vaccinated                  226
vaccines                              226
people_fully_vaccinated_per_hundred   226
people_vaccinated_per_hundred         226
total_vaccinations_per_hundred        226
people_fully_vaccinated               226
people_vaccinated                     226
total_vaccinations                    226
serious_or_critical                    81
total_recovered                        22
active_cases                           22
positive_rate                          14
total_tests_per_1m_population          14
total_tests                            14
total_deaths_per_1m_population          8
total_deaths                            8
population                              0
continent                               0
total_cases_per_1m_population           0
total_confirmed                         0
country                                 0
dtype: int64
```

In [34]:
```
vaccine_data_df.isna().sum().sort_values(ascending=False)
```

Out[34]:
```
daily_vaccinations_raw                51107
people_fully_vaccinated               47680
people_fully_vaccinated_per_hundred   47680
people_vaccinated_per_hundred         45188
people_vaccinated                     45188
total_vaccinations                    42875
total_vaccinations_per_hundred        42875
daily_vaccinations                      295
daily_vaccinations_per_million          295
source_website                            0
day                                       0
month                                     0
year                                      0
country                                   0
source_name                               0
vaccines                                  0
iso_code                                  0
date                                      0
weekday                                   0
dtype: int64
```

In [35]:
```
covid_data_df.isna().sum().sort_values(ascending=False)
```

Out[35]:
```
daily_new_deaths           26937
active_cases               18040
daily_new_cases            10458
cumulative_total_deaths     6560
date                           0
country                        0
cumulative_total_cases         0
year                           0
month                          0
day                            0
weekday                        0
dtype: int64
```

There are columns with NaN values so removing them

In [36]:
```
final_summary_data_df['pecentage_vaccinated'].fillna(0, inplace=True)
```

In [37]:
```
final_summary_data_df['people_fully_vaccinated_per_hundred'].fillna(0, inplace=Tru
final_summary_data_df['people_vaccinated_per_hundred'].fillna(0, inplace=True)
final_summary_data_df['total_vaccinations_per_hundred'].fillna(0, inplace=True)
final_summary_data_df['people_fully_vaccinated'].fillna(0, inplace=True)
final_summary_data_df['people_vaccinated'].fillna(0, inplace=True)
final_summary_data_df['total_vaccinations'].fillna(0, inplace=True)
final_summary_data_df['serious_or_critical'].fillna(0, inplace=True)
final_summary_data_df['total_deaths_per_1m_population'].fillna(0, inplace=True)
final_summary_data_df['total_deaths'].fillna(0, inplace=True)
final_summary_data_df['total_tests_per_1m_population'].fillna(0, inplace=True)
final_summary_data_df['total_tests'].fillna(0, inplace=True)
final_summary_data_df['positive_rate'].fillna(0, inplace=True)
final_summary_data_df['active_cases'].fillna(0, inplace=True)
final_summary_data_df['total_recovered'].fillna(0, inplace=True)
final_summary_data_df['vaccines'].fillna(0, inplace=True)
```

Removing NaN values

In [42]:
```
covid_data_df['daily_new_deaths'].fillna(0, inplace=True)
covid_data_df['cumulative_total_deaths'].fillna(0, inplace=True)
covid_data_df['daily_new_cases'].fillna(0, inplace=True)
covid_data_df['active_cases'].fillna(0, inplace=True)
```

In [38]:
```
final_summary_data_df.isna().sum().sort_values(ascending=False)
```

Out[38]:
```
country                              0
population                           0
positive_rate                        0
people_fully_vaccinated_per_hundred  0
people_vaccinated_per_hundred        0
total_vaccinations_per_hundred       0
people_fully_vaccinated              0
people_vaccinated                    0
total_vaccinations                   0
vaccines                             0
total_tests_per_1m_population        0
continent                            0
total_tests                          0
total_deaths_per_1m_population       0
total_cases_per_1m_population        0
serious_or_critical                  0
active_cases                         0
total_recovered                      0
total_deaths                         0
total_confirmed                      0
pecentage_vaccinated                 0
dtype: int64
```

In [41]:
```
vaccine_data_df.isna().sum().sort_values(ascending=False)
```

Out[41]:
```
country                                 0
people_fully_vaccinated_per_hundred     0
day                                     0
month                                   0
year                                    0
source_website                          0
source_name                             0
vaccines                                0
daily_vaccinations_per_million          0
people_vaccinated_per_hundred           0
iso_code                                0
total_vaccinations_per_hundred          0
daily_vaccinations                      0
daily_vaccinations_raw                  0
people_fully_vaccinated                 0
people_vaccinated                       0
total_vaccinations                      0
date                                    0
weekday                                 0
dtype: int64
```

In [43]:
```
covid_data_df.isna().sum().sort_values(ascending=False)
```

Out[43]:
```
date                      0
country                   0
cumulative_total_cases    0
daily_new_cases           0
active_cases              0
cumulative_total_deaths   0
daily_new_deaths          0
year                      0
month                     0
day                       0
weekday                   0
dtype: int64
```

Thus there are no NaN values

2.7 Incorrect and Invalid Data

Usually real world data are filled manually sometimes so there is possibility that the values filled are incorrect due to human error. In order to detect such error we can use `.describe()` function of the Pandas Library

In [44]:
```
final_summary_data_df.describe()
```

Out[44]:

	total_confirmed	total_deaths	total_recovered	active_cases	serious_or_critical	total_cases
count	2.260000e+02	2.260000e+02	2.260000e+02	2.260000e+02	226.000000	
mean	2.305651e+06	2.782338e+04	2.037158e+06	6.193142e+04	172.898230	
std	7.575510e+06	9.806942e+04	7.262591e+06	2.241851e+05	718.311893	
min	2.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	
25%	2.412600e+04	1.875000e+02	8.559500e+03	6.050000e+01	0.000000	
50%	1.793750e+05	1.946500e+03	7.787100e+04	1.172000e+03	4.000000	
75%	1.090902e+06	1.323375e+04	8.581695e+05	1.468400e+04	42.750000	
max	8.420947e+07	1.026646e+06	8.124426e+07	1.938567e+06	8318.000000	

There seems to be a problem with active_cases column. As the minimum value formulated is negative which is impractical. A simple fix would be to ignore the rows where the values are negative(i.e. less than zero).This can be done using `.drop` Function

There are few negative values in data set which can neglected

Cleaned Datset

In [45]:
```
final_summary_data_df
```

Out[45]:

	country	continent	total_confirmed	total_deaths	total_recovered	active_cases	ser
0	Afghanistan	Asia	179267	7690.0	162202.0	9375.0	
1	Albania	Europe	275574	3497.0	271826.0	251.0	
2	Algeria	Africa	265816	6875.0	178371.0	80570.0	
3	Andorra	Europe	42156	153.0	41021.0	982.0	
4	Angola	Africa	99194	1900.0	97149.0	145.0	
...	...	...	...	...	...	...	
221	Wallis And Futuna Islands	Australia/Oceania	454	7.0	438.0	9.0	
222	Western Sahara	Africa	10	1.0	9.0	0.0	
223	Yemen	Asia	11819	2149.0	9009.0	661.0	
224	Zambia	Africa	320591	3983.0	315997.0	611.0	
225	Zimbabwe	Africa	249206	5482.0	242417.0	1307.0	

226 rows × 21 columns

In [46]:
```
covid_data_df
```

Out[46]:

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_
0	2020-2-15	Afghanistan	0.0	0.0	0.0	
1	2020-2-16	Afghanistan	0.0	0.0	0.0	
2	2020-2-17	Afghanistan	0.0	0.0	0.0	
3	2020-2-18	Afghanistan	0.0	0.0	0.0	
4	2020-2-19	Afghanistan	0.0	0.0	0.0	
...	...	...	...	...	...	
184782	2022-5-10	Zimbabwe	248642.0	106.0	963.0	
184783	2022-5-11	Zimbabwe	248778.0	136.0	1039.0	
184784	2022-5-12	Zimbabwe	248943.0	165.0	1158.0	
184785	2022-5-13	Zimbabwe	249131.0	188.0	1283.0	
184786	2022-5-14	Zimbabwe	249206.0	75.0	1307.0	

184787 rows × 11 columns

In [47]:
```
vaccine_data_df
```

Out[47]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated
0	Afghanistan	AFG	2021-02-22	0.0	0.0	0.0
1	Afghanistan	AFG	2021-02-23	0.0	0.0	0.0
2	Afghanistan	AFG	2021-02-24	0.0	0.0	0.0
3	Afghanistan	AFG	2021-02-25	0.0	0.0	0.0
4	Afghanistan	AFG	2021-02-26	0.0	0.0	0.0
...	...	...	...	...	...	...
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0

84740 rows × 19 columns

Final Size

In [48]:
```
print('Final Covid-19 Status Summary Dataset Shape and Size')
print('Rows: ',final_summary_data_df.shape[0])
print('Column: ',final_summary_data_df.shape[1])
print('Total size: ',final_summary_data_df.size)
```

```
Final Covid-19 Status Summary Dataset Shape and Size
Rows:  226
Column:  21
Total size:  4746
```

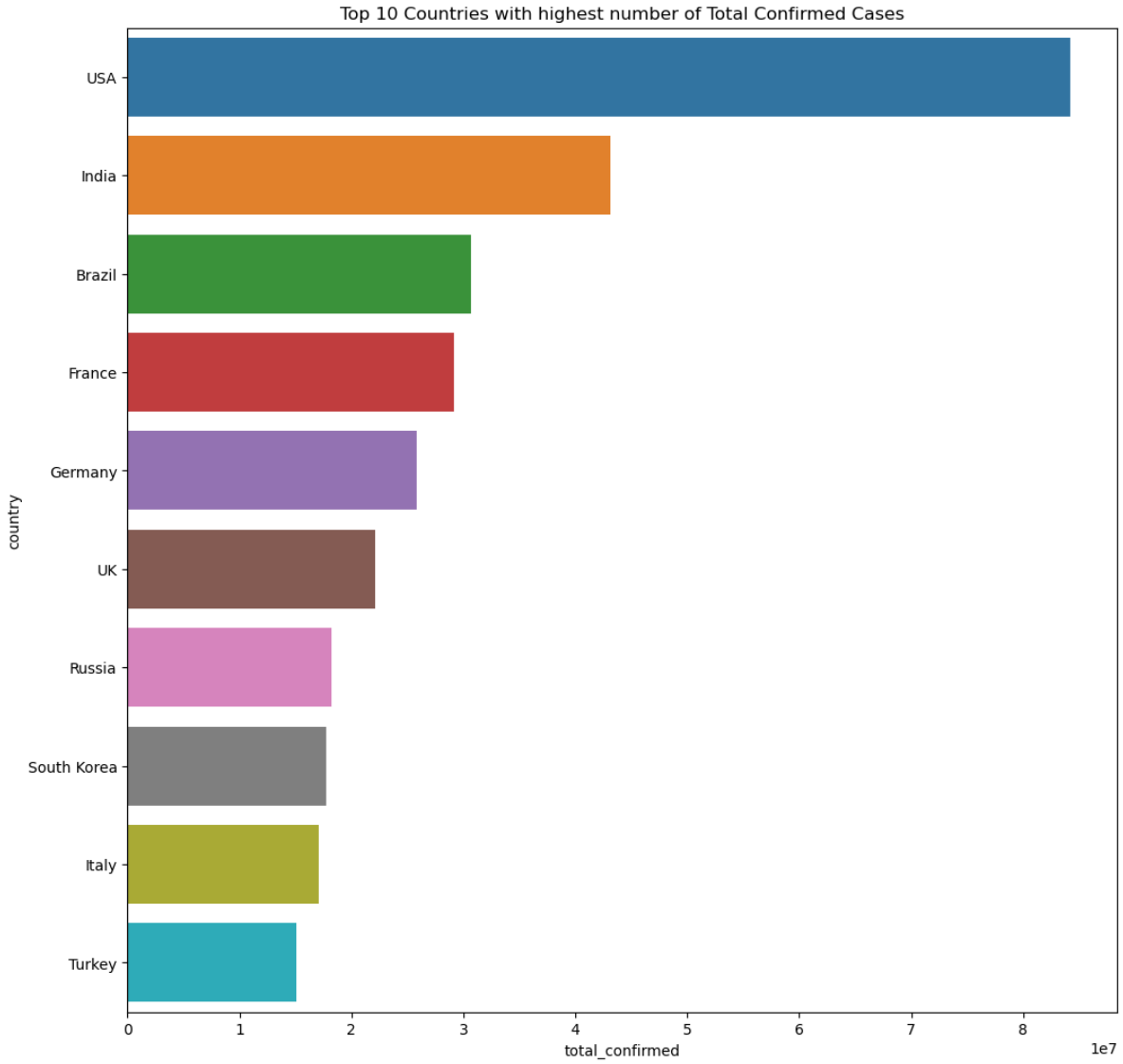
3. Exploratory Data Analysis (EDA) and Visualization. Quantitative and qualitative analysis

Data Visualization aims to communicate Data clearly and effectively through Graphical representation

1. Top 10 Countries with highest number of Total Confirmed Cases

In [49]:
```
plt.figure(figsize=(12,12))
top_cases_df= final_summary_data_df.sort_values('total_confirmed',ascending = Fals
sns.barplot(y = 'country',
            x = 'total_confirmed',
            data = top_cases_df);
plt.title('Top 10 Countries with highest number of Total Confirmed Cases')
```

Out[49]:
```
Text(0.5, 1.0, 'Top 10 Countries with highest number of Total Confirmed Cases')
```



The above graph shows that:-

1. USA has the highest number of Total Confirmed Cases.

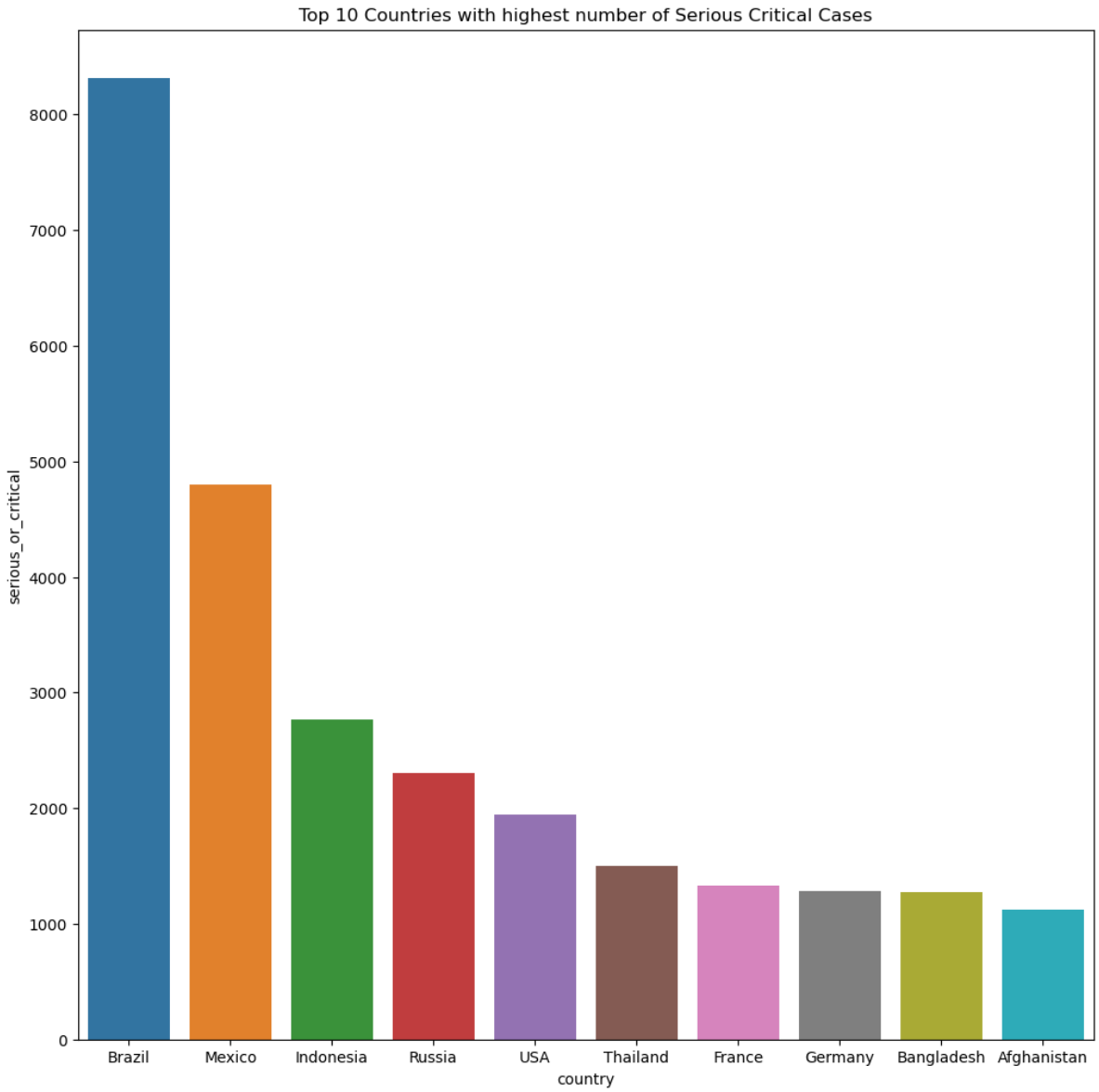
2. Brazil has the second highest number of Total Confirmed Cases.

3. India has the third highest number of Total Confirmed Cases.

2. Top 10 Countries with highest number of Serious Critical Cases

In [50]:
```
plt.figure(figsize=(12,12))
top_cases_df= final_summary_data_df.sort_values('serious_or_critical',ascending = 

sns.barplot(x = 'country',
            y = 'serious_or_critical',
            data = top_cases_df);
plt.title('Top 10 Countries with highest number of Serious Critical Cases');
```



The above graph shows that:-

1. USA has the highest number of Serious Critical Cases.

2. India has the second highest number of Serious Critical Cases.

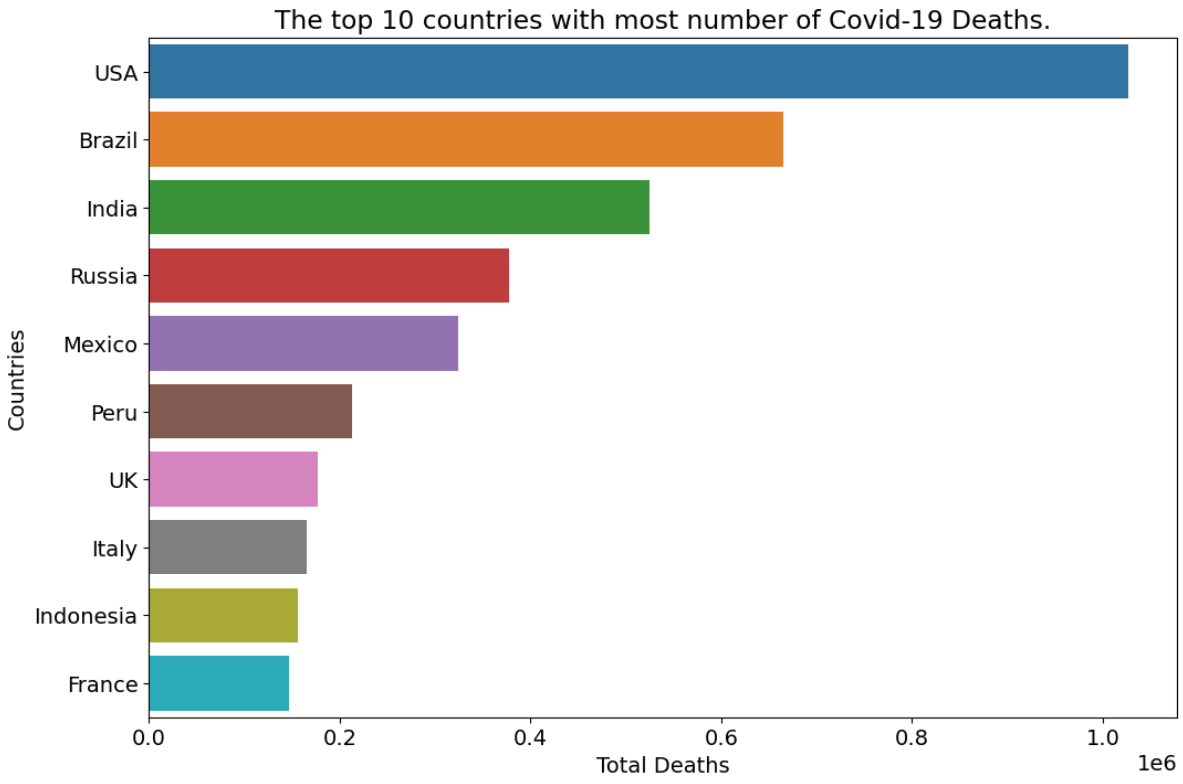
3. Brazil has the third highest number of Serious Critical Cases.

3.The top 10 countries with most number of Covid-19 Deaths.

In [51]:
```
location_group =  final_summary_data_df.groupby('country')
ans1 = location_group[['total_deaths']].sum()
ans = ans1.reset_index()
final_ans = ans.sort_values('total_deaths',ascending = False).head(10)
print(final_ans)
```

```
       country  total_deaths
212        USA     1026646.0
26      Brazil      664920.0
94       India      524214.0
165     Russia      377670.0
131     Mexico      324465.0
158       Peru      213023.0
211         UK      176708.0
101      Italy      165244.0
95   Indonesia      156458.0
72      France      147257.0
```

In [52]:
```
plt.figure(figsize=(12,8))
matplotlib.rcParams['font.size']=14
sns.barplot(y='country',x='total_deaths',data = final_ans);
plt.title('The top 10 countries with most number of Covid-19 Deaths.')
plt.xlabel('Total Deaths')
plt.ylabel('Countries');
```



The above graph shows that:-

1. USA has the highest number of Total Death Cases.

2. Brazil has the second highest number of Total Death Cases.

3. Mexico has the third highest number of Total Death Cases.

4. Total Recovered Cases Vs Active Cases Vs Total Death Cases

In [61]:
```
plt.figure(figsize=(110,88))
data_df = final_summary_data_df.reset_index().dropna(subset=['active_cases', 'total
data_df['active_percent'] = data_df['active_cases']/data_df['population'] * 100
data_df['recovered_percent'] = data_df['total_recovered']/data_df['population'] * 1
data_df['deaths_percent'] = data_df['total_deaths']/data_df['population'] * 100
data_df['confirmed_percent'] = data_df['total_confirmed']/data_df['population'] * 1
data_df = data_df.sort_values('confirmed_percent', ascending=False).drop_duplicates
fig = go.Figure(data=[go.Bar(
                    name="Deaths",
                    x=data_df['country'],
                    y=data_df['deaths_percent'],
                    marker_color='red',
                ),
                go.Bar(
                    name="Active",
                    x=data_df['country'],
                    y=data_df['active_percent'],
                    marker_color='blue',
                ),
                go.Bar(
                    name="Recovered",
                    x=data_df['country'],
                    y=data_df['recovered_percent'],
                    marker_color='lightgreen',
                )

            ]);


fig.update_layout(
        xaxis_title="Country",
        yaxis_title="Percentages(%)",
        plot_bgcolor='rgba(0,0,0,0)',
        barmode='stack'
    )
```

```
<Figure size 11000x8800 with 0 Axes>
```

To see the results hover over the Graph above which will give a detailed count of Recoverd, Active and Death percentages country wise.

5. Types of Vaccines in use

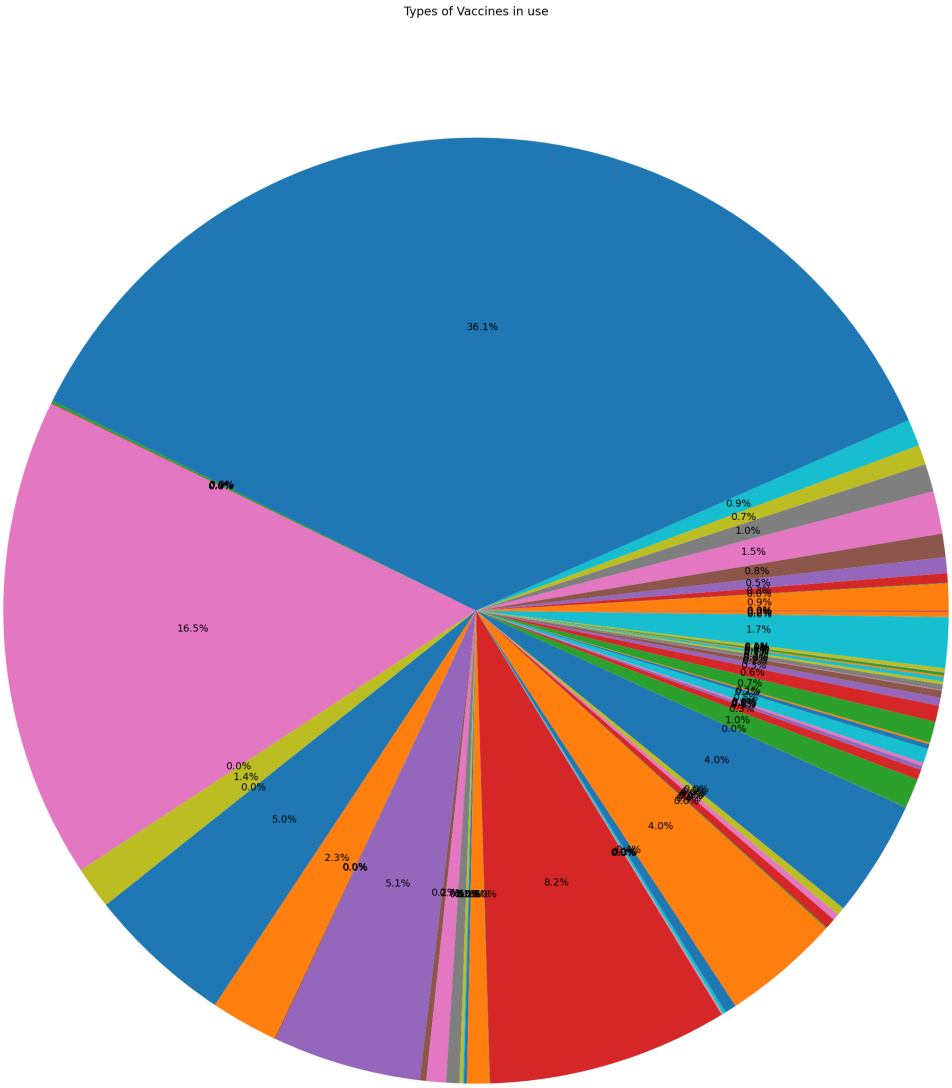
In [68]:
```
plt.figure(figsize=(60,30))
data_df = vaccine_data_df[['vaccines','total_vaccinations','country']]

data_df = vaccine_data_df.groupby('vaccines')['total_vaccinations'].sum()

data = pd.DataFrame(data_df).reset_index()
mylabels = data['vaccines']

plt.pie(data_df,autopct='%1.1f%%' );
#plt.legend(mylabels)
plt.title('Types of Vaccines in use')
```

Out[68]:
```
Text(0.5, 1.0, 'Types of Vaccines in use')
```



From The above pie Chart we can analyse that the most preferred set of vaccines is Johnson&Johnson, Moderna, Pfiser/BioNTech

Quantitative and qualitative analysis: Asking and Answering Questions.

1. Which Set of Vaccines are preferred by which Country?

In [69]:
```
plt.figure(figsize=(30,30))
data_df = vaccine_data_df[['vaccines','total_vaccinations','country']]
print("To see which country is prefering which vaccine hover over the World Map.")
print("To see which vaccine is preferred by  which country click on the Vaccine in
fig = px.choropleth(data_df, locations="country",
                    locationmode = 'country names',
                    color="vaccines",
                    hover_name="country",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.update_layout(
```

```
legend_orientation = 'h')
```

```
To see which country is prefering which vaccine hover over the World Map.
To see which vaccine is preferred by  which country click on the Vaccine in the Ta
ble Below.
```

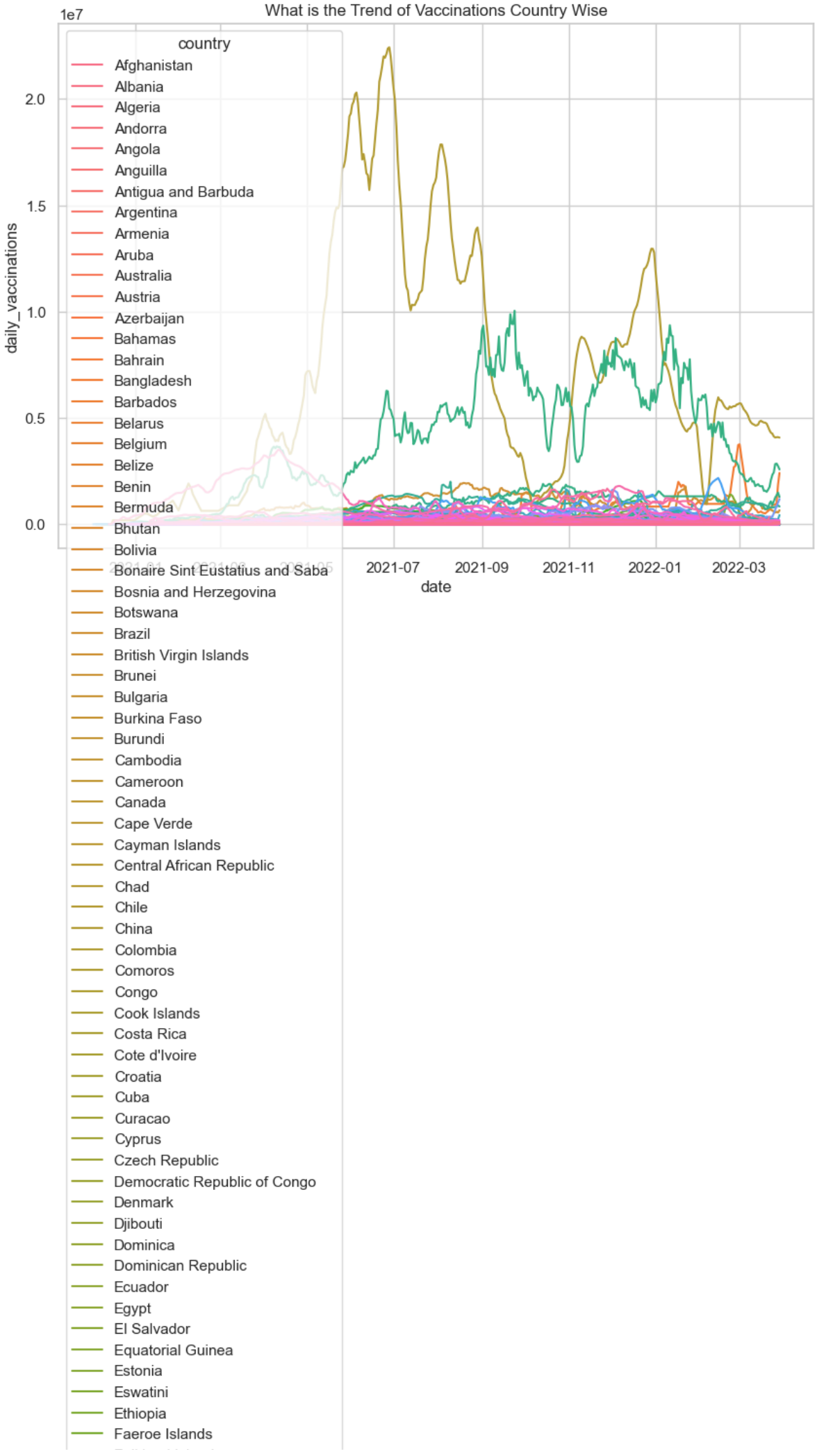
```
<Figure size 3000x3000 with 0 Axes>
```

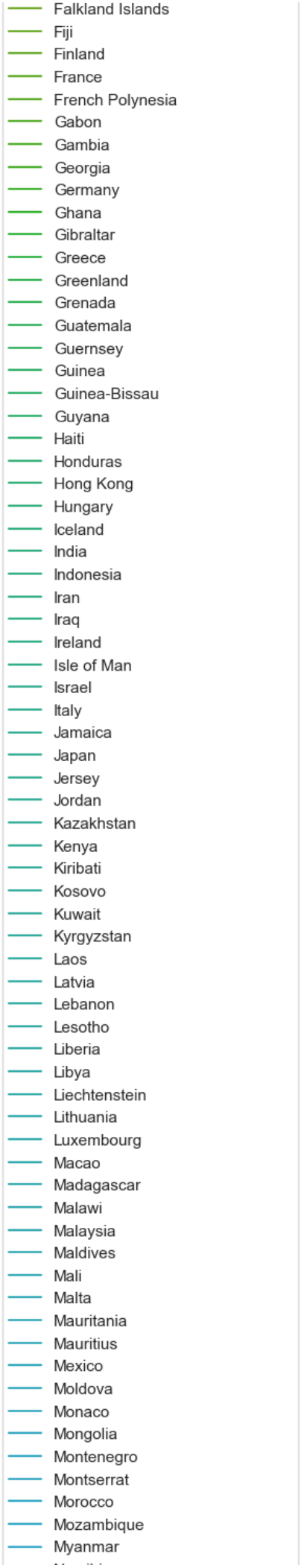
The most preferred set of vaccines that is Johnson&Johnson, Moderna, Pfiser/BioNTech is high used in North America and Europe. In India Covaxin and Oxford/AstraZeneca.

2. What is the Trend of Vaccinations Country Wise?

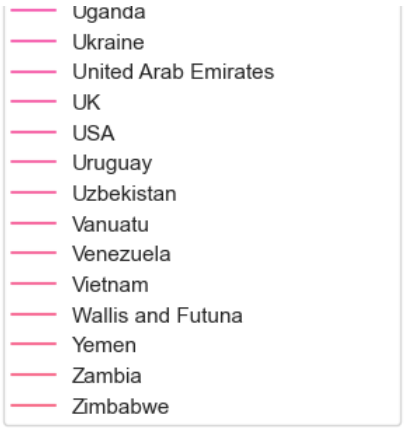
In [78]:
```
plt.figure(figsize=(10,7))
sns.set(style="whitegrid")
new_vaccine_df = vaccine_data_df.copy()
new_vaccine_df['date'] = pd.to_datetime(new_vaccine_df['date'])
new_covid_df = covid_data_df.copy()
new_covid_df['date'] = pd.to_datetime(new_covid_df['date'])
fig = sns.lineplot( x = 'date', y ='daily_vaccinations',data = new_vaccine_df,hue=
plt.title('What is the Trend of Vaccinations Country Wise')
```

Out[78]:
```
Text(0.5, 1.0, 'What is the Trend of Vaccinations Country Wise')
```


What is the Trend of Vaccinations Country Wise
1e7
daily_vaccinations
2.0
1.5
1.0
0.5
0.0
2021-07
2021-09
2021-11
2022-01
2022-03
date
country
Afghanistan
Albania
Algeria
Andorra
Angola
Anguilla
Antigua and Barbuda
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bermuda
Bhutan
Bolivia
Bonaire Sint Eustatius and Saba
Bosnia and Herzegovina
Botswana
Brazil
British Virgin Islands
Brunei
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada
Cape Verde
Cayman Islands
Central African Republic
Chad
Chile
China
Colombia
Comoros
Congo
Cook Islands
Costa Rica
Cote d'Ivoire
Croatia
Cuba
Curacao
Cyprus
Czech Republic
Democratic Republic of Congo
Denmark
Djibouti
Dominica
Dominican Republic
Ecuador
Egypt
El Salvador
Equatorial Guinea
Estonia
Eswatini
Ethiopia
Faeroe Islands


Falkland Islands
Fiji
Finland
France
French Polynesia
Gabon
Gambia
Georgia
Germany
Ghana
Gibraltar
Greece
Greenland
Grenada
Guatemala
Guernsey
Guinea
Guinea-Bissau
Guyana
Haiti
Honduras
Hong Kong
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Isle of Man
Israel
Italy
Jamaica
Japan
Jersey
Jordan
Kazakhstan
Kenya
Kiribati
Kosovo
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Macao
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Mauritania
Mauritius
Mexico
Moldova
Monaco
Mongolia
Montenegro
Montserrat
Morocco
Mozambique
Myanmar


Namibia
Nauru
Nepal
Netherlands
New Caledonia
New Zealand
Nicaragua
Niger
Nigeria
Niue
North Macedonia
Norway
Oman
Pakistan
Palestine
Panama
Papua New Guinea
Paraguay
Peru
Philippines
Pitcairn
Poland
Portugal
Qatar
Romania
Russia
Rwanda
Saint Helena
Saint Kitts and Nevis
Saint Lucia
Saint Vincent and the Grenadines
Samoa
San Marino
Sao Tome and Principe
Saudi Arabia
Senegal
Serbia
Seychelles
Sierra Leone
Singapore
Sint Maarten (Dutch part)
Slovakia
Slovenia
Solomon Islands
Somalia
South Africa
South Korea
South Sudan
Spain
Sri Lanka
Sudan
Suriname
Sweden
Switzerland
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Timor
Togo
Tokelau
Tonga
Trinidad and Tobago
Tunisia
Turkey
Turkmenistan
Turks and Caicos Islands
Tuvalu



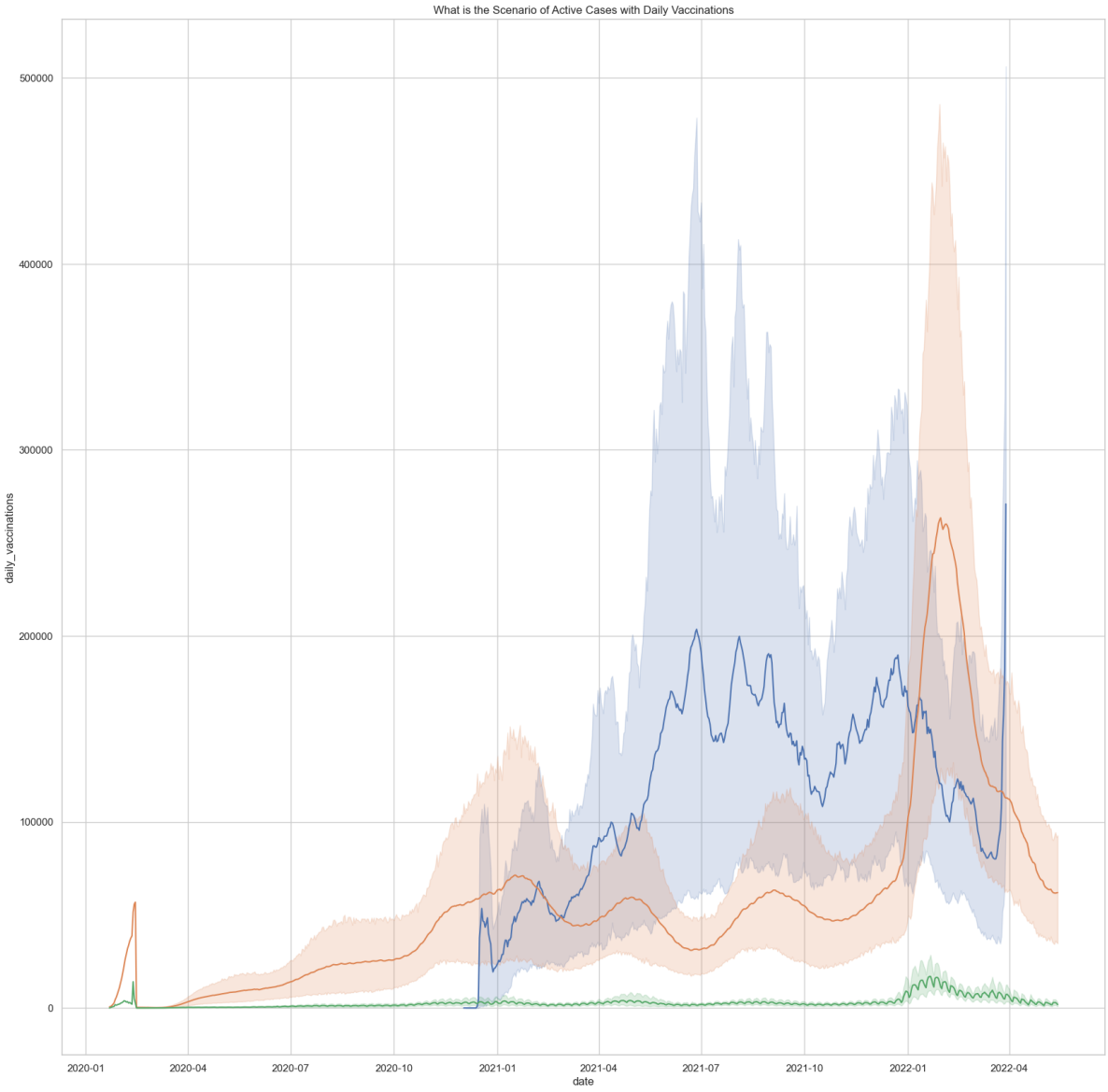
From the above Graph we can analyse that:-

Daily Vaccinations have high Rate in USA, UK, China and India

3. What is the Scenario of Active Cases with Daily Vaccinations ?

In [79]:
```
 plt.figure(figsize=(20,20))
sns.set(style="whitegrid")
new_vaccine_df = vaccine_data_df.copy()
new_vaccine_df['date'] = pd.to_datetime(new_vaccine_df['date'])
new_covid_df = covid_data_df.copy()
new_covid_df['date'] = pd.to_datetime(new_covid_df['date'])
fig = sns.lineplot( x = 'date', y ='daily_vaccinations',data = new_vaccine_df,lege
fig = sns.lineplot(x='date',y= 'active_cases', data= new_covid_df,legend = 'brief'
fig = sns.lineplot(x='date',y= 'daily_new_cases', data= new_covid_df,legend = 'bri
plt.title("What is the Scenario of Active Cases with Daily Vaccinations")
```

Out[79]:
```
Text(0.5, 1.0, 'What is the Scenario of Active Cases with Daily Vaccinations')
```



Red Region : Active Cases

Blue Region : Daily Vaccinations

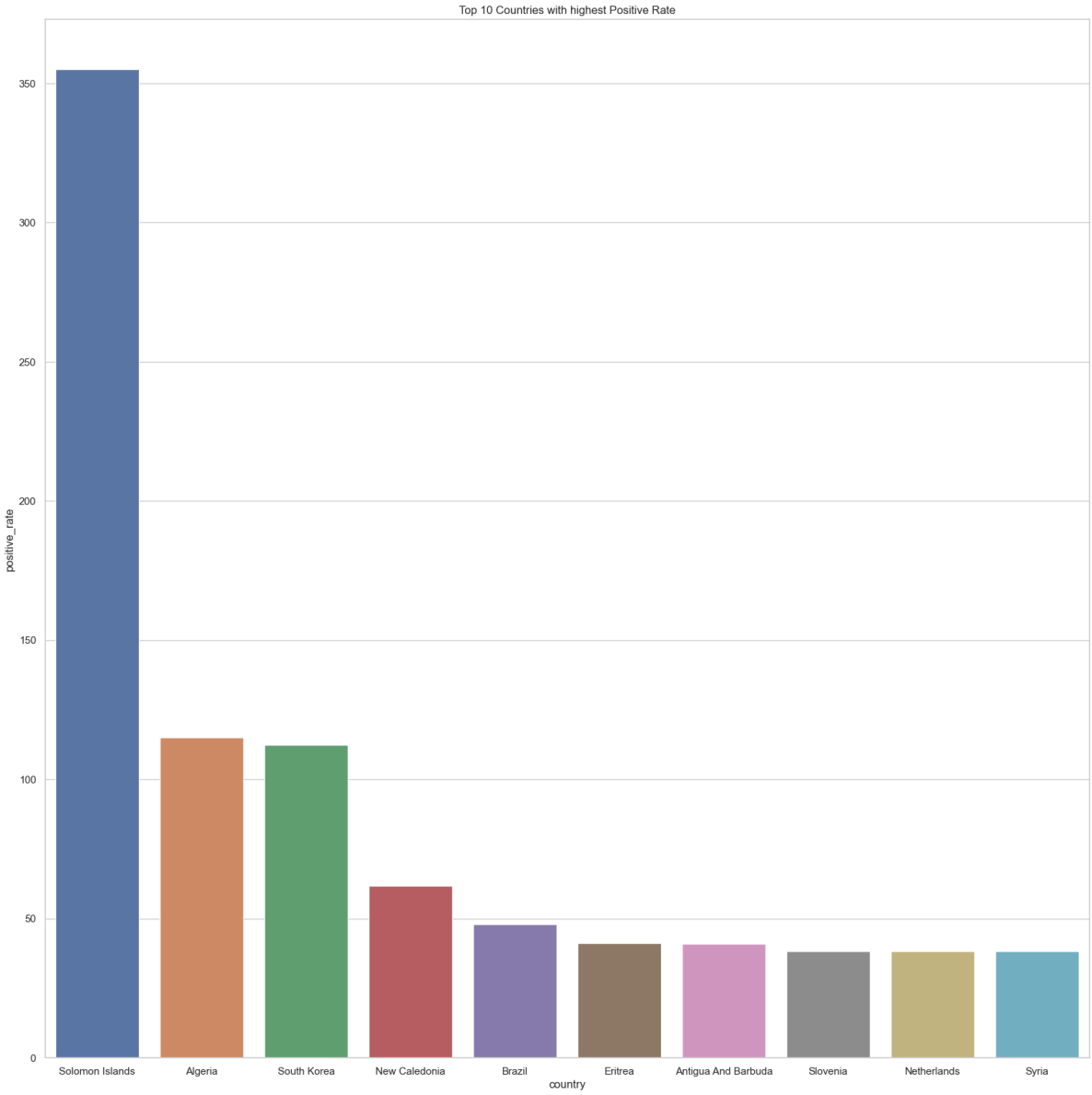
Green Region : Daily New Cases

With increasing rate of Vaccination the Rate of Active Cases seems to getting affected moderateley in the mid of March.

4. What is Country Wise Positive Rate?

In [65]:
```
plt.figure(figsize=(20,20))
top_cases_df= final_summary_data_df.sort_values('positive_rate',ascending = False)

sns.barplot(x = 'country',
            y = 'positive_rate',
            data = top_cases_df);
plt.title('Top 10 Countries with highest Positive Rate');
```



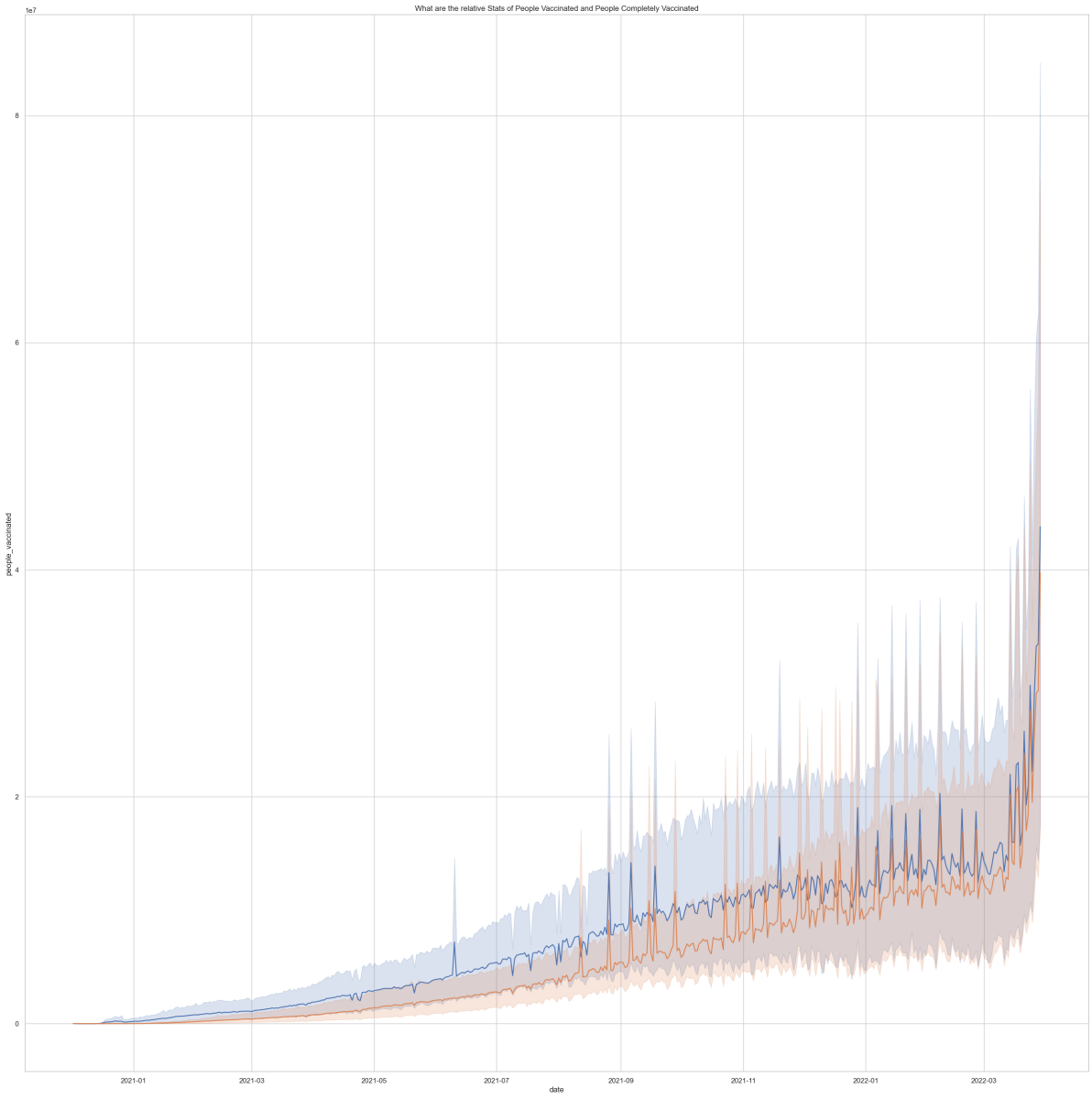
From the above graph it looks like French Polynesia had the worst hit of Positive Rate followed with Seychelles and Brazil.

5. What are the relative Stats of People Vaccinated and People Completely Vaccinated?

In [81]:
```
fig = plt.figure(figsize=(30,30))
print("Blue Region:people_vaccinated")
print("Red Region:people_fully_vaccinated")
sns.set(style="whitegrid")
new_vaccine_df = vaccine_data_df.copy()
new_vaccine_df['date'] = pd.to_datetime(new_vaccine_df['date'])
new_covid_df = covid_data_df.copy()
new_covid_df['date'] = pd.to_datetime(new_covid_df['date'])
fig = sns.lineplot( x = 'date', y ='people_vaccinated',data = new_vaccine_df,legen
fig = sns.lineplot(x='date',y= 'people_fully_vaccinated', data= new_vaccine_df,leg
plt.title("What are the relative Stats of People Vaccinated and People Completely V
```

```
Blue Region:people_vaccinated
Red Region:people_fully_vaccinated
```

Out[81]:
```
Text(0.5, 1.0, 'What are the relative Stats of People Vaccinated and People Comple
tely Vaccinated')
```



6.Which country has vaccinated most of its population?

In [83]:
```
location_group =  vaccine_data_df.groupby('country')
ans1 = location_group[['people_fully_vaccinated']].sum()
ans1['people_fully_vaccinated'] =ans1['people_fully_vaccinated'].astype(int)
ans = ans1.reset_index()
final_ans1 = ans.sort_values('people_fully_vaccinated',ascending = False)
final_ans1['population'] = final_summary_data_df['population']
#final_ans['percentage_vaccinated'] =(final_ans.total_vaccinations/ final_ans.popul
final_ans1['total_deaths'] = final_summary_data_df['total_deaths']
final_ans1['active_cases'] = final_summary_data_df['active_cases']
final_ans1['total_recovered'] = final_summary_data_df['total_recovered']
final_ans1
```

Which country has vaccinated most of its population

Out[83]:

	country	people_fully_vaccinated	population	total_deaths	active_cases	total_recovered
132	Morocco	2133597349	117269	0.0	6.0	1.0
180	South Africa	2073198573	99481	167.0	515.0	42553.0
159	Romania	2064644712	112297269	60455.0	3008.0	3624459.0
207	Ukraine	1951609752	1407928	3871.0	9418.0	141354.0
190	Taiwan	1948522890	11438136	138.0	3873.0	13514.0
...	...	...	...	...	...	...
35	Canada	-2147483648	38358465	40228.0	281554.0	3499564.0
18	Belgium	-2147483648	11683561	31613.0	143434.0	3941350.0
193	Thailand	-2147483648	5322203	5353.0	82.0	576684.0
10	Australia	-2147483648	26050899	7794.0	386179.0	6199822.0
42	Colombia	-2147483648	7610260	9361.0	0.0	0.0

218 rows × 6 columns

USA , Israel and India are leading in Population with Fully Vaccinated People.

In [84]:

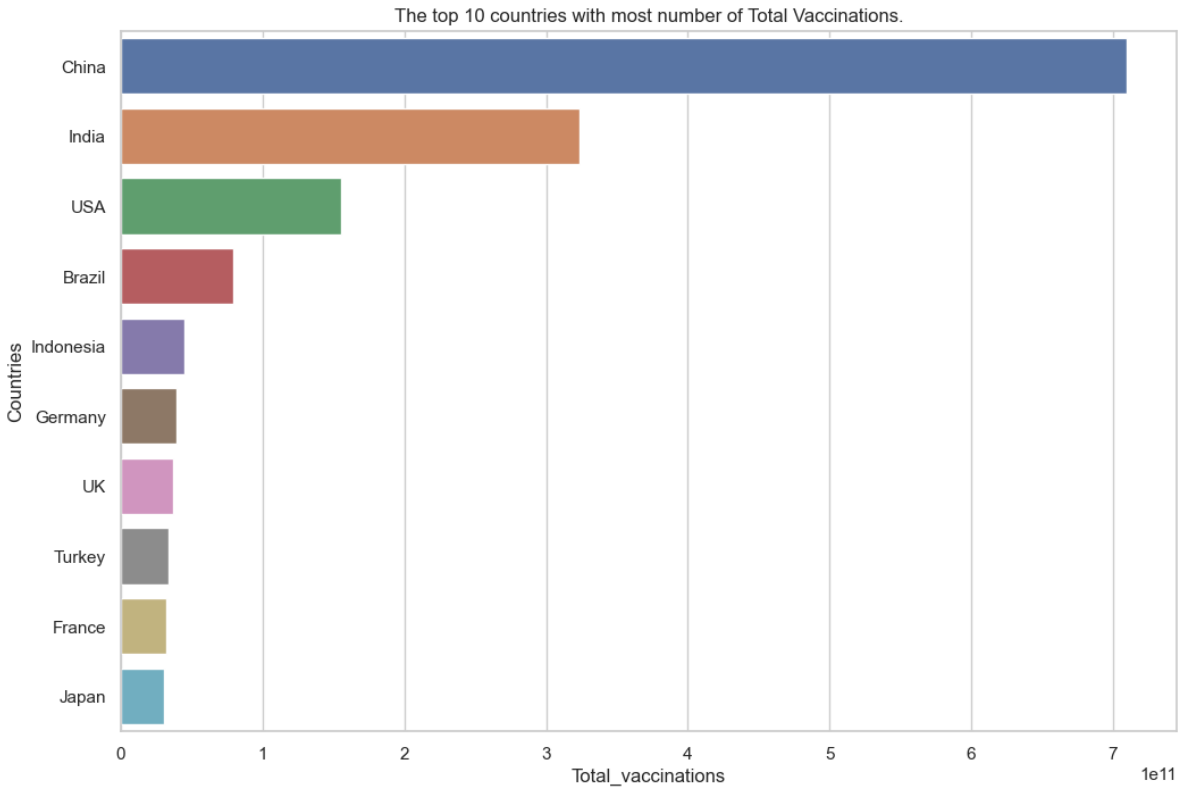
```
print("To see Country wise data hover over the World Map.")
fig = px.choropleth(final_ans1, locations="country",
                    locationmode='country names',
                    color="people_fully_vaccinated",
                    hover_name="country",
                    hover_data=['total_deaths', 'active_cases', 'total_recovered'],
                    color_continuous_scale="Sunset"
                   )

fig.show()
```

```
To see Country wise data hover over the World Map.
```

In [85]:

```
location_group =  vaccine_data_df.groupby('country')
ans1 = location_group[['total_vaccinations']].sum()
ans = ans1.reset_index()
final_ans = ans.sort_values('total_vaccinations',ascending = False).head(10)
plt.figure(figsize=(12,8))
matplotlib.rcParams['font.size']=14
sns.barplot(y='country',x='total_vaccinations',data = final_ans);
plt.title('The top 10 countries with most number of Total Vaccinations.')
plt.xlabel('Total_vaccinations')
plt.ylabel('Countries');
```



USA, UK, and India are leading in Total Vaccinations

Inferences and Conclusions.

A. Total Confirmed Cases.

1. USA has the highest number of Total Confirmed Cases. 2. Brazil has the second highest number of Total Confirmed Cases. 3. India has the third highest number of Total Confirmed Cases.

B. Serious Critical Cases.

1. USA has the highest number of Serious Critical Cases. 2. India has the second highest number of Serious Critical Cases. 3. Brazil has the third highest number of Serious Critical Cases.

C.Total Death Cases.

1. USA has the highest number of Total Death Cases. 2. Brazil has the second highest number of Total Death Cases. 3. Mexico has the third highest number of Total Death Cases.

D. Types of Vaccines in use.

1.The most preferred set of vaccines is Johnson&Johnson, Moderna, Pfiser/BioNTechin North America and Europe. In India Covaxin and Oxford/AstraZeneca.

E. Positive Rate.

French Polynesia had the worst hit of Positive Rate followed with Seychelles and Brazil.

F. Effect of Vaccination.

With increasing rate of Vaccination the Rate of Active Cases seems to getting affected moderateley in the mid of March.

G.Vaccination Countrywise.

USA, UK, and India are leading in Total Vaccinations

Future Work.

1. With the help of the Dataset we can analyse the factors like Politics,economy & demography that are influencing the Vaccination.

2. There can also be the analysis of the GDP change due to Vaccination of Various Countries using interesting Visualization

3. In the Future the dataset can be used to analyse public sentiments about Vaccine.

4. With the help of deep learning Algorithms one can predict the Future Turn of events

5. One can also analyse whether the lockdowns are effective in controlling the cases.