COVID Vaccines Analysis

Aim –

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.

Introduction –

The COVID 19 pandemic caused due to the Corona virus devastated the world by causing several fatalities around the world. This virus originated in Wuhan, China in 2019 and was later spread throughout the world due to human contact in one way or the other. The disease showed symptoms as basic as mild fever and cold but also caused life threatening symptoms like breathing problems caused by damage to the lungs. As this virus was new to the world and there was no vaccine or cure to it at the initial period there were several deaths around the world. The countries around the world were forced to shut themselves to others in order to avoid the further spread of the virus and people were stuck inside their houses and faced many issues with their finances, mental health etc., and felt like animals in a cage. An effort was made to find a cure or vaccine by several health organizations to bring a stop to this pandemic.

In later stages of 2020 several experimental vaccines were developed and was administered to humans. The efforts were successful as the vaccines were helpful in reducing the affects the virus and even if people were infected, they were not in any life threating situation and escaped the illness having only minor symptoms.

Many countries later developed their own vaccines and also helped other countries without the resources by providing them with vaccines developed.

Problem Statements –

1. In this project we have analyzed the top 10 fully vaccinated countries
2. We have analyzed the top 5 vaccinated countries
3. We have analyzed the top 5 daily vaccinating countries
4. We have analyzed the total number of daily vaccinations, people who have fully vaccinated, people who is vaccinated
5. We have analyzed year wise daily vaccinating details, fully vaccinated people details, vaccinated people details
6. We have analyzed the country wise vaccines and iso_code details.

Methodology –

Step 1 –

Data Importing –

In power BI desktop with the help of the get data option import the CSV data which is named as country_vaccinations and clicked load option.

Step 2 –

Data Cleaning –

After loading the data and after analyzing the data | understood that there are 86512 rows and 15 columns. And in that some of the columns contained null values I have replaced the null values by 0 with the use of replace functions and started working on the data.

Step 3 –

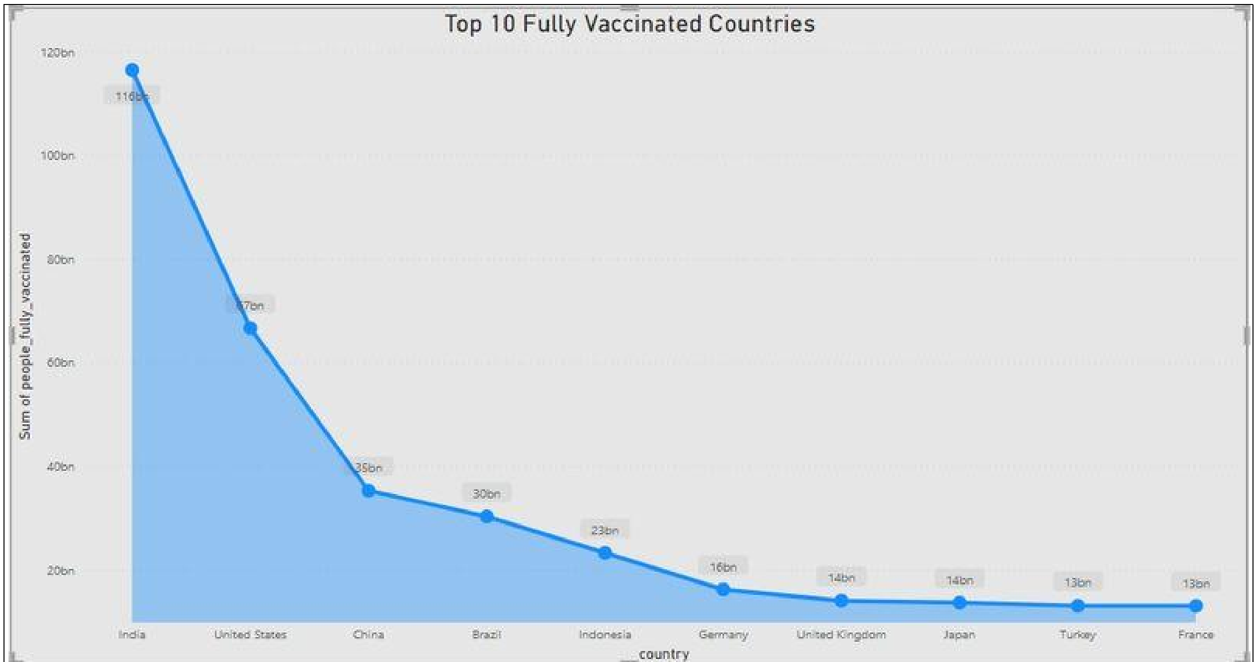
Visualizations –

In visualization part with the help of power BI desktop software I have used different kinds of charts, graphs, cards and table to display the data in the format which will be easy to understand.

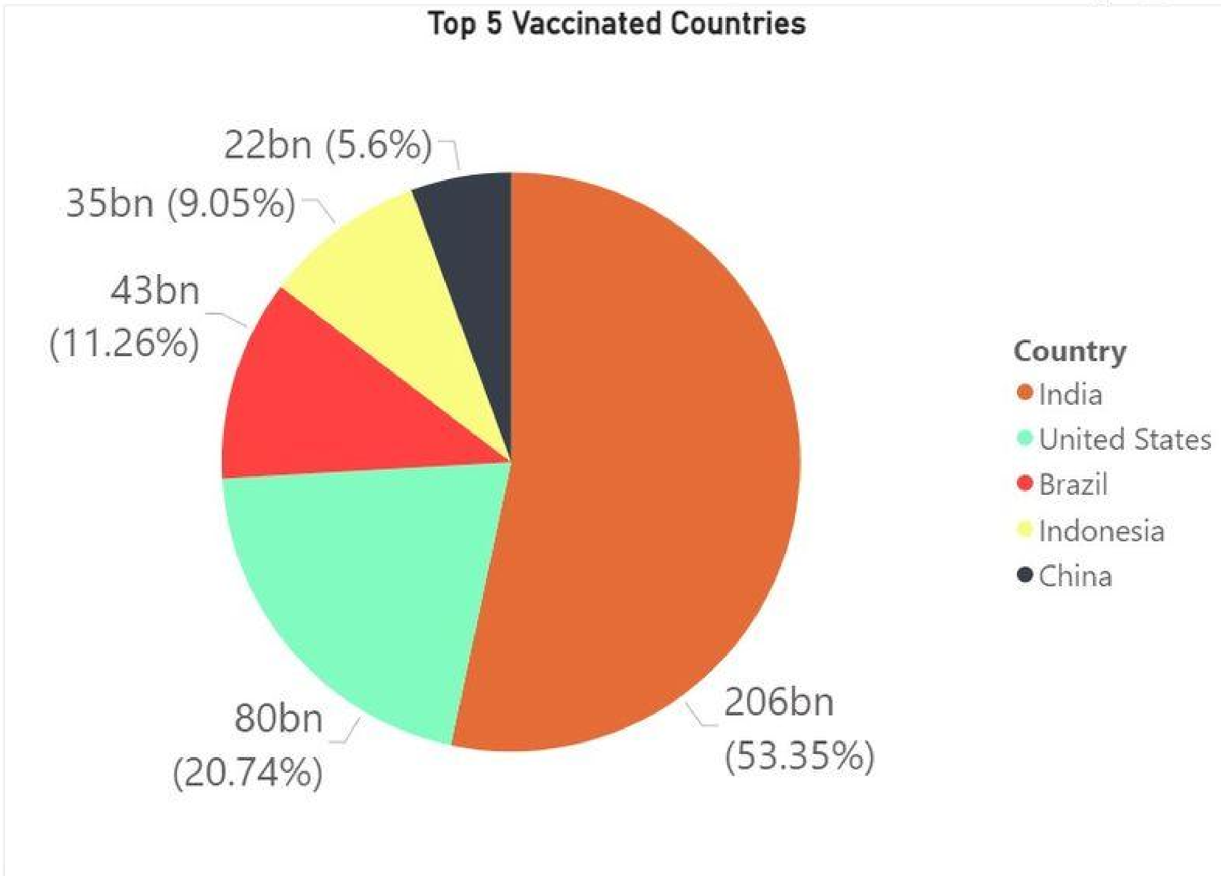
Analysis –

In the analysis part first | have analyzed the top 10 fully vaccinated countries by using area chart and have used the filter option to find the top countries and the result obtained as below,

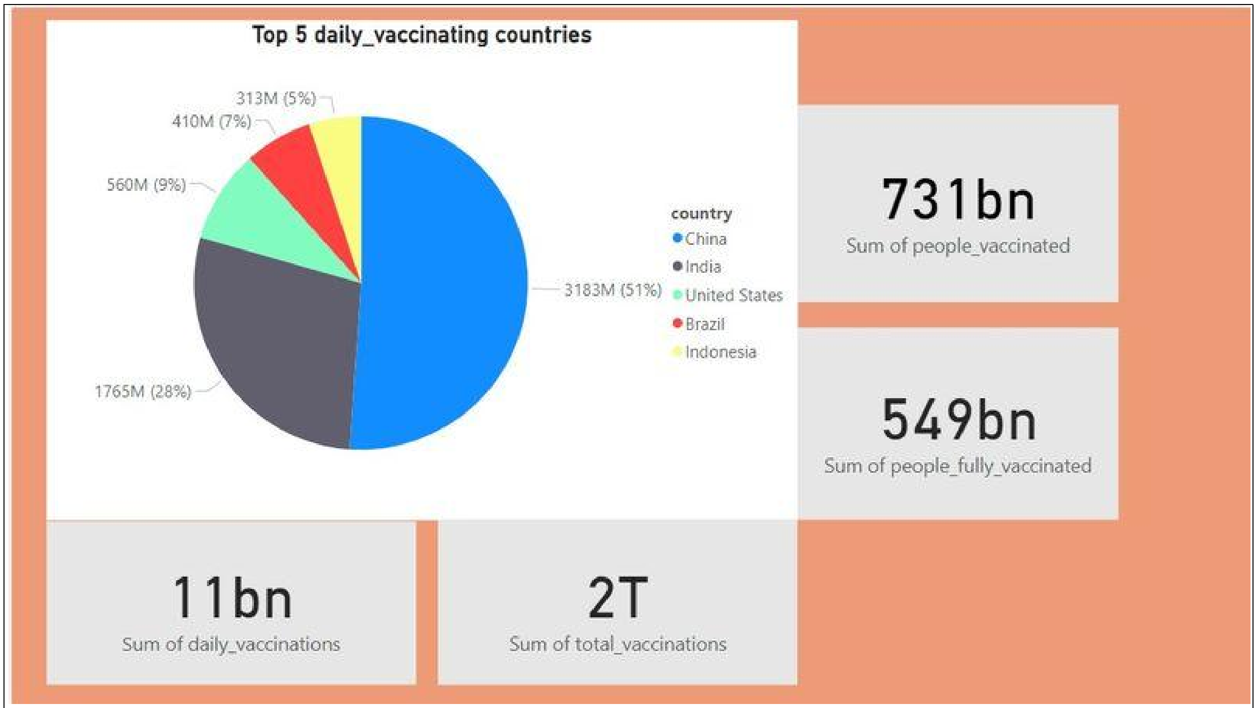
From the below image we can able to come to know that India is the top country in terms full vaccination with 116 billon , followed by united states of America and china with 67 billion and 35 billion respectively.



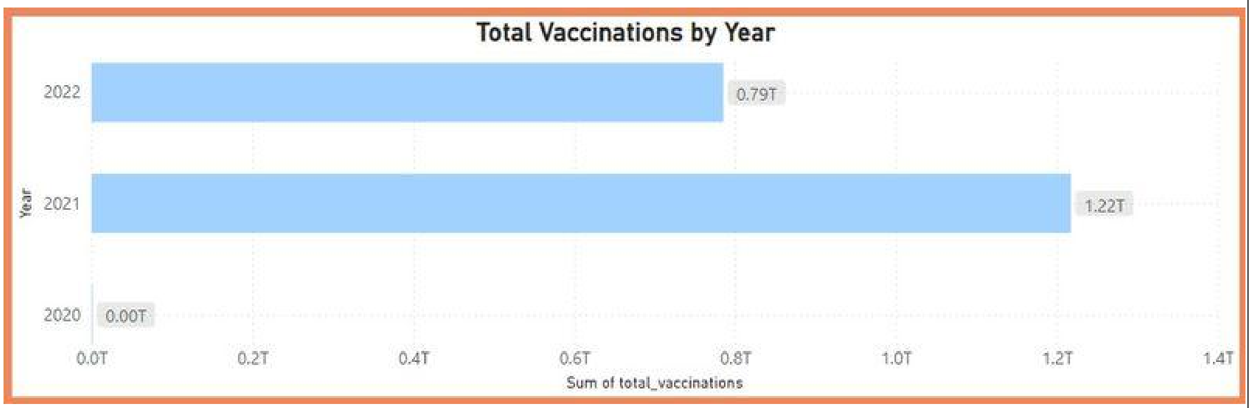
In the second analysis we have analyzed the top 5 vaccinated countries with the help of pie chart and used filter option to find the top countries and with that we came to know that India is the top country with more number of vaccinated peoples followed by United States of America and Brazil.



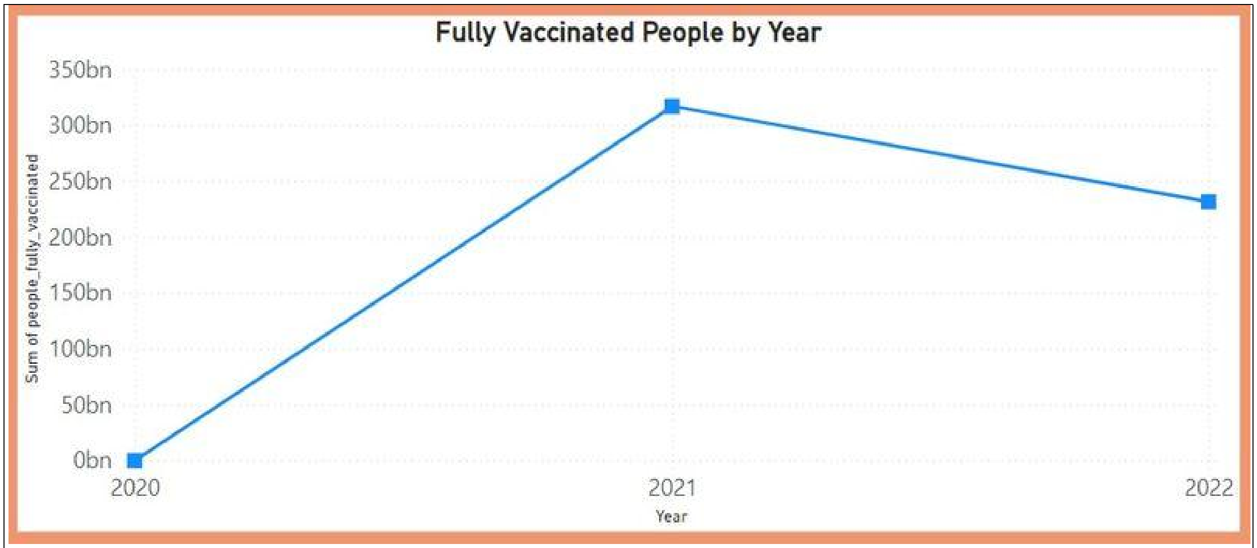
In the third analysis we have analyzed the top 5 country with daily vaccinations with the help of pie chart and used filter option to find the top country with daily vaccinations and with that we came to know that China is the top country with more number of vaccinations followed by India and United States of America.



From the above images we have came to know about the statics of daily vaccination, people who have fully vaccinated and people who is vaccinated and I used cards for this to display the value.



Above image have shown that daily vaccinating details year wise and here we can conclude that 2021 is a year which having maximum daily vaccinating details and I have used bar chart.

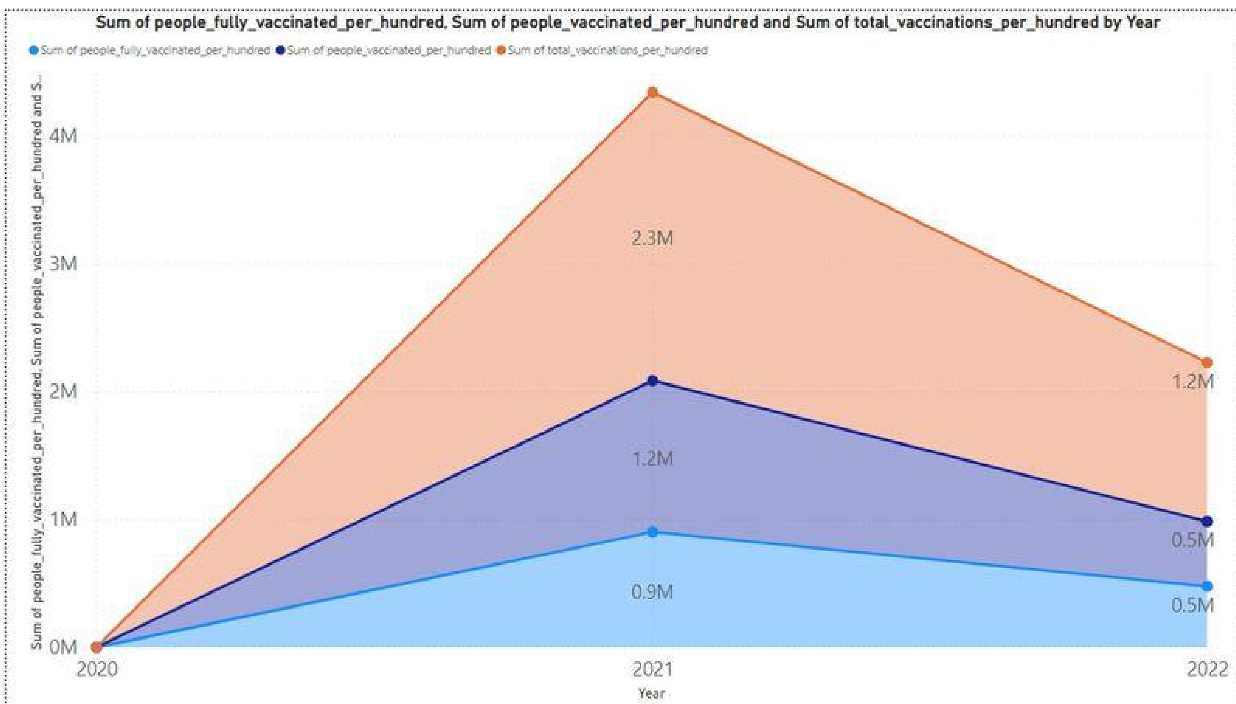


Above image have shown that fully vaccinating details year wise and here we can conclude that 2021 is a year which is having maximum number fully vaccinated peoples and I have used Line chart.

country	vaccines	iso_code
Afghanistan	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing	AFG
Albania	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	ALB
Algeria	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V	DZA
Andorra	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	AND
Angola	Oxford/AstraZeneca	AGO
Anguilla	Oxford/AstraZeneca, Pfizer/BioNTech	AIA
Antigua and Barbuda	Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V	ATG
Argentina	CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V	ARG
Armenia	Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V	ARM
Aruba	Pfizer/BioNTech	ABW
Australia	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	AUS
Austria	Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech	AUT
Azerbaijan	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	AZE
Bahamas	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	BHS
Bahrain	Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik V	BHR
Bangladesh	Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac	BGD
Barbados	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing	BRB
Belarus	Sinopharm/Beijing, Sputnik V	BLR
Belgium	Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	BEL
Belize	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing	BLZ
Benin	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	BEN
Bermuda	Oxford/AstraZeneca, Pfizer/BioNTech	BMU
Bhutan	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing	BTN

From the above chart I have used table to understand the vaccine details and iso_code details country wise.

From the above images we have analyzed the daily vaccinations per million, people who have fully vaccinated per hundred, people who have vaccinated per hundred, total vaccinations per hundred.



From the above image we came to know about people who have vaccinated per hundred, people who have fully vaccinated per hundred, total vaccinated per hundred and we came. to know about that 2021 was the peak year in all the 3 cases.

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

Title: Analyzing the Effectiveness and Impact of COVID-19 Vaccination Programs

Abstract: The global response to the COVID-19 pandemic has been characterized by the rapid development and distribution of vaccines. These vaccines have played a crucial role in controlling the spread of the virus, reducing severe cases, and ultimately saving lives. This comprehensive analysis focuses on COVID-19 vaccination programs, offering insights into their development, distribution, effectiveness, and broader societal implications.

- Vaccine Development to COVID-19 Vaccines, Vaccine Platforms (mRNA, Viral Vector, Protein Subunit) ,Mechanisms of Action ,Efficacy and Variants ,Safety Profiles and Adverse Events

- Vaccine Distribution - Global Distribution Challenges , Supply Chain Logistics ,Vaccine Hesitancy and Public Perception, Equity in Access, International Collaboration
- Effectiveness and Impact , Reduction in COVID-19 Cases , Impact on Hospitalizations and Mortality ,Long-term Public Health Implications , Emerging Variants and Vaccine Adaptation
- Societal and Economic Implications -Economic Recovery , Social Reintegration , Psychological and Mental Health Effects , Vaccine Passports and Privacy Concerns
- Lessons Learned and Future Outlook - Key Takeaways from COVID-19 Vaccination Programs , Preparedness for Future Pandemics, Ethical Considerations , Conclusion and Policy Recommendations

This analysis provides a comprehensive overview of the development, distribution, effectiveness, and societal impacts of COVID-19 vaccines, offering valuable insights for policymakers, healthcare professionals, and the general public as we continue to navigate the evolving landscape of the pandemic.

```python
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import os

for dirname, _, filenames in os.walk('/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A Num
Py version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected v
ersion 1.23.5
```

```
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
/input/covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv
/input/covid-world-vaccination-progress/country_vaccinations.csv
```

In [2]:

```python
data = pd.read_csv("/input/covid-world-vaccination-progress/country_vaccinations.csv")
data.head()
```

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million	vaccines	source_name	source_website
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	0.0	0.0	NaN	NaN	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfize	World Health Organizati	https://covid19.who.int/

													r/Bi...	on	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

In [3]:

```
data.describe()
```

Out[3]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	41294.000000	38802.000000	86213.000000
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313055e+05	80.188543	40.927317	35.523243	3257.049157
std	2.246004e+08	7.078731e+07	5.713920e+07	1.212427e+06	7.682388e+05	67.913577	29.290759	28.376252	3934.312440
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
25%	5.264100e+05	3.494642e+05	2.439622e+05	4.668000e+03	9.000000e+02	16.050000	11.370000	7.020000	636.000000
50%	3.590096e+06	2.187310e+06	1.722140e+06	2.530900e+04	7.343000e+03	67.520000	41.435000	31.750000	2050.000000
75%	1.701230e+07	9.152520e+06	7.559870e+06	1.234925e+05	4.409800e+04	132.735000	67.910000	62.080000	4682.000000

max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	124.760000	122.370000	117497.000000

In [4]:

```
pd.to_datetime(data.date)

data.country.value_counts()
```

Out[4]:

```
Norway                          482
Latvia                          480
Denmark                         476
United States                   471
Russia                          470
                               ...
Bonaire Sint Eustatius and Saba 146
Tokelau                         114
Saint Helena                     92
Pitcairn                         85
Falkland Islands                 67
Name: country, Length: 223, dtype: int64
```

In [5]:

```
data = data[data.country.apply(lambda x: x not in ["England", "Scotland", "Wales", "Northern Ireland"])]

data.country.value_counts()
```

Out[5]:

```
Norway                           482
Latvia                           480
Denmark                          476
United States                    471
Canada                           470
                                ...
Bonaire Sint Eustatius and Saba  146
Tokelau                          114
Saint Helena                      92
Pitcairn                          85
Falkland Islands                  67
Name: country, Length: 219, dtype: int64
```

In [6]:

```
data.vaccines.value_counts()
```

Out[6]:

```
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
 7608
Oxford/AstraZeneca
 6022
Oxford/AstraZeneca, Pfizer/BioNTech
 4629
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
 4491
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
Name: vaccines, Length: 84, dtype: int64
```

In [7]:

```python
df = data[["vaccines", "country"]]
df.head()
```

Out[7]:

	vaccines	country
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	Afghanistan
1	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	Afghanistan
2	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	Afghanistan
3	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	Afghanistan

4	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	Afghanistan

In [8]:

```
dict_ = {}
for i in df.vaccines.unique():
  dict_[i] = [df["country"][j] for j in df[df["vaccines"]==i].index]
vaccines = {}
for key, value in dict_.items():
  vaccines[key] = set(value)
for i, j in vaccines.items():
  print(f"{i}:>>{j}")
```

```
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>>{'Camer
oon', 'Afghanistan', 'Belize', 'Namibia', 'Trinidad and Tobago'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>>{'Oman', 'Bosnia and He
rzegovina', 'Albania', 'Azerbaijan'}

Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V:>>{'Algeria', 'Zimbabwe
'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Guernsey', 'United Kingdom', 'Fi
ji', 'Sweden', 'Australia', 'Jersey', 'Sint Maarten (Dutch part)', 'Finland', 'An
dorra', 'Japan', 'Isle of Man'}

Oxford/AstraZeneca:>>{'Montserrat', 'Kiribati', 'Saint Helena', 'Saint Vincent an
d the Grenadines', 'Liberia', 'Falkland Islands', 'Solomon Islands', 'Tuvalu', 'V
anuatu', 'Democratic Republic of Congo', 'Pitcairn', 'Mali', 'Papua New Guinea',
'Nigeria', 'Samoa', 'Nauru', 'Togo', 'Angola', 'Tonga', 'Sao Tome and Principe'}

Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Anguilla', 'Saudi Arabia', 'Gibraltar', '
Saint Lucia', 'Cayman Islands', 'New Zealand', 'Saint Kitts and Nevis', 'Panama',
 'Bermuda', 'Costa Rica', 'Kosovo'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>>{'Antigua and Barbuda'}
```

```
CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik
 V:>>{'Argentina'}

Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V:>>{'Armenia'}

Pfizer/BioNTech:>>{'Monaco', 'Tokelau', 'Cook Islands', 'New Caledonia', 'Turks a
nd Caicos Islands', 'Niue', 'Aruba'}

Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Czechi
a', 'Slovenia', 'Netherlands', 'Germany', 'Austria', 'South Korea', 'Lithuania',
'Italy'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Bahamas', 'Eswatini', 'G
renada'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sputnik Light, Sputnik V:>>{'Bahrain'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sinovac:>>{'Bangladesh'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>>{'Maldives', 'Peru', 'Su
riname', 'Barbados', 'Dominica'}

Sinopharm/Beijing, Sputnik V:>>{'Belarus', 'Kyrgyzstan'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Cyprus', 'Portu
gal', 'Iceland', 'Malta', 'Belgium', 'Croatia', 'Jamaica', 'Luxembourg', 'Poland',
 'France', 'Greece', 'Spain', 'Romania', 'Bulgaria', 'Estonia', 'Ireland', 'Canad
a'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>>{'Benin', 'Brazil
'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>>{'Cape Verde',
'Bhutan'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik
V:>>{'Moldova', "Cote d'Ivoire", 'Morocco', 'Bolivia'}

Moderna, Pfizer/BioNTech:>>{'Faeroe Islands', 'Norway', 'Bonaire Sint Eustatius a
nd Saba', 'Curacao', 'Qatar', 'Israel'}

Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinova
c:>>{'Botswana'}
```

```
Johnson&Johnson, Oxford/AstraZeneca:>>{'British Virgin Islands', 'South Sudan', '
Malawi'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijin
g:>>{'Nepal', 'Brunei', 'Kenya', 'Kuwait'}

Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing:>>{'Gambia', 'Mozambique',
 'Madagascar', 'Senegal', 'Lesotho', 'Zambia', 'Burkina Faso'}

Sinopharm/Beijing:>>{'Equatorial Guinea', 'Burundi', 'Chad'}

Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac:>>{'Somalia', 'Ca
mbodia'}

Covaxin, Oxford/AstraZeneca:>>{'Central African Republic'}

CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>>{'Chile', 'Ecuador'}

CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001:>>{'China'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>>{'Uganda
', 'Ukraine', 'Colombia'}

Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing:>>{'Mauritius', 'Comoros'}

Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V:>>{'Congo'}

Abdala, Soberana Plus, Soberana02:>>{'Cuba'}

Johnson&Johnson, Moderna, Pfizer/BioNTech:>>{'United States', 'Liechtenstein', 'D
enmark', 'Switzerland'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac,
 Sputnik V:>>{'Egypt', 'Djibouti', 'Guinea'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>>{'Dominican Rep
ublic', 'Georgia', 'El Salvador'}

Covaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac:>>{'Ethi
opia'}

Johnson&Johnson, Pfizer/BioNTech:>>{'South Africa', 'French Polynesia'}

Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>>{'Gabon'}

Oxford/AstraZeneca, Sputnik V:>>{'Ghana'}
```

```
Moderna:>>{'Greenland', 'Wallis and Futuna'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>>{'Guatemala'}

Oxford/AstraZeneca, Sinopharm/Beijing:>>{'Niger', 'Myanmar', 'Mauritania', 'Sierr
a Leone', 'Guinea-Bissau'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>>{'Sr
i Lanka', 'Guyana'}

Johnson&Johnson, Moderna:>>{'Haiti'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>>{'Hond
uras'}

Pfizer/BioNTech, Sinovac:>>{'Hong Kong'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sputnik V:>>{'Hungary', 'Jordan'}

Covaxin, Oxford/AstraZeneca, Sputnik V:>>{'India'}

Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm
/Beijing, Sinovac:>>{'Indonesia'}

COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm
/Beijing, Soberana02, SpikoGen, Sputnik V:>>{'Iran'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>>{'Lebanon', '
Iraq', 'Montenegro', 'Mongolia', 'Serbia'}

QazVac, Sinopharm/Beijing, Sputnik V:>>{'Kazakhstan'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac,
 Sputnik Light, Sputnik V:>>{'Laos'}

Johnson&Johnson, Moderna, Novavax, Pfizer/BioNTech:>>{'Latvia'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>>{'No
rth Macedonia', 'Libya'}

Pfizer/BioNTech, Sinopharm/Beijing:>>{'Macao'}

CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>>{'Mala
ysia'}
```

```
CanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac,
Sputnik V:>>{'Mexico'}

Abdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik
 Light, Sputnik V:>>{'Nicaragua'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>>{'Northern Cyprus', 'Timor', 'Urug
uay'}

CanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sinovac, Sputnik V:>>{'Pakistan'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sinovac, Sputnik Light, Sputnik V:>>{'Palestine', 'Philippines'}

Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac,
 Sputnik V:>>{'Paraguay'}

EpiVacCorona, Sputnik V:>>{'Russia'}

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing,
 Sinovac, Sputnik V:>>{'Tunisia', 'Rwanda'}

Pfizer/BioNTech, Sputnik V:>>{'San Marino'}

Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V:>>{'Seychelles'}

Moderna, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>>{'Singapore'}

Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik
V:>>{'Slovakia'}

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinova
c:>>{'Sudan'}

Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, S
putnik V:>>{'Syria'}

Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Taiwan'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>>{'Tajikistan'}

Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing:>>{'Tanzania'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>>{'Thai
land'}
```

```
Pfizer/BioNTech, Sinovac, Turkovac:>>{'Turkey'}

EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001:>>
{'Turkmenistan'}

Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik
V:>>{'United Arab Emirates'}

Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V,
ZF2001:>>{'Uzbekistan'}

Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik V:>>{'Vene
zuela'}

Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik
V:>>{'Vietnam'}

Johnson&Johnson, Oxford/AstraZeneca, Sinovac:>>{'Yemen'}
```

In [9]:

```
import plotly.express as px

import plotly.offline as py

vaccine_map = px.choropleth(data, locations = 'iso_code', color = 'vaccines')

vaccine_map.update_layout(height=300, margin={"r":0,"t":0,"l":0,"b":0})

vaccine_map.show()
```

Machine Learning and Deep Learning Based Time Series Prediction and Forecasting of Ten Nations' COVID- 19 Pandemic

- **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 vaccin ations** - 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;

Content:

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed

# It is defined by the /python Docker image/docker-python

# For example, here's several helpful packages to load
```

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)


# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all f
iles under the input directory


import matplotlib.pyplot as plt


# plotly

# import plotly.plotly as py

from plotly.offline import init_notebook_mode, iplot, plot

import plotly.express as px

import plotly as py

init_notebook_mode(connected=True)

import plotly.graph_objs as go


from pandas_profiling import ProfileReport

import scipy


# seaborn library

import seaborn as sns


# word cloud library
```

```python
from wordcloud import WordCloud


import os

for dirname, _, filenames in os.walk('/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))


# You can write up to 20GB to the current directory (/working/) that gets preserved as output when you create a version using "Save & Run All"

# You can also write temporary files to /temp/, but they won't be saved outside of the current session
```

```
/input/covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv

/input/covid-world-vaccination-progress/country_vaccinations.csv
```

In [2]:

```python
data = pd.read_csv("/input/covid-world-vaccination-progress/country_vaccinations.csv")

data.head()
```

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million	vaccines	source_name	source_website
0	Afghanistan	AFG	2021-0	0.0	0.0	NaN	NaN	NaN	0.0	0.0	NaN	NaN	Johnson&Johnson, Oxfo	World Health	https://covid19.who.int/

			2-22										rd/AstraZeneca, Pfizer/Bi...	Organization	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi..	World Health Organizati	https://covid19.who.int/

														.	on	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/	

In [3]:

```
report = ProfileReport(data)

report
```

Abstract

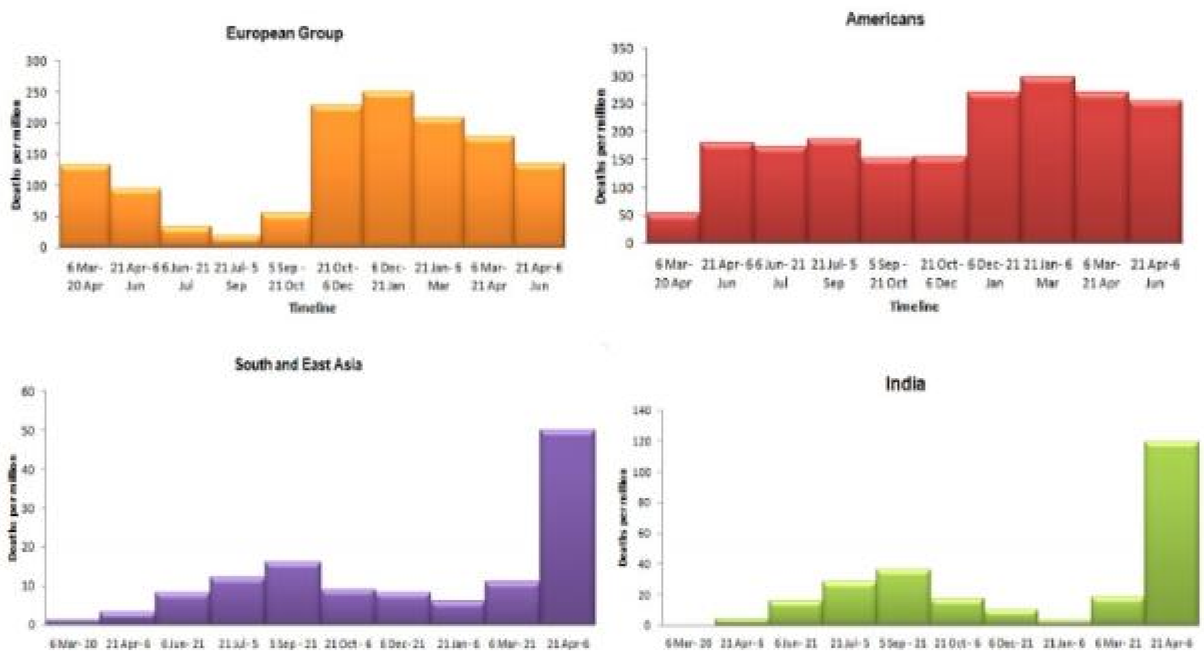
In the paper, the authors investigated and predicted the future environmental circumstances of a COVID-19 to minimize its efects using artifcial intelligence techniques. The experimental investigation of COVID-19 instances has been performed in ten countries, including India, the United States, Russia, Argentina, Brazil, Colombia, Italy, Turkey, Germany, and France

using machine learning, deep learning, and time series models. The confrmed, deceased, and recovered datasets from January 22, 2020, to May 29, 2021, of Novel COVID-19 cases were considered from the Kaggle COVID dataset repository. The country-wise Exploratory Data Analysis visually represents the active, recovered, closed, and death cases from March 2020 to May 2021. The data are pre-processed and scaled using a MinMax scaler to extract and normalize the features to obtain an accurate prediction rate. The proposed methodology employs Random Forest Regressor, Decision Tree Regressor, K Nearest Regressor, Lasso Regression, Linear Regression, Bayesian Regression, Theilsen Regression, Kernel Ridge Regressor, RANSAC Regressor, XG Boost, Elastic Net Regressor, Facebook Prophet Model, Holt Model, Stacked Long Short-Term Memory, and Stacked Gated Recurrent Units to predict active COVID-19 confrmed, death, and recovered cases. Out of diferent machine learning, deep learning, and time series models, Random Forest Regressor, Facebook Prophet, and Stacked LSTM outperformed to predict the best results for COVID-19 instances with the lowest root-mean-square and highest R_2 score values.

Keywords COVID-19 · Prediction · XG Boost · Facebook Prophet · Holt model · Stacked gated recurrent units · RANSAC regressor · Random forest regressor · Stacked long short-term memory

Introduction

Throughout history, the world has confronted several major pandemic and epidemic problems. The frst recorded pandemic occurred in Athens during the Peloponnesian War in 430 BC, followed by the Antonine Plague in 165 A.D., in 250 A.D.—the Cyprian Plague, in 541 A.D.—the Justinian Plague, in the eleventh century—leprosy, in 1350—The Black Death, in 1492—The Columbian Exchange, in 1665— The Great Plague of London, in 1817—The First Cholera Pandemic, in 1855—The Third Plague Pandemic, in 1875— Fiji Measles Pandemic, in 1889—Russian Flu, in 1918— Spanish Flu, in 1957—Asian Flu, in 1981—HIV/AIDS, in 2003-SARS, and 2019—COVID-19 [1]. While still a public health concern, Coronavirus 19 (also known as COVID-19) is an infectious sickness that occurred by the severe acute respiratory syndrome coronavirus 2. The frst recorded case of SARS (severe acute respiratory syndrome) was identifed in December of 2019 in Wuhan, China. The disease has since spread to many other nations and healthcare systems worldwide. At the same time, humans inhale contaminated air, including airborne droplets and particles that are smaller than 0.1 microns, and COVID-19 spreads [2]. Inhalation of these particles is more dangerous when people are closely packed together; nevertheless, they can be inhaled further apart, especially indoors. Infected fuids sprayed on the skin, in the eyes, nose, or mouth, or on surfaces contaminated with them may result in transmission. Someone can carry and spread the disease for up to 20 days even if they have no symptoms. During COVID19,afrstwavebeganinthe spring, which receded signifcantly throughout the summer, and a second wave appeared in the fall of 2020. The initial wave of the epidemic devastated several nations, and many patients perished. The severity of this early phase was exacerbated by a lack of specialist equipment and a lack of understanding of the disease [4]. We all learned from our mistakes during the frst wave of the pandemic, and as a result, our confdence in being able to handle the second wave much better was strong. Despite this, the second wave had considerably greater infection rates, more patients in ICUs, and, in certain countries, more fatalities [5]. Figure 1 depicts the death rates from March 6, 2020, to June 6, 2021, with Europe and the Americas having the most signifcant mortality rates compared to India and South and East Asia. Europe had 1,172,912 death cases, the Americas had 1,926,520, South and East Asia had 739,802 death cases, and India had 402,728 COVID death cases as of July



Related Work

Since 2020, researchers have made signifcant attempts to anticipate the onset of COVID illness in people or the end of the disease around the globe. Keeping this in mind, Shastri et al. [1] suggested a deep learning-based model, such as a recurrent neural network, to forecast the future circumstances of new coronaviruses by studying instances from India and the United States. Ten diferent nations with the most signifcant number of verifed cases were investigated. It was shown that the predictive accuracy of a range of six separate time series modeling approaches for coronavirus epidemic detection varied by Papastefanopoulos et al. [2]. Using an LSTM model, Chimmula et al. [3] predicted the end of the COVID-19 pandemic and worldwide epidemics due to antiviral drugs and improved access to healthcare. Indicating the date of the pandemic's demise, the writers anticipate that it will be fnished by June of 2020. Using a deep learning model, Togacar et al. [4] identifed coronavirus in datasets containing instances of pneumonia, as well as standard X-ray imaging data. The COVID-19 disease can diagnosedwith 99.27% accuracy with the model that the authors used. COVID-19 drug and vaccine research achievements were evaluated using artifcial intelligence techniques in a recent study by Arshadi et al. [5]. In addition, the scientists gave information about the compounds, peptides, and epitopes in the CoronaDB-AI library, which were discovered both in silico and in vitro. Categorizing chest X-rays into two groups was proposed by the researchers led by Elaziz et al. [6]. The accuracy percentage for the frst and second datasets was 96.09% and 98.09%, respectively. Alimadadi et al. [7] presented a Alaska et al. [9] evaluated the efcacy of deep learning models in predicting COVID-19 illness using laboratory data from 600 patients and got 91.89% accuracy. Their approach was also utilized to help medical professionals validate test data and for clinical prediction research. The Johns Hopkins dashboard data, which were the primary source of the Punn et al.'s [10] research, were utilized with machine learning and deep learning models. The team's goal was to grasp the exponential growth of the COVID-19 and then make predictions about how widespread it may become across the country. Table 1 on the left shows the researchers who worked on the forecast and detection of COVID-19

Contribution Outline

The overall goal of this research is to build models that can calculate two necessary evaluative measures: RMSE and R_2 Score for confrmed, death, and recovered cases from ten diferent nations to help future forecasts. The steps are as follows:

Step 1: Initially, data are pre-processed to capture char

acteristics utilizing various variables, such as active cases,

recovered cases, and COVID-19 fatality cases.

Step 2: Exploratory Data Analysis of COVID-19's active

cases, closed cases, confrmed cases, recovered cases, and

death cases are calculated to summarize or interpret the

information that is hidden in rows or columns, and scaling

techniques such as Min–Max have been applied to normalize

each feature that is obtained from these attributes.

Step 3: Later, utilizing confrmed cases, recovered cases,

and death cases from 10 diferent nations, the gathered data

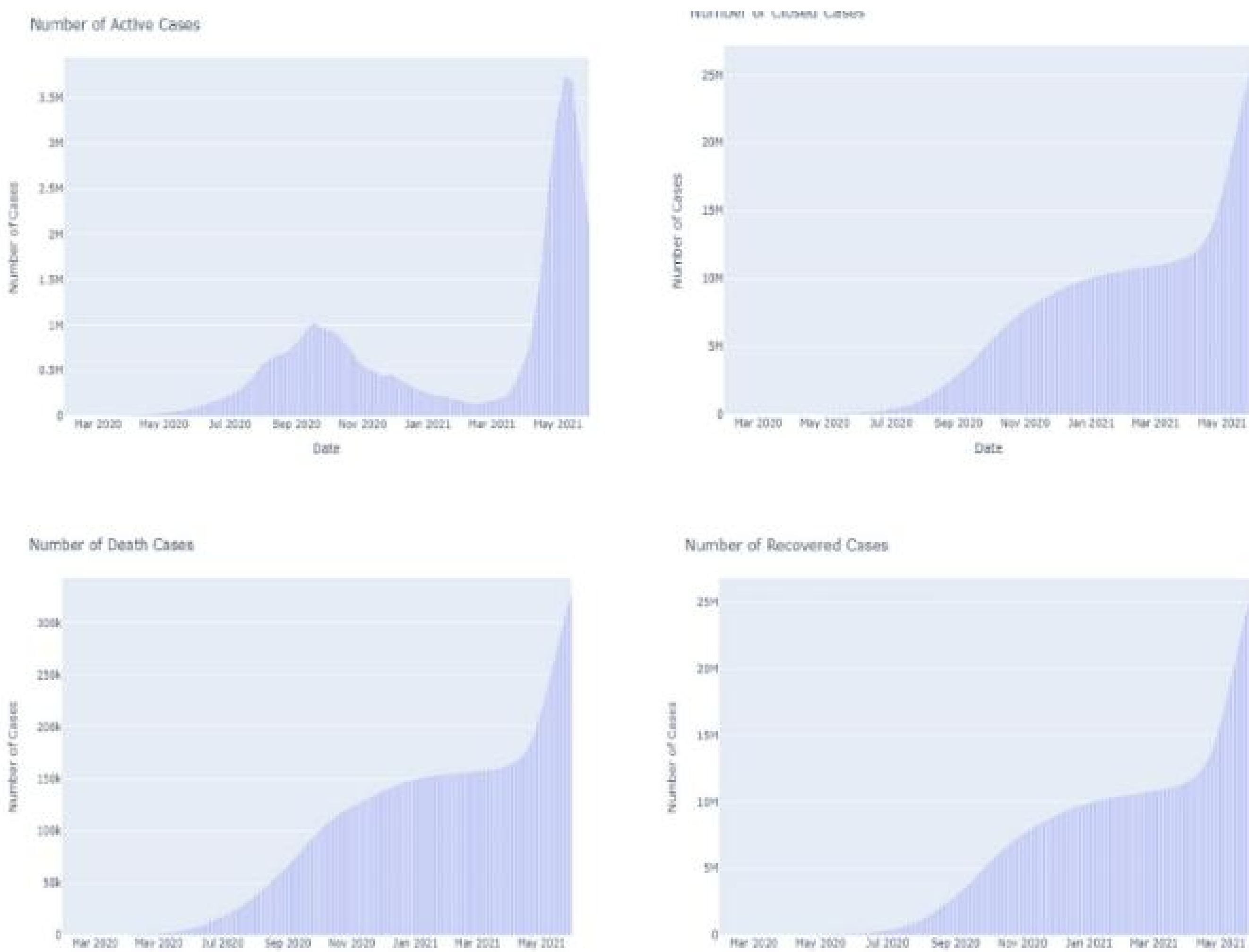
were used to anticipate the future conditions of a new Coro
naVirus. To get the fndings, several machine learning mod
els, time series models, and deep learning models were used,

alysis of the existing work

me	Dataset	Technique	Results	Limitations
[11]	1065 CT pathogenic images	Transfer Learning Model, CNN, Graph-Net	Accuracy: 89.5% Specificity: 0.88 Sensitivity: 0.87	Factors such as low signal-to-noise ratio and complex data integration led to reducing the efficacy of deep learning models
hyay et al. [12]	Data collected from WHO (Jan. 16–20,2020)	Long Short-Term Memory, Gated Recurrent Unit	Accuracy: 87%	The model failed to represent the spatio temporal components of the LSTM network
al. [4]	Data collected from Qatar University	Stacking Technique, Fuzzy Color, Deep Learning Model	Classification accuracy: 99.27%	Publications of COVID-19 images were limited. The system did not work with the low resolution and different size input images
l. [1]	Dataset was sourced from the Ministry of Health and Family Welfare	Deep Neural Network, Long Short-Term Memory, Recurrent Neural Network, Polynomial Regression	Accuracy ConvLSTM: 98%	The comparative analysis had been performed only for two countries
al. [13]	COVID-19 chest X-ray dataset	Bayesian Deep Learning	Accuracy: 80%	After reviewing the data, it was impossible to conclude anything regarding markers for imaging, discoveries concerning improved diagnosis and therapy for COVID-19
[10]	Data collected from Jan 22, 2020 to Apr 1 2020 at Johns Hopkins University	Support Vector Machine, Deep Neural Network, Long Short-Term Memory, Polynomial Regression	RMSE confirmed: 455.92 Death: 117.94 Recovered: 809.71	The study needed to work on more algorithms to enhance the RMSE score
l. [9]	Samples collected from the Albert Einstein Israelite Hospital in Sao Paulo, Brazil	Artificial Neural Network, Convolution Neural Network, Long Short-Term Memory	Accuracy: 86.66% F1 Score: 91.89% Recall: 99.42% AUC: 62.50% Precision: 86.75%	The primary disadvantage of the study was the sheer amount of data. To increase the number of patients for whom the lab findings could not be assessed, the procedure was applied on 600 patients
. [14]	180 COVID-19 and 200 chest X-ray images	CNN model, SVM, ResNet50	Accuracy: 91.6%	The study needed to incorporate work on different imagistic patterns of COVID-19
l. [15]	337 patient images from real-world data	Deep learning, nCOVnet	Accuracy: 97.62%	The system worked on a small dataset
. [6]	Dataset collected from Joseph Paul Cohen and Paul Morrison Lan Dao	Manta Ray Foraging Optimization, Fractional Multichannel Exponent Moments	Accuracy: 96.09% Accuracy: 98.09%	The system dealt with resource limitations and high CPU time

Table 2 Analysis of COVID-19 cases among the top ten countries

Countries	Confirmed cases	Death cases	Recovered cases
India	27,894,800	325,972	25,454,320
USA	33,251,939	594,306	–
Russia	4,995,613	118,781	46,16,422
Argentina	3,732,263	77,108	3,288,467
Brazil	16,471,600	461,057	14,496,224
Colombia	3,363,061	87,747	3,141,549
Italy	4,213,055	126,002	3,845,087
Turkey	33,251,939	47,271	5,094,279
Russia	4,995,613	118,781	4,616,422
Germany	3,684,672	88,413	3,479,700

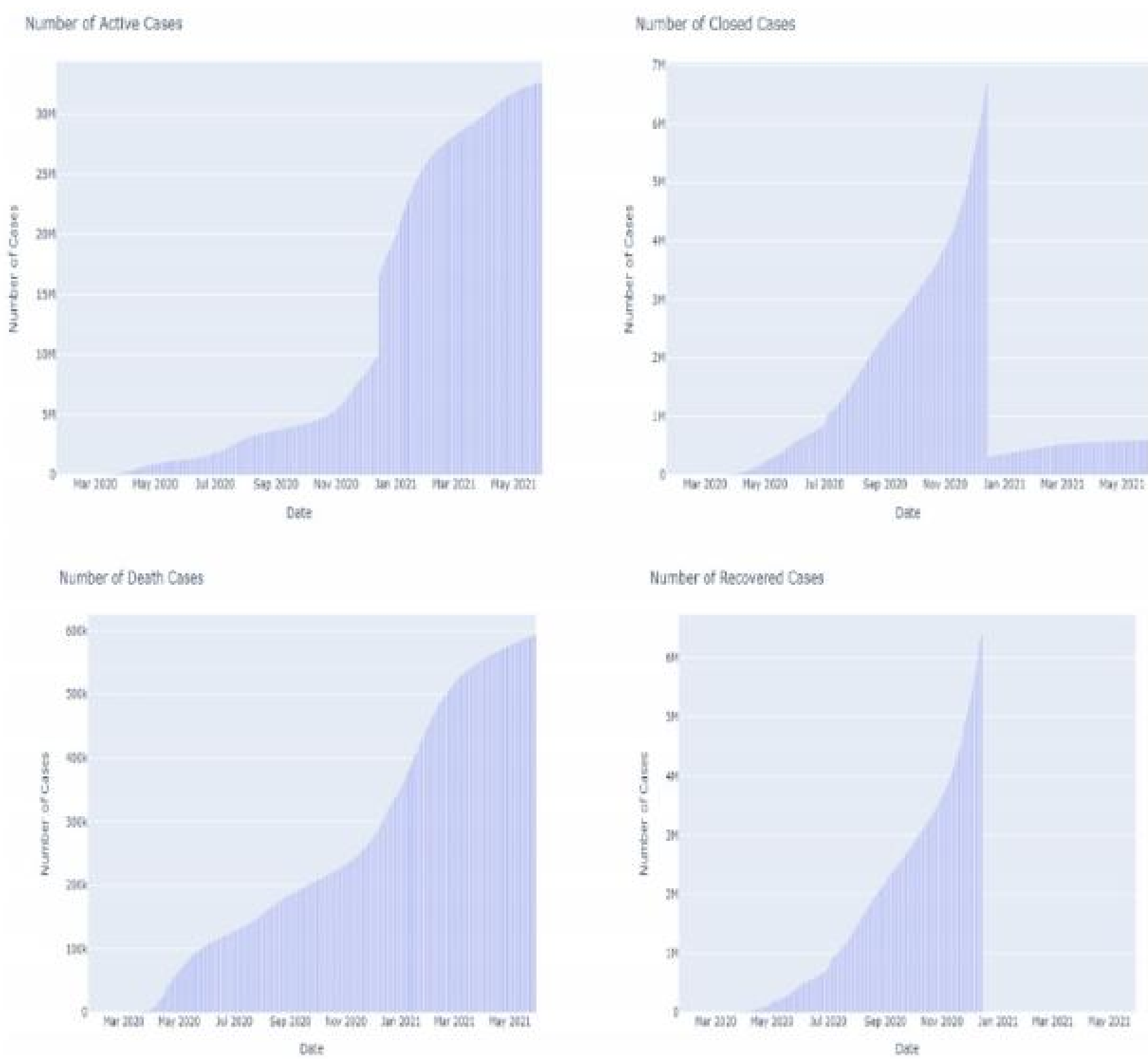


This work employed an exploratory analysis of ten diferent countries after pre-processing to assess its features via statistical graphs. Figures shown below depicts the graphical analysis of active cases, death cases, closed points, and recovered cases that have been recorded from Jan 2020 to May 2021. It wasdetermined in Fig. 3 that 27,894,800 cases had been confrmed, 2,114,508 were still active, 325,972 had died, 25,780,292 had been closed, and 25,454,320 people had been recovered from Jan 2020 to 29 May 2021. Additionally, the numbers of confrmed cases, deaths, and recovered cases each day were, respectively, 57,397, 671, and 52,375. According to Fig. 4, it has been discovered that US has 3,325,189,940 instances with high certainty, 3,266,576,333 cases with moderate certainty, 594,306 cases with low certainty, and 0 cases with a medium certainty which were seen from January 1st, 2020 to May 29th, 2021. Additionally, the daily average of confrmed cases was reported as 673,128, while the daily average of deaths was recorded as 12,030. Finally, the daily average of recovered cases was recorded as 0. As demonstrated in Fig. 5, the numbers of confrmed, active, and death cases have been as follows: 49,956,313.0, 260,410.0, 118,781.0, 47,352,203.0, and 46,164,322.0 from January 1, 2020 to May 29, 2021. Finally, the total number of confrmed cases was 10,300. The number of death cases was 245, and the total number of recovered cases was 9518. In Fig. 6, it was discovered that Argentina has reported 373,263.0 total cases, with 366,688.0 currently active cases,77,108currentlyknowndeathcases,336,575 previously known to be closed cases, and 328,467 previously known recovered cases from January 1st, 2020 to May 31st, 2021.

In addition to this, there were around 8239.0 confrmed

cases of the disease each day, approximately 170.0 deaths

per day, and approximately 7259.0 recovered cases per day



COVİD 19 WORLD VACCINATION PROGRESS

The data 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 vaccin ations** - 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;

Content:

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed

# It is defined by the /python Docker image: https://github.com/docker-python

# For example, here's several helpful packages to load



import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)



# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all f
iles under the input directory



import matplotlib.pyplot as plt



# plotly

# import plotly.plotly as py

from plotly.offline import init_notebook_mode, iplot, plot

import plotly.express as px

import plotly as py

init_notebook_mode(connected=True)

import plotly.graph_objs as go



from pandas_profiling import ProfileReport

import scipy
```

```
# seaborn library

import seaborn as sns



# word cloud library

from wordcloud import WordCloud



import os

for dirname, _, filenames in os.walk('/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))



# You can write up to 20GB to the current directory (/working/) that gets preserved a
s output when you create a version using "Save & Run All"

# You can also write temporary files to /temp/, but they won't be saved outside of th
e current session
```

```
/input/covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv

/input/covid-world-vaccination-progress/country_vaccinations.csv
```

In [2]:

```
data = pd.read_csv("/input/covid-world-vaccination-progress/country_vaccinations.csv")

data.head()
```

Out[2]:

	country	iso_co	dat	total_vaccinatio	people_vaccinat	people_fully_vaccinate	daily_vaccinations_ra	daily_vaccinati	total_vaccinations_per_hundr	people_vaccinated_per_hundr	people_fully_vaccinated_per_hund	daily_vaccinations_per_milli	vaccines	source_na	source_website

		d e	e	ns	ed	d	w	ons	ed	ed	red	on		me	
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	0.0	0.0	NaN	NaN	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	NaN	NaN	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

In [3]:

```
report = ProfileReport(data)

report
```

```
Summarize dataset: 100%

106/106 [00:26<00:00, 3.82it/s, Completed]

Generate report structure: 100%

1/1 [00:07<00:00, 7.25s/it]

Render HTML: 100%

1/1 [00:03<00:00, 3.96s/it]
```

Out[3]:

Missing Data

We will fix some shortcomings to make data visualization easier and more understandable.

In [4]:

```
data.info()
```

```
<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
```

In [5]:

```
data.shape
```

Out[5]:

```
(86512, 15)
```

In [6]:

```
data.isna().sum()
```

Out[6]:

```
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
```

As can be seen, there is quite much missing data.

Drop the total_vaccinations column from these deficiencies first

In [7]:

```
data = data.drop(data[data.total_vaccinations.isna()].index)
```

In [8]:

```
data.isna().sum()
```

Out[8]:

```
country                                  0
iso_code                                 0
date                                     0
total_vaccinations                       0
people_vaccinated                     2717
people_fully_vaccinated               5097
daily_vaccinations_raw                8245
daily_vaccinations                     223
total_vaccinations_per_hundred           0
people_vaccinated_per_hundred         2717
people_fully_vaccinated_per_hundred   5097
```

```
daily_vaccinations_per_million          223
vaccines                                  0
source_name                               0
source_website                            0
dtype: int64
```

As you can see the missing data in the total_vaccinations column has been removed.

Now let's remove the missing data from the people_vaccinated column

In [9]:

```python
check_data = data.drop(data[data.people_vaccinated.isna()].index)
```

In [10]:

```python
check_data.head()
```

Out[10]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million	vaccines	source_name	source_website
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	0.00	0.00	NaN	NaN	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

6	Afghanistan	AFG	2021-02-28	8200.0	8200.0	NaN	NaN	1367.0	0.02	0.02	NaN	34.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
22	Afghanistan	AFG	2021-03-16	54000.0	54000.0	NaN	NaN	2862.0	0.14	0.14	NaN	72.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
44	Afghanistan	AFG	2021-04-07	120000.0	120000.0	NaN	NaN	3000.0	0.30	0.30	NaN	75.0	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
59	Afghanist	AFG	2021-	240000.0	240000.0	NaN	NaN	8000.0	0.60	0.60	NaN	201.0	Johnson &Johnson,	World Healt	https://covid19.who.i

	an		04-22										Oxford/AstraZeneca, Pfizer/Bi...	h Organization	nt/

As you can see the missing data in the people_vaccinated column has been removed.

Let's look at the values between the columns by looking at the correlation map

```
In [11]:

plt.subplots(figsize = (10,10))

sns.heatmap(data.corr(), annot = True, square = True)

plt.show()
```

people_vaccinated and people_vaccinated_per_hundred

The data of the total_vaccinations column and the people_vaccinated column look almost the same.

As can bee seen from the heatmap, these features have almost ideal correlation.

```
In [12]:

diff = check_data.total_vaccinations.mean() - check_data.people_vaccinated.mean()

diff_per_hundred = check_data.total_vaccinations_per_hundred.mean() - check_data.peop
le_vaccinated_per_hundred.mean()



data.people_vaccinated = data.people_vaccinated.fillna(data.total_vaccinations - diff)

data.people_vaccinated_per_hundred = data.people_vaccinated_per_hundred.fillna(data.t
otal_vaccinations_per_hundred - diff_per_hundred)
```

Let's check if everything ok.

In [13]:

```
data.isna().sum()
```

Out[13]:

```
country                                  0
iso_code                                 0
date                                     0
total_vaccinations                       0
people_vaccinated                        0
people_fully_vaccinated               5097
daily_vaccinations_raw                8245
daily_vaccinations                     223
total_vaccinations_per_hundred           0
people_vaccinated_per_hundred            0
people_fully_vaccinated_per_hundred   5097
daily_vaccinations_per_million         223
vaccines                                 0
source_name                              0
source_website                           0
dtype: int64
```

daily_vaccinations_raw and daily_vaccinations

The data of the daily_vaccinations column and the daily_vaccinations_raw column look almost the same.

As can bee seen from the heatmap, these features have almost ideal correlation.

In [14]:

```
diff = check_data.daily_vaccinations.mean() - check_data.daily_vaccinations_raw.mean()


data.daily_vaccinations_raw = data.daily_vaccinations_raw.fillna(data.daily_vaccinati
ons - diff)
```

In [15]:

```
data.isna().sum()
```

Out[15]:

```
country                                 0
iso_code                                0
date                                    0
total_vaccinations                      0
people_vaccinated                       0
people_fully_vaccinated              5097
daily_vaccinations_raw                223
daily_vaccinations                    223
total_vaccinations_per_hundred          0
people_vaccinated_per_hundred           0
people_fully_vaccinated_per_hundred  5097
daily_vaccinations_per_million        223
vaccines                                0
source_name                             0
source_website                          0
dtype: int64
```

In [16]:

```
data.daily_vaccinations = data.daily_vaccinations.fillna(0)

data.daily_vaccinations_raw = data.daily_vaccinations_raw.fillna(0)
```

In [17]:

```
data.isna().sum()
```

Out[17]:

```
country                                 0
iso_code                                0
date                                    0
total_vaccinations                      0
people_vaccinated                       0
people_fully_vaccinated              5097
daily_vaccinations_raw                  0
daily_vaccinations                      0
total_vaccinations_per_hundred          0
people_vaccinated_per_hundred           0
people_fully_vaccinated_per_hundred  5097
daily_vaccinations_per_million        223
vaccines                                0
source_name                             0
source_website                          0
dtype: int64
```

The data of the total_vaccinations column and the people_fully_vaccinated column look almost the same.

As can bee seen from the heatmap, these features have almost ideal correlation.

people_fully_vaccinated

In [18]:

```
diff = check_data.total_vaccinations.mean() - check_data.people_fully_vaccinated.mean
()
```

```
data.people_fully_vaccinated = data.people_fully_vaccinated.fillna(data.total_vaccina
tions - diff)
```

In [19]:

```
data.isna().sum()
```

Out[19]:

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
people_fully_vaccinated                0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred 5097
daily_vaccinations_per_million       223
vaccines                               0
```

```
source_name                     0
source_website                  0
dtype: int64
```

people_fully_vaccinated_per_hundred

The data of the total_vaccinations_per_hundred column and the people_fully_vaccinated_per_hundred column look almost the same.

As can bee seen from the heatmap, these features have almost ideal correlation.

In [20]:

```
diff = check_data.total_vaccinations_per_hundred.mean() - check_data.people_fully_vac
cinated_per_hundred.mean()



data.people_fully_vaccinated_per_hundred = data.people_fully_vaccinated_per_hundred.f
illna(data.total_vaccinations_per_hundred - diff)
```

In [21]:

```
data.isna().sum()
```

Out[21]:

```
country                         0
iso_code                        0
date                            0
total_vaccinations              0
people_vaccinated               0
people_fully_vaccinated         0
daily_vaccinations_raw          0
daily_vaccinations              0
total_vaccinations_per_hundred  0
```

```
people_vaccinated_per_hundred           0
people_fully_vaccinated_per_hundred     0
daily_vaccinations_per_million        223
vaccines                                0
source_name                             0
source_website                          0
dtype: int64
```

Since there is not much similarity between them and the others in the daily_vaccinations_per_million correlation map, we will assign the value 0 instead of the missing data.

In [22]:

```
data.daily_vaccinations_per_million = data.daily_vaccinations_per_million.fillna(0)
```

In [23]:

```
data.isna().sum()
```

Out[23]:

```
country                               0
iso_code                              0
date                                  0
total_vaccinations                    0
people_vaccinated                     0
people_fully_vaccinated               0
daily_vaccinations_raw                0
daily_vaccinations                    0
total_vaccinations_per_hundred        0
```

```
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
daily_vaccinations_per_million         0
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

There is no missing data in our columns.

Our missing data has been cleared.

Everything worked fine

iso_code

Let's see if there is any missing data in iso_code

In [24]:

```
data[data.iso_code.isna()].country.unique()
```

Out[24]:

```
array([], dtype=object)
```

Thats the iso-codes which are used for these countries : GB-ENG for England, NC for Northern Cyprus, GB-NIR for Northern Ireland, GB-SCT for Scotland, GB-WLS for Wales.

We will fill missing iso-codes with appropriate ones.

In [25]:

```
data[data.country == 'England'] = data[data.country == 'England'].fillna('GB-ENG')

data[data.country == 'Northern Ireland'] == data[data.country == 'Northern Ireland'].
fillna('GB-NIR')

data[data.country == 'Scotland'] = data[data.country == 'Scotland'].fillna('GB-SCT')
```

```python
data[data.country == 'Wales'] = data[data.country == 'Wales'].fillna('GB-WLS')

data = data.fillna('NC')
```

In [26]:

```python
data.isna().sum()
```

Out[26]:

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
people_fully_vaccinated                0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
daily_vaccinations_per_million         0
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

Finally we managed to organize the lost data

Lets transform date column

In [27]:

```
data["date"] = pd.to_datetime(data["date"])

data = data.sort_values("date", ascending = True )

data["date"] = data["date"].dt.strftime("%Y-%m-%d")
```

In [28]:

```
unique_dates = data["date"].unique()
```

In [29]:

```
data.head()
```

Out[29]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million	vaccines	source_name	source_website
58517	Norway	NOR	2020-12-02	0.0	0.0	-1.834694e+07	0.0	0.0	0.0	0.0	-44.719168	0.0	Moderna, Pfizer/BioNTech	Norwegian Institute of Public Health	https://github.com/folkehelseinstituttet/surve...
5851	Norwa	NOR	2020-	0.0	0.0	-1.834694e+07	0.0	0.0	0.0	0.0	-44.719168	0.0	Moderna, Pfizer/B	Norwegian Ins	https://github.com/folkehelseinstituttet/s

8	y		12-03										ioNTech	titute of Public Health	urve...
43117	Latvia	LVA	2020-12-04	1.0	1.0	-1.834693e+07	0.0	0.0	0.0	0.0	-44.719168	0.0	Johnson&Johnson, Moderna, Novavax, Pfizer/BioN...	National Health Service	https://data.gov.lv/dati/eng/dataset/covid19-v...
58519	Norway	NOR	2020-12-04	0.0	0.0	-1.834694e+07	0.0	0.0	0.0	0.0	-44.719168	0.0	Moderna, Pfizer/BioNTech	Norwegian Institute of Public Health	https://github.com/folkehelseinstituttet/surve...
58520	Norwa	NOR	2020-12	0.0	0.0	-1.834694e+07	0.0	0.0	0.0	0.0	-44.719168	0.0	Moderna, Pfizer/BioNTec	Norwegian Institute	https://github.com/folkehelseinstituttet/surve...

	y		-05										h	of Public Health	

our dates are listed

Data Visualization

First, let's watch total_vaccinations and daily_vaccinations animatedly on the world map.

Let's take a look at the rate of vaccination in countries

In [30]:

```
fig = px.choropleth(

    data,

    locations="iso_code",

    color="total_vaccinations",

    title='Number of people vaccinated',

    color_continuous_scale='viridis',

    animation_frame="date",

    projection = "natural earth",

    range_color = [0,5000000],



)
```

date=2020-12-022020-12-022021-03-292021-07-242021-11-182022-03-1501M2M3M4M5Mtotal_vaccinationsNumber of people vaccinated▶■

Animated world map by date of total vaccinations by country

In [31]:

```
fig = px.choropleth(

    data,

    locations= "iso_code",

    color = "daily_vaccinations",

    animation_frame = "date",

    color_continuous_scale= "viridis",

    projection= "natural earth",

    range_color= [0,1000000] ,

    title = "Number of daily vaccinations"

)

fig.show()
```

date=2020-12-022020-12-022021-03-292021-07-242021-11-182022-03-1500.2M0.4M0.6M0.8M1Mdaily_vaccinationsNumber of daily vaccinations▶■

Animated world map by date of daily vaccinations by country

First we will create a new table by selecting the columns we will use.

In [32]:

```
columns = ["country","iso_code","total_vaccinations","people_vaccinated","total_vacci
nations_per_hundred","vaccines","daily_vaccinations"]

vacc_data = data[columns].groupby("country").max().sort_values("total_vaccinations",
ascending = True)
```

In [33]:

```
vacc_data.head()
```

Out[33]:

	iso_code	total_vaccinations	people_vaccinated	total_vaccinations_per_hundred	vaccines	daily_vaccinations
country						
Pitcairn	PCN	94.0	47.0	200.00	Oxford/AstraZeneca	1.0
Tokelau	TKL	1936.0	968.0	141.52	Pfizer/BioNTech	23.0
Niue	NIU	4161.0	1650.0	257.81	Pfizer/BioNTech	87.0
Montserrat	MSR	4211.0	1897.0	84.54	Oxford/AstraZeneca	53.0
Falkland Islands	FLK	4407.0	2632.0	124.91	Oxford/AstraZeneca	189.0

In [34]:

```
vacc_data.info()
```

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

Index: 223 entries, Pitcairn to China

Data columns (total 6 columns):

 #   Column                          Non-Null Count  Dtype

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

 0   iso_code                        223 non-null    object
```

```
 1   total_vaccinations             223 non-null    float64
 2   people_vaccinated              223 non-null    float64
 3   total_vaccinations_per_hundred 223 non-null    float64
 4   vaccines                       223 non-null    object
 5   daily_vaccinations             223 non-null    float64
dtypes: float64(4), object(2)
memory usage: 12.2+ KB
```

In [35]:

```
fig = px.choropleth(
    vacc_data,
    locations= "iso_code",
    color = "total_vaccinations_per_hundred",
    title = "Number of total vaccinations per hunderd",
    color_continuous_scale= "rainbow"
)
fig.show("notebook")
```

50100150200250300total_vaccinations_per_hundredNumber of total vaccinations per hunderd

As can be seen on the map, countries have vaccination percentages.

Let's draw a map to see which countries these vaccines are used in.

In [36]:

```
fig = px.choropleth(
    locations = vacc_data.iso_code,
    color = vacc_data.vaccines,
```

```
    title = "name of the vaccine",
    color_continuous_scale= "rainbow"
)
```

colorOxford/AstraZenecaPfizer/BioNTechSinopharm/BeijingModernaOxford/AstraZeneca, Pfizer/BioNTechModerna, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZenecaModerna, Oxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingPfizer/BioNTech, Sputnik VJohnson&Johnson, Moderna, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sputnik VJohnson&Johnson, ModernaJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/BeijingJohnson&Johnson, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingPfizer/BioNTech, Sinopharm/Beijing, Sputnik VOxford/AstraZeneca, Sinopharm/BeijingOxford/AstraZeneca, Pfizer/BioNTech, SinovacCovaxin, Oxford/AstraZeneca, Sinopharm/BeijingOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, SinovacModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VCovaxin, Oxford/AstraZenecaJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingPfizer/BioNTech, Sinopharm/BeijingOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacCovaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Moderna, Novavax, Pfizer/BioNTechSinopharm/Beijing, Sputnik VJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VJohnson&Johnson, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VEpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VAbdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VOxford/AstraZeneca, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VModerna, Pfizer/BioNTech, Sinopharm/Beijing, SinovacPfizer/BioNTech, SinovacModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacQazVac, Sinopharm/Beijing, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik VCovaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacAbdala, Soberana Plus, Soberana02Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCOVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik VPfizer/BioNTech, Sinovac, TurkovacEpiVacCorona, Sputnik VCanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VAbdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VCanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCovaxin, Oxford/AstraZeneca, Sputnik VCanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001name of the vaccine

Vaccine types can be seen on the sides according to the colors of the countries. By looking at this map, it can be seen which country has which vaccine.

How many people have been vaccinated

First, let's look at the statistics of the countries, then let's show these countries on the world map.

In [37]:

```
vacc_country = data.groupby(["country", "iso_code", "vaccines"])['total_vaccinations',
```

```
                                                                                 'total_vaccina
tions_per_hundred',
                                                                                'daily_vaccinat
ions',
                                                                                'daily_vaccinat
ions_per_million',
                                                                                'people_vaccina
ted',
                                                                                'people_vaccina
ted_per_hundred',
                                                                                 'people_fully_
vaccinated', 'people_fully_vaccinated_per_hundred'
                                                                                ].max().reset_i
ndex()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:


Indexing with multiple keys (implicitly converted to a tuple of keys) will be dep
recated, use a list instead.
```

In [38]:

```
def trace_bar(data, feature, title, xlab, ylab,color):
    data = data.sort_values(feature, ascending=False)
    trace = go.Bar(
            x = data['country'],
            y = data[feature],
            marker=dict(color=color),
            text=data['country']
        )
```

```
data = [trace]


layout = dict(title = title,
        xaxis = dict(
            title = xlab,
            showticklabels=True,
            tickangle=45,
            zeroline=True,
            zerolinewidth=1,
            zerolinecolor='grey',
            showline=True,
            linewidth=2,
            linecolor='black',
            mirror=True,
            tickfont=dict(
                    size=10,
                    color='black'),),



        yaxis = dict(
            title = ylab,
            gridcolor='lightgrey',
            zeroline=True,
            zerolinewidth=1,
```

```
                zerolinecolor='grey',

                showline=True,

                linewidth=2,

                linecolor='black',

                mirror=True),


            plot_bgcolor = 'rgba(0, 0, 0, 0)',

            paper_bgcolor = 'rgba(0, 0, 0, 0)',

            hovermode = 'closest'

             )

    fig = dict(data = data, layout = layout)

    iplot(fig)
```

In [39]:

```
trace_bar(vacc_country, 'total_vaccinations', 'Vaccination total per country', 'Count
ry', 'Vaccination total', "purple" )
```

ChinaIndiaUnited StatesBrazilIndonesiaJapanBangladeshPakistanVietnamMexicoGermanyRussiaPhilippinesTurkeyIranFranceUnited KingdomItalyThailandSouth KoreaEnglandArgentinaSpainCanadaColombiaEgyptMalaysiaPeruSaudi ArabiaAustraliaMoroccoPolandChileTaiwanMyanmarUzbekistanNepalSri LankaVenezuelaCambodiaCubaNetherlandsSouth AfricaEcuadorUkraineNigeriaEthiopiaMozambiqueBelgiumUnited Arab EmiratesPortugalSwedenKazakhstanGreeceRwandaAustriaIsraelIraqAngolaCzechiaKenyaUgandaRomaniaHungaryGuatemalaSwitzerlandDominican RepublicHong KongSingaporeAlgeriaAzerbaijanDenmarkTunisiaBoliviaGhanaScotlandHondurasFinlandBelarusNorwayNew ZealandIrelandTajikistanEl SalvadorCote d'IvoireCosta RicaLaosJordanNicaraguaZimbabweSerbiaParaguayUruguayPanamaKuwaitTurkmenistanSlovakiaOmanWalesQatarSudanGuineaAfghanistanMongoliaLebanonCroatiaTanzaniaLithuaniaBulgariaNorthern IrelandPalestineBahrainLibyaZambiaSyriaBeninSloveniaKyrgyzstanLatviaGeorgiaTogoAlbaniaNigerMauritaniaBotswanaSenegalMauritiusSomaliaBurkina FasoSierra LeoneMoldovaArmeniaMalawiEstoniaBosnia and HerzegovinaMaliNorth

MacedoniaKosovoBhutanCyprusCameroonTrinidad and TobagoJamaicaFijiMadagascarTimorLuxembourgMaltaMacaoLiberiaBruneiCentral African RepublicDemocratic Republic of CongoMaldivesLesothoGuyanaNamibiaCongoYemenIcelandCape VerdeMontenegroComorosNorthern CyprusSouth SudanPapua New GuineaGuinea-BissauGabonEswatiniSurinameEquatorial GuineaBelizeChadNew CaledoniaFrench PolynesiaGambiaSolomon IslandsBahamasBarbadosSamoaHaitiCuracaoJerseySao Tome and PrincipeSeychellesVanuatuIsle of ManArubaDjiboutiAndorraGuernseyTongaCayman IslandsKiribatiBermudaAntigua and BarbudaSaint LuciaGibraltarFaeroe IslandsGrenadaGreenlandLiechtensteinSaint Vincent and the GrenadinesTurks and Caicos IslandsSan MarinoChinaBangladeshPhilippinesThailandColombiaMoroccoNepalSouth AfricaBelgiumRwandaKenyaDominican RepublicTunisiaBelarusCote d'IvoireSerbiaSlovakiaAfghanistanBulgariaSyriaTogoMauritiusMalawiBhutanMadagascarBruneiNamibiaComorosEswatiniFrench PolynesiaHaitiIsle of ManCayman IslandsFaeroe IslandsSan MarinoBonaire Sint Eustatius and SabaTuvaluPitcairn00.5B1B1.5B2B2.5B3B

Vaccination total per countryCountryVaccination total

In [40]:

```
trace_bar(vacc_country, 'daily_vaccinations', 'Daily vaccinations per country', 'Country', 'Daily vaccinations', "red" )
```

ChinaIndiaBangladeshUnited StatesPakistanJapanBrazilIndonesiaVietnamMexicoPhilippinesEthiopiaTurkeyIranGermanyRussiaUnited KingdomSouth KoreaThailandEnglandFranceSpainItalyEgyptTaiwanMalaysiaCanadaSri LankaVenezuelaArgentinaGhanaIraqUzbekistanColombiaSaudi ArabiaMoroccoPeruNepalMyanmarEcuadorKazakhstanPolandMozambiqueCubaRwandaNicaraguaChileAustraliaUgandaCambodiaNetherlandsNigeriaAlgeriaUkraineKenyaSouth AfricaIsraelDominican RepublicAngolaGuineaTunisiaSudanUnited Arab EmiratesBelgiumHondurasPortugalLaosBotswanaDenmarkRomaniaHungaryMongoliaCote d'IvoireAustriaSwedenGreeceBoliviaZimbabweCzechiaGuatemalaSwitzerlandBhutanIrelandParaguaySyriaJordanCosta RicaHong KongAzerbaijanNigerTajikistanNew ZealandSingaporeBelarusEl SalvadorFinlandPanamaScotlandAfghanistanNorwayMauritaniaSerbiaWalesOmanBeninSenegalMaliLebanonUruguayCameroonPalestineTurkmenistanKuwaitBurkina FasoLiberiaTogoTanzaniaQatarZambiaSlovakiaCroatiaSomaliaSierra LeoneGeorgiaLithuaniaBulgariaDemocratic Republic of CongoLibyaNorthern IrelandKyrgyzstanBahrainMalawiMauritiusKosovoSloveniaCentral African RepublicArmeniaTrinidad and TobagoAlbaniaGuinea-BissauLatviaNamibiaMoldovaNorth MacedoniaLesothoFijiBosnia and HerzegovinaJamaicaCyprusMontenegroTimorEswatiniMadagascarYemenEstoniaSouth SudanGuyanaMaltaBruneiMaldivesIcelandLuxembourgGabonCongoMacaoCape

VerdeSurinameIsle of ManNorthern CyprusComorosPapua New GuineaNew CaledoniaEquatorial GuineaBelizeBarbadosHaitiChadGambiaCuracaoBahamasSamoaFrench PolynesiaArubaSao Tome and PrincipeSeychellesSolomon IslandsTongaVanuatuSaint LuciaAndorraGuernseyKiribatiAntigua and BarbudaJerseyBermudaGreenlandGibraltarCayman IslandsFaeroe IslandsSint Maarten (Dutch part)DominicaSaint Vincent and the GrenadinesCook IslandsDjiboutiSaint Kitts and NevisLiechtensteinSan MarinoGrenadaNauruTurks and Caicos IslandsChinaBrazilTurkeyThailandTaiwanGhanaPeruMozambiqueUgandaKenyaTunisiaLaosCote d'IvoireCzechiaSyriaTajikistanPanamaWalesUruguayLiberiaCroatiaDemocratic Republic of CongoMauritiusAlbaniaLesothoTimorGuyanaGabonNorthern CyprusBarbadosSamoaTongaAntigua and BarbudaFaeroe IslandsSaint Kitts and NevisBurundiBonaire Sint Eustatius and SabaPitcairn05M10M15M20M

Daily vaccinations per countryCountryDaily vaccinations

In [41]:

```
trace_bar(vacc_country, 'daily_vaccinations_per_million', 'Daily vaccinations per mil
lion per country', 'Country','Daily vaccinations per million', "magenta" )
```

BhutanIsle of ManBotswanaNiueFalkland IslandsNauruNicaraguaCook IslandsMongoliaGibraltarWallis and FutunaCubaGuernseySaint HelenaArubaTaiwanSeychellesRwandaDenmarkAndorraBangladeshSri LankaAnguillaPitcairnCuracaoSan MarinoSint Maarten (Dutch part)IcelandFaeroe IslandsTuvaluIsraelGreenlandEcuadorBermudaLaosWalesLiechtensteinTongaKazakhstanIrelandMontenegroBruneiNew CaledoniaSouth KoreaVietnamCambodiaCosta RicaDominican RepublicTokelauMalaysiaPanamaSamoaVenezuelaMauritiusNew ZealandMaltaJapanFijiNetherlandsBarbadosNorthern CyprusChileUnited Arab EmiratesChinaCayman IslandsHondurasSaint Kitts and NevisTurkeyPortugalEnglandPhilippinesMaldivesIranMauritaniaSingaporeSpainCanadaUnited KingdomUruguayTunisiaCyprusGhanaGermanyBelgiumTurks and Caicos IslandsNepalHungaryScotlandFinlandAntigua and BarbudaGuineaTrinidad and TobagoMexicoNorwayBahrainQatarUzbekistanParaguayNorthern IrelandAustriaKiribatiJerseyThailandCape VerdeEthiopiaDominicaOmanPeruGuyanaAustraliaHong KongSaudi ArabiaFrench PolynesiaLuxembourgBelizeBritish Virgin IslandsEl SalvadorItalySao Tome and PrincipeSwedenFranceKosovoMoroccoMontserratSwitzerlandUnited StatesSurinameGreeceIraqSaint LuciaMacaoMozambiqueLithuaniaArgentinaPakistanEswatiniKuwaitSerbiaSloveniaCzechiaLatviaBoliviaBrazilPolandGuinea-BissauCroatiaJordanTajikistanAzerbaijanBahamasTimorLebanonBelarusColombiaPalestineMonacoEstoniaLiberiaSaint Vincent and the GrenadinesNorth MacedoniaBonaire Sint Eustatius and SabaGeorgiaIndiaRussiaRomaniaZimbabweIndonesiaTurkmenistanLesothoSlovakiaMyan marNamibiaVanuatuUgandaComorosAlbaniaArmeniaEgyptUkraineAlgeriaAngolaGuatemalaGrenadaSyriaBeninJamaicaTogoCote d'IvoireKenyaBosnia and HerzegovinaMoldovaSierra LeoneBulgariaCentral African RepublicSouth AfricaSudanEquatorial GuineaKyrgyzstanSenegalLibyaNigerSolomon IslandsGabonMaliSomaliaZambiaBurkina FasoAfghanistanCameroonGambiaCongoNigeriaMalawiSouth SudanDjiboutiTanzaniaPapua New GuineaHaitiMadagascarYemenDemocratic Republic of CongoChadBurundiBhutanNicaraguaGuernseyDenmarkCuracaoIsraelLiechtensteinNew CaledoniaTokelauNew ZealandNorthern CyprusSaint Kitts and NevisIranUruguayTurks and Caicos IslandsGuineaUzbekistanThailandGuyanaBelizeFranceSurinameLithuaniaSloveniaGuinea-BissauTimorEstoniaIndiaLesothoComorosAngolaTogoBulgariaSenegalSomaliaCongoPapua New GuineaBurundi020k40k60k80k100k120k

Daily vaccinations per million per countryCountryDaily vaccinations per million

In [42]:

```
trace_bar(vacc_country, 'people_vaccinated', 'People vaccinated per country', 'Country','People vaccinated', "lightblue" )
```

ChinaIndiaBrazilUnited StatesIndonesiaJapanVietnamRussiaPhilippinesPakistanBangladeshMexicoSpainIranGermanyTurkeyThailandColombiaFranceUnited KingdomItalySouth KoreaEgyptEnglandArgentinaCanadaPolandPeruMalaysiaMyanmarSaudi ArabiaUzbekistanMoroccoNepalEthiopiaAustraliaVenezuelaNigeriaSouth AfricaCubaTaiwanChileSri LankaUkraineEcuadorCambodiaMozambiqueUgandaNetherlandsKenyaAngolaIraqUnited Arab EmiratesPortugalKazakhstanBelgiumRwandaGhanaGreeceSwedenGuatemalaAlgeriaDominican RepublicTunisiaCote d'IvoireBoliviaCzechiaAustriaIsraelHong KongHungarySwitzerlandBelarusLaosNicaraguaRomaniaHondurasAzerbaijanSudanTajikistanAfghanistanZimbabweSingaporeDenmarkJordanEl SalvadorFinlandScotlandTurkmenistanCosta RicaNorwayNew ZealandIrelandTanzaniaParaguayGuineaPanamaKuwaitSerbiaOmanUruguayBeninSlovakiaWalesLebanonZambiaQatarCroatiaSyriaMongoliaLibyaNigerBurkina FasoPalestineLithuaniaSomaliaBulgariaSierra LeoneGeorgiaMalawiTogoMauritaniaKyrgyzstanSenegalBotswanaNorthern IrelandCameroonLatviaMaliAlbaniaSloveniaBahrainArmeniaMadagascarLiberiaMauritiusMoldovaBosnia and HerzegovinaCentral African RepublicKosovoEstoniaDemocratic Republic of CongoNorth MacedoniaLesothoJamaicaTrinidad and TobagoTimorCyprusBhutanCongoFijiYemenMacaoSouth SudanGuinea-BissauLuxembourgMaltaGuyanaNamibiaBruneiMaldivesEswatiniCape VerdeComorosGambiaPapua New GuineaIcelandGabonMontenegroNorthern

CyprusChadSurinameEquatorial GuineaBelizeSolomon IslandsNew CaledoniaFrench PolynesiaSamoaBahamasHaitiBarbadosDjiboutiVanuatuSao Tome and PrincipeCuracaoArubaSeychellesJerseyKiribatiTongaIsle of ManChinaVietnamSpainFranceArgentinaSaudi ArabiaVenezuelaSri LankaNetherlandsKazakhstanGuatemalaCzechiaBelarusSudanJordanNorwayPanamaSlovakiaSyriaLithuaniaTogoCameroonArmeniaCentral African RepublicJamaicaFijiMaltaCape VerdeMontenegroSolomon IslandsBarbadosSeychellesCayman IslandsGibraltarSaint Kitts and NevisBonaire Sint Eustatius and SabaTuvaluPitcairn00.5B1B1.5B2B2.5B3B

People vaccinated per countryCountryPeople vaccinated

In [43]:

```
trace_bar(vacc_country, 'people_vaccinated_per_hundred', 'People vaccinated per hundred per country', 'Country','People vaccinated per hundred', "orange" )
```

GibraltarCubaUnited Arab EmiratesGuernseyChinaQatarPortugalFinlandSpainVietnamBermudaCyprusJapanIsle of ManKuwaitBrazilSan MarinoNiuePitcairnColombiaPhilippinesPanamaNepalChileBruneiSingaporeMaltaCayman IslandsBhutanArgentinaMacaoSouth KoreaCambodiaSamoaAustraliaHong KongCanadaCook IslandsUruguaySeychellesPeruFaeroe IslandsCosta RicaNauruItalyIcelandMalaysiaNew ZealandDenmarkPolandEcuadorNicaraguaArubaJerseyIrelandScotlandTaiwanAntigua and BarbudaTurks and Caicos IslandsFranceWalesCroatiaBelgiumNorwayThailandSri LankaUzbekistanWallis and FutunaMauritiusEnglandNetherlandsHondurasUnited KingdomVenezuelaSwedenUnited StatesBangladeshGreeceGermanyLaosLuxembourgFijiAustriaIranNorthern IrelandAndorraFalkland IslandsOmanNorthern CyprusSaudi ArabiaMaldivesMonacoGreenlandLithuaniaBonaire Sint Eustatius and SabaIsraelTongaLatviaRussiaSaint HelenaTurkmenistanIndonesiaTokelauIndiaBahrainEl SalvadorAnguillaLiechtensteinSwitzerlandMongoliaTurkeyBelarusRwandaBritish Virgin IslandsMoroccoHungaryFrench PolynesiaMexicoDominican RepublicCuracaoNew CaledoniaEstoniaCzechiaKiribatiSint Maarten (Dutch part)Cape VerdeSloveniaBotswanaTunisiaSaint Kitts and NevisGuyanaBoliviaKosovoBelizePakistanBarbadosAzerbaijanParaguayTrinidad and TobagoTuvaluTajikistanTimorSlovakiaSao Tome and PrincipeKazakhstanSerbiaRomaniaMyanmarMontenegroJordanSurinameDominicaAlbaniaMozambiqueEgyptGuatemalaBahamasNorth MacedoniaGeorgiaComorosGrenadaPalestineMontserratArmeniaLebanonVanuatuLesothoUkraineSouth AfricaSolomon IslandsZimbabweAngolaEswatiniMauritaniaSaint Vincent and the GrenadinesLibyaSaint LuciaUgandaBosnia and HerzegovinaGhanaBulgariaCote d'IvoireJamaicaGuinea-BissauGuineaIraqBeninMoldovaKyrgyzstanKenyaSierra LeoneEthiopiaLiberiaCentral African RepublicTogoEquatorial GuineaNamibiaAlgeriaDjiboutiGambiaZambiaGabonAfghanistanSyriaCongoSomaliaSuda nNigeriaBurkina FasoNigerSenegalMalawiTanzaniaMaliCameroonSouth SudanMadagascarPapua New GuineaYemenChadHaitiDemocratic Republic of CongoBurundiGibraltarPortugalJapanPitcairnBruneiMacaoCanadaCosta RicaDenmarkIrelandWalesUzbekistanUnited KingdomGermanyNorthern IrelandMaldivesTongaTokelauSwitzerlandMoroccoNew CaledoniaSloveniaKosovoTrinidad and TobagoKazakhstanSurinameBahamasMontserratSouth AfricaSaint Vincent and the GrenadinesBulgariaBeninLiberiaDjiboutiCongoSenegalMadagascarBurundi050100150200250300

People vaccinated per hundred per countryCountryPeople vaccinated per hundred

In [44]:

```
def plot_scatter(data, x, y, size, color, hover_name, title):

    fig = px.scatter(data, x=x, y=y, size=size, color=color, hover_name=hover_name, title=title, size_max=80)

    fig.update_layout({"legend_orientation":"h"})

    fig.update_layout(legend = dict(yanchor ="top", y = -0.2))

    fig.update_layout({"legend_title":"Vaccine scheme"})

    fig.update_layout({"plot_bgcolor":"rgba(0,0,0,0)","paper_bgcolor":"rgba(0,0,0,0)"})

    fig.update_xaxes(showline=True, linewidth=2, linecolor='black', mirror=True)

    fig.update_yaxes(showline=True, linewidth=2, linecolor='black', mirror=True)

    fig.update_xaxes(zeroline=True, zerolinewidth=1, zerolinecolor='grey')

    fig.update_yaxes(zeroline=True, zerolinewidth=1, zerolinecolor='grey')

    fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey')

    fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey')

    fig.show()
```

In [45]:

```
plot_scatter(vacc_country, x = "total_vaccinations", y = "daily_vaccinations",

            size = "total_vaccinations", color="vaccines",
```

```
            hover_name = "country", title = "Vaccinations (Total vs. Daily) grouped p
er country and vaccines")
```

01B2B3B4B050M

Vaccine schemeJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingOxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZenecaOxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sputnik VCanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTechJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingSinopharm/Beijing, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Pfizer/BioNTechCovaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacJohnson&Johnson, Oxford/AstraZenecaJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/BeijingSinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacCovaxin, Oxford/AstraZenecaCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacCanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacCovaxin, Oxford/AstraZeneca, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik VAbdala, Soberana Plus, Soberana02Johnson&Johnson, Moderna, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCovaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacJohnson&Johnson, Pfizer/BioNTechPfizer/BioNTech, Sinopharm/Beijing, Sputnik VOxford/AstraZeneca, Sputnik VModernaModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VOxford/AstraZeneca, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, ModernaJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VPfizer/BioNTech, SinovacJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VCovaxin, Oxford/AstraZeneca, Sputnik VJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCOVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VQazVac, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Novavax, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTech, Sinopharm/BeijingCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VAbdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, SinovacCanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VCovaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VEpiVacCorona, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTech, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sputnik VModerna, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VMedigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VJohnson&Johnson, Pfizer/BioNTech, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacPfizer/BioNTech, Sinovac, TurkovacEpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik VAbdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, SinovacVaccinations (Total vs. Daily) grouped per country and vaccinestotal_vaccinationsdaily_vaccinations

In [46]:

```
plot_scatter(vacc_country,x = "people_vaccinated", y = "daily_vaccinations_per_millio
n", size = "total_vaccinations", color = "vaccines", hover_name = "country",title = "
Vaccinations (daily / million vs. iso_code) grouped per country and vaccines")
```

01B2B3B4B−100k0100k

Vaccine schemeJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingOxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZenecaOxford/AstraZeneca, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sputnik VCanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTechJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingSinopharm/Beijing, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VModerna, Pfizer/BioNTechCovaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacJohnson&Johnson, Oxford/AstraZenecaJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/BeijingSinopharm/BeijingJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacCovaxin, Oxford/AstraZenecaCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacCanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, SinovacCovaxin, Oxford/AstraZeneca, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik VAbdala, Soberana Plus, Soberana02Johnson&Johnson, Moderna, Pfizer/BioNTechJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCovaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, SinovacJohnson&Johnson, Pfizer/BioNTechPfizer/BioNTech, Sinopharm/Beijing, Sputnik VOxford/AstraZeneca, Sputnik VModernaModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VOxford/AstraZeneca, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, ModernaJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VPfizer/BioNTech, SinovacJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VCovaxin, Oxford/AstraZeneca, Sputnik VJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCOVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VQazVac, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VJohnson&Johnson, Moderna, Novavax, Pfizer/BioNTechOxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTech, Sinopharm/BeijingCanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacCanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VAbdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik VOxford/AstraZeneca, Pfizer/BioNTech, SinovacCanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VCovaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VEpiVacCorona, Sputnik VJohnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik VPfizer/BioNTech, Sputnik VOxford/AstraZeneca, Sinopharm/Beijing, Sputnik VModerna, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacJohnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik VMedigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTechModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik VJohnson&Johnson, Pfizer/BioNTech, Sinopharm/BeijingModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, SinovacPfizer/BioNTech, Sinovac, TurkovacEpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik VModerna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik VAbdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik VJohnson&Johnson, Oxford/AstraZeneca, SinovacVaccinations (daily / million vs. iso_code) grouped per country and vaccinespeople_vaccinateddaily_vaccinations_per_million

In [47]:

```
trace = go.Choropleth(

            locations = vacc_country['country'],

            locationmode='country names',

            z = vacc_country['total_vaccinations'],

            text = vacc_country['country'],

            autocolorscale =False,

            reversescale = True,

            colorscale = 'viridis',

            marker = dict(

                line = dict(

                    color = 'rgb(0,0,0)',

                    width = 0.5)
```

```
            ),
            colorbar = dict(
                title = 'Total vaccinations',
                tickprefix = '')
        )

data = [trace]
layout = go.Layout(
    title = 'Total vaccinations per country',
    geo = dict(
        showframe = True,
        showlakes = False,
        showcoastlines = True,
        projection = dict(
            type = 'natural earth'
        )
    )
)

fig = dict( data=data, layout=layout )
```

00.5B1B1.5B2B2.5B3BTotal vaccinationsTotal vaccinations per country

in the total vaccination of countries;

China is the country with the most vaccines overall.

The country with the least vaccines is Chad.

In [48]:

```
trace = go.Choropleth(

            locations = vacc_country['country'],

            locationmode='country names',

            z = vacc_country['daily_vaccinations'],

            text = vacc_country['country'],

            autocolorscale =False,

            reversescale = True,

            colorscale = 'viridis',

            marker = dict(

                line = dict(

                    color = 'rgb(0,0,0)',

                    width = 0.5)

            ),

            colorbar = dict(

                title = 'Daily vaccinations',

                tickprefix = '')

        )


data = [trace]

layout = go.Layout(

    title = 'Daily vaccinations per country',

    geo = dict(

        showframe = True,
```

```
        showlakes = False,

        showcoastlines = True,

        projection = dict(

            type = 'natural earth'

        )

    )

)


fig = dict( data=data, layout=layout )
```

05M10M15M20MDaily vaccinationsDaily vaccinations per country

in the daily vaccination of countries;

China is the country with the most daily vaccinations.

The country with the least daily vaccination is Chad.

In [49]:

```
trace = go.Choropleth(

            locations = vacc_country['country'],

            locationmode='country names',

            z = vacc_country['daily_vaccinations_per_million'],

            text = vacc_country['country'],

            autocolorscale =False,

            reversescale = True,

            colorscale = 'viridis',

            marker = dict(

                line = dict(
```

```
                    color = 'rgb(0,0,0)',
                    width = 0.5)
            ),
            colorbar = dict(
                title = 'Daily vaccinations per million',
                tickprefix = '')
        )

data = [trace]
layout = go.Layout(
    title = 'Daily vaccinations per million per country',
    geo = dict(
        showframe = True,
        showlakes = False,
        showcoastlines = True,
        projection = dict(
            type = 'natural earth'
        )
    )
)

fig = dict( data=data, layout=layout )
```

20k40k60k80k100kDaily vaccinations per millionDaily vaccinations per million per country

Number of daily vaccines per million;

Bhudan is the country with the most daily vaccinations per million.

The country with the lowest daily vaccination per million is Chad.

In [50]:

```
trace = go.Choropleth(

            locations = vacc_country['country'],

            locationmode='country names',

            z = vacc_country['people_vaccinated'],

            text = vacc_country['country'],

            autocolorscale =False,

            reversescale = True,

            colorscale = 'viridis',

            marker = dict(

                line = dict(

                    color = 'rgb(0,0,0)',

                    width = 0.5)

            ),

            colorbar = dict(

                title = 'People vaccinated',

                tickprefix = '')

        )


data = [trace]

layout = go.Layout(
```

```
    title = 'People vaccinated per country',
    geo = dict(
        showframe = True,
        showlakes = False,
        showcoastlines = True,
        projection = dict(
            type = 'natural earth'
        )
    )
)

fig = dict( data=data, layout=layout )
```

00.5B1B1.5B2B2.5B3BPeople vaccinatedPeople vaccinated per country

people vaccinated in countries;

China is the country with the most people vaccinated.

Chad is the country with the least number of people vaccinated.

In [51]:

linkcode

```
trace = go.Choropleth(
            locations = vacc_country['country'],
            locationmode='country names',
            z = vacc_country['people_vaccinated_per_hundred'],
            text = vacc_country['country'],
            autocolorscale =False,
```

```
            reversescale = True,
            colorscale = 'viridis',
            marker = dict(
                line = dict(
                    color = 'rgb(0,0,0)',
                    width = 0.5)
            ),
            colorbar = dict(
                title = 'People vaccinated per hundred',
                tickprefix = '')
        )

data = [trace]
layout = go.Layout(
    title = "People vaccinated per hundred per country",
    geo = dict(
        showframe = True,
        showlakes = False,
        showcoastlines = True,
        projection = dict(
            type = 'natural earth'
        )
    )
)
```

```
fig = dict( data=data, layout=layout )
```

50100150200250300People vaccinated per hundredPeople vaccinated per hundred per country

percentage of people vaccinated in countries;

China has the highest percentage of people vaccinated.

democratic republic of congo has the lowest percentage of people vaccinated.

COVID-19 World Vaccination Progress

Basic Visualization

1. **Vaccination by Country**
   1.1 Total Vaccinations
   1.2 People Vaccinated
   1.3 People Fully Vaccinated
2. **Vaccination by Country per Hundred**
   2.1 Total Vaccinations
   2.2 People Vaccinated
   2.3 People Fully Vaccinated
3. **Daily Vaccinations**
   3.1 Daily Vaccinations by Country
   3.2 Daily Vaccinations by Country per Million

Advanced Visualization

1. **Total Vaccination & 30-day Rolling**
2. **Daily Vaccination**
   2.1 Day of Week
   2.2 Month
3. **Total Vaccination Status Across Countries**

Import packages

In [1]:

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline
```

Pandas default settings

In [2]:

```
# pd.set_option('display.max_rows', 500)

pd.set_option('display.max_columns', 30)

pd.set_option('display.float_format', '{:,.2f}'.format)
```

In [3]:

```
#Load Dataset



df_vaccination = pd.read_csv('../input/covid-world-vaccination-progress/country_vacci
nations.csv')



#data is from : https://gpreda/covid-world-vaccination-progress
```

Exploring the dataset

In [4]:

```
#Display first 5 rows


df_vaccination.head()
```

Out[4]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million	vaccines	source_name	source_website
0	Afghanistan	AFG	2021-02-22	0.00	0.00	NaN	NaN	NaN	0.00	0.00	NaN	NaN	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson &Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

In [5]:

```
df_vaccination.info()
```

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

RangeIndex: 77709 entries, 0 to 77708

Data columns (total 15 columns):

 #   Column                               Non-Null Count  Dtype

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

 0   country                              77709 non-null  object

 1   iso_code                             77709 non-null  object

 2   date                                 77709 non-null  object

 3   total_vaccinations                   40048 non-null  float64

 4   people_vaccinated                    37945 non-null  float64

 5   people_fully_vaccinated              35465 non-null  float64

 6   daily_vaccinations_raw               32591 non-null  float64

 7   daily_vaccinations                   77429 non-null  float64

 8   total_vaccinations_per_hundred       40048 non-null  float64

 9   people_vaccinated_per_hundred        37945 non-null  float64

 10  people_fully_vaccinated_per_hundred  35465 non-null  float64

 11  daily_vaccinations_per_million       77429 non-null  float64

 12  vaccines                             77709 non-null  object

 13  source_name                          77709 non-null  object

 14  source_website                       77709 non-null  object

dtypes: float64(9), object(6)

memory usage: 8.9+ MB
```

Content

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;
- **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;

In [6]:

```
#Find the number or rows and columns



df_vaccination.shape



#There are 76095 rows and 15 columns
```

Out[6]:

```
(77709, 15)
```

In [7]:

```
df_vaccination.isnull().sum()


#There are no empty rows for country, iso_code or date columns.
```

Out[7]:

```
country                                    0
iso_code                                   0
date                                       0
total_vaccinations                     37661
people_vaccinated                      39764
people_fully_vaccinated                42244
daily_vaccinations_raw                 45118
daily_vaccinations                       280
total_vaccinations_per_hundred         37661
people_vaccinated_per_hundred          39764
people_fully_vaccinated_per_hundred    42244
daily_vaccinations_per_million           280
vaccines                                   0
source_name                                0
source_website                             0
dtype: int64
```

In [8]:

```
# General Overview of the calculations in data



df_vaccination.describe()
```

Out[8]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million
count	40,048.00	37,945.00	35,465.00	32,591.00	77,429.00	40,048.00	37,945.00	35,465.00	77,429.00
mean	40,384,563.30	15,903,249.33	12,278,486.43	276,490.71	135,957.05	73.11	38.60	32.81	3,417.77
std	202,533,898.56	63,330,416.02	49,156,651.29	1,245,210.53	797,868.94	63.46	28.74	27.62	4,028.63
min	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	472,141.25	314,538.00	211,664.00	5,240.00	972.00	13.31	9.72	5.58	704.00
50%	3,141,270.00	1,939,970.00	1,442,791.00	26,278.00	7,869.00	59.37	37.40	27.31	2,253.00

75%	15,091,858.25	8,102,781.00	6,399,728.00	129,159.00	45,644.00	123.68	65.33	58.32	4,933.00
max	3,063,391,000.00	1,266,426,000.00	1,228,340,000.00	24,741,000.00	22,424,286.00	333.76	123.75	121.14	117,497.00

Data Preparation

In [9]:

```
#drop the source_name,source_website and vaccine columns



df_vaccine_country = df_vaccination.drop(['source_name','source_website','vaccines'], axis=1)

df_vaccine_country.head()
```

Out[9]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million
0	Afghanistan	AFG	2021-02-22	0.00	0.00	NaN	NaN	NaN	0.00	0.00	NaN	NaN
1	Afghan	AF	20	NaN	NaN	NaN	NaN	1,367.	NaN	NaN	NaN	34.00

	istan	G	21-02-23					00				
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00

In [10]:

```
# convert Date column to date type and fill na values with 0 for calculation
```

```
df_vaccine_country["date"] = pd.to_datetime(df_vaccine_country["date"], format = '%Y-%m-%d')
```

```
df_vaccine_country = df_vaccine_country.replace([np.inf, -np.inf], np.nan)

df_vaccine_country = df_vaccine_country.fillna(0)

df_vaccine_country.isnull().sum()
```

Out[10]:

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
people_fully_vaccinated                0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
daily_vaccinations_per_million         0
dtype: int64
```

In [11]:

```
#Function to find total, avergae, maximum  and minimum of different vaccinations status by country
```

```python
def vaccination_country(col_name,func_name):

    '''
    Function that requires vaccination column name, and sum/mean/max/min function nam
e as string arguments.
    '''
    if func_name == 'sum':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .sum()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                            )

    elif func_name == 'mean':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .mean()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                                )
    elif func_name == 'max':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
```

```
                                    .max()

                                    .sort_values(by=col_name,ascending= False)

                                    .reset_index()

                                   )

    elif func_name == 'min':


        return  (df_vaccine_country[['country',col_name]].groupby(by='country')

                                    .min()

                                    .sort_values(by=col_name,ascending= False)

                                    .reset_index()

                                   )
```

In [12]:

```
# Calculating different vaccinations for visualizations

max_total_vaccinations = vaccination_country('total_vaccinations','max')



sum_people_vaccinated = vaccination_country('people_vaccinated','sum')

sum_people_fully_vaccinated = vaccination_country('people_fully_vaccinated','sum')



avg_total_vaccinations = vaccination_country('total_vaccinations_per_hundred','mean')

avg_people_vaccinated = vaccination_country('people_vaccinated_per_hundred','mean')

avg_people_fully_vaccinated = vaccination_country('people_fully_vaccinated_per_hundre
d','mean')

avg_daily_vaccinations = vaccination_country('daily_vaccinations_per_million','mean')
```

In [13]:

```
#Function for Country with maximum and minimum daily vaccinations

def daily_vaccination_country(col_name,func_name):



    '''

    A function that requires daily_vaccination column and max/min function name as st
ring arguments.

    '''



    daily_vaccination = (df_vaccine_country

                                 .pivot_table(index='country',columns='date',values=c
ol_name)

                                    )

    if func_name == 'max':



        daily_vaccination['Highest Daily Vaccination'] = daily_vaccination.max(axis=1)

        daily_vaccination['Date - Highest Daily Vaccination'] = daily_vaccination.idx
max(axis=1)

        daily_vaccination.sort_values(by='Highest Daily Vaccination',ascending=False,
inplace=True)

        daily_vaccination.rename_axis('',axis=1,inplace=True)



        return daily_vaccination[['Highest Daily Vaccination','Date - Highest Daily V
accination']].reset_index()



    elif func_name == 'min':
```

```
        daily_vaccination.replace(0.00,np.nan,inplace=True)

        daily_vaccination['Lowest Daily Vaccination'] = daily_vaccination.min(axis=1)

        daily_vaccination['Date - Lowest Daily Vaccination'] = daily_vaccination.idxm
in(axis=1)

        daily_vaccination.sort_values(by='Lowest Daily Vaccination',ascending=False,i
nplace=True)

        daily_vaccination.rename_axis('',axis=1,inplace=True)



        return daily_vaccination[['Lowest Daily Vaccination','Date - Lowest Daily Vac
cination']].reset_index()
```

In [14]:

```
#Calculating highest and lowest daily vaccination and the respective dates.

highest_daily_vaccination = daily_vaccination_country('daily_vaccinations','max')

lowest_daily_vaccination = daily_vaccination_country('daily_vaccinations','min')
```

Data Visualization

1.1 Top & Bottom 5 Countries in terms of Total Vaccination

In [15]:

```
#Set sns theme and default figsize for all the sns visualizations.

sns.set_theme(style='whitegrid')

sns.set(rc={'figure.figsize' : (12,5)})



fig, axes  = plt.subplots(2,1)



sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.head(),ax=
axes[0])
```

```
axes[0].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Top 5 Countries in te
rms of total vaccinations!')



sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.tail(),ax=
axes[1])

axes[1].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Bottom 5 Countries in
 terms of total vaccinations!')



fig.tight_layout()

plt.show()
```

1.2 Top & Bottom 5 Countries in terms of People Vaccinated

In [16]:

```
fig, axes = plt.subplots(2,1)



sns.barplot(x='country',y='people_vaccinated',data=sum_people_vaccinated.head(),ax=ax
es[0])

axes[0].set(xlabel = '', ylabel = 'People Vaccinated', title ='Top 5 Countries in ter
ms of people vaccinated!')



sns.barplot(x='country', y='people_vaccinated',data=sum_people_vaccinated.tail(),ax=a
xes[1])

axes[1].set(xlabel = '', ylabel = 'People Vaccinated', title ='Bottom 5 Countries in
terms of people vaccinated!')
```

```
fig.tight_layout()

plt.show()
```

1.3 Top & Bottom 5 Countries in terms of People Fully Vaccinated

In [17]:

```
fig, axes = plt.subplots(2,1)



sns.barplot(x='country',y='people_fully_vaccinated',data=sum_people_fully_vaccinated.
head(),ax=axes[0])

axes[0].set(xlabel = '', ylabel = 'People Fully Vaccinated', title ='Top 5 Countries
in terms of people fully vaccinated!')



sns.barplot(x='country',y='people_fully_vaccinated',data=sum_people_fully_vaccinated.
tail(),ax=axes[1])

axes[1].set(xlabel = '', ylabel = 'People Fully Vaccinated', title ='Bottom 5 Countri
es in terms of people fully vaccinated!')




# plt.ticklabel_format(style='plain', axis='y') #Uncomment if y label needs to displa
y accurate values



fig.tight_layout()

plt.show()
```

2.1 Top & Bottom 5 Countries in terms of Total Vaccinations per Hundred

In [18]:

```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='total_vaccinations_per_hundred',data=avg_total_vaccinations.head(),ax=axes[0])

axes[0].set(xlabel='', ylabel='Average Vaccinations per 100', title='Top 5 Countries in terms of average vaccinations per hundred!')

sns.barplot(x='country', y='total_vaccinations_per_hundred',data=avg_total_vaccinations.tail(),ax=axes[1])

axes[1].set(xlabel='', ylabel='Average Vaccinations per 100', title='Bottom 5 Countries in terms of average vaccinations per hundred!')

fig.tight_layout(h_pad=3)

plt.show()
```

2.2 Top & Bottom 5 Countries in terms of People Vaccinated per Hundred

In [19]:

```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='people_vaccinated_per_hundred',data=avg_people_vaccinated.head(),ax=axes[0])

axes[0].set(xlabel='', ylabel='People Vaccinated per 100', title='Top 5 Countries in terms of average people vaccinated per hundred!')

sns.barplot(x='country', y='people_vaccinated_per_hundred',data=avg_people_vaccinated.tail(),ax=axes[1])

axes[1].set(xlabel='', ylabel='People Vaccinated per 100', title='Bottom 5 Countries in terms of average people vaccinated per hundred!')
```

```
fig.tight_layout()

plt.show()
```

2.3 Top & Bottom 5 Countries in terms of People Fully Vaccinated per Hundred

In [20]:

```
fig, axes = plt.subplots(2,1)



sns.barplot(x='country', y='people_fully_vaccinated_per_hundred',data=avg_people_full
y_vaccinated.head(),ax=axes[0])

axes[0].set(xlabel='', ylabel='People Fully Vaccinated per 100', title='Top 5 Countri
es in terms of average people fully vaccinated per hundred!')



sns.barplot(x='country', y='people_fully_vaccinated_per_hundred',data=avg_people_full
y_vaccinated.tail(),ax=axes[1])

axes[1].set(xlabel='', ylabel='People Fully Vaccinated per 100', title='Bottom 5 Coun
tries in terms of average people fully vaccinated per hundred!')



fig.tight_layout(h_pad=3)

plt.show()
```

3.1 Highest & Lowest 5 Daily Vaccination by Country

unfold_moreShow hidden code

In [22]:

```
fig, axes = plt.subplots(1,2)
```

```python
sns.barplot(data=daily_top5_highest,x="country", y="Highest Daily Vaccination",ax=axes[0],hue='Date - Highest Daily Vaccination')

axes[0].set(xlabel='',ylabel='Daily Vaccination',title='Highest Daily Vaccination by Country')



sns.barplot(data=daily_top5_lowest,x="country", y="Lowest Daily Vaccination",ax=axes[1],hue='Date - Lowest Daily Vaccination')

axes[1].set(xlabel='',ylabel='Daily Vaccination',title='Lowest Daily Vaccination by Country')




# plt.ticklabel_format(style='plain', axis='y')

fig.tight_layout()

plt.show()
```

3.2 Top & Bottom 5 Daily Vaccination by Country per Million

In [23]:

```python
fig, axes = plt.subplots(2,1)



sns.barplot(x='country', y='daily_vaccinations_per_million',data=avg_daily_vaccinations.head(),ax=axes[0])

axes[0].set(xlabel='', ylabel='Daily Vaccinations per Million', title='Top 5 Countries in terms of daily vaccinations per million!')



sns.barplot(x='country', y='daily_vaccinations_per_million',data=avg_daily_vaccinations.tail(),ax=axes[1])
```

```
axes[1].set(xlabel='', ylabel='Daily Vaccinations per Million', title='Bottom 5 Countries in terms of daily vaccinations per million!')


fig.tight_layout(h_pad=3)

plt.show()
```

Advanced Data Visualization

Import Plotly Library

In [24]:

```
from plotly.offline import init_notebook_mode

import plotly.express as px

init_notebook_mode(connected=True)
```

1. Total Vaccination & 30-day Rolling by Top 5 Country

In [25]:

```
#Top 5 country with highest total vaccinations

list(max_total_vaccinations['country'].head())
```

Out[25]:

```
['China', 'India', 'United States', 'Brazil', 'Indonesia']
```

In [26]:

```
# Filter the top 5 countries and find their 30 day rolling average of total_vaccinations

top5_country_total = ['China', 'India', 'United States', 'Brazil', 'Indonesia']

top5_country_total_day = df_vaccine_country[df_vaccine_country['country'].isin(top5_country_total)].copy()
```

```
top5_country_total_day['30 - Day Rolling'] = top5_country_total_day['total_vaccinatio
ns'].rolling(window=30).mean()
```

In [27]:

```
fig = px.line(top5_country_total_day,x="date",y="total_vaccinations",color='country',

                 labels={"country" : 'Top 5 Country', 'date' : 'Date', 'total_vaccina
tions' : "Total Vaccinations"},

                title="Total Vaccination Progress - Top 5 Country",template='plotly_d
ark')



for country in top5_country_total_day['country'].unique():

    fig.add_scatter(x=top5_country_total_day[top5_country_total_day['country'] == cou
ntry]['date']

                    ,y=top5_country_total_day[top5_country_total_day['country'] == co
untry]['30 - Day Rolling']

                    ,mode="lines",name='30 Day Rolling Vaccination ' + country)
```

2.1 Daily Vaccination by Day of Week by Top 5 Country

In [28]:

```
# Get the name of the day of vaccinations

top5_country_total_day['Day of Week'] = top5_country_total_day['date'].apply(lambda x:
x.day_name())

top5_country_total_day['Day of Week'].unique()
```

Out[28]:

```
array(['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',

       'Saturday'], dtype=object)
```

In [29]:

```
fig = px.box(top5_country_total_day,x='Day of Week',y='daily_vaccinations',color='cou
ntry',

             labels={"country" : 'Top 5 Country', 'daily_vaccinations' : "Daily Vacci
nation"},

                title="Dailly Vaccination by Day of Week - Top 5 Country",template='p
lotly_dark')
```

2.2 Daily Vaccination by Month by Top 5 Country

In [30]:

```
# Get the name of the month of vaccinations

top5_country_total_day['Month'] = top5_country_total_day['date'].apply(lambda x:x.mon
th_name())

top5_country_total_day['Month'].unique()
```

Out[30]:

```
array(['January', 'February', 'March', 'April', 'May', 'June', 'July',

       'August', 'September', 'October', 'November', 'December'],

      dtype=object)
```

In [31]:

linkcode

```
fig = px.bar(top5_country_total_day,x='Month',y='daily_vaccinations',color='country',

             labels={"country" : 'Top 5 Country', 'daily_vaccinations' : "Daily Vacci
nation"},

                title="Dailly Vaccination by Month - Top 5 Country",template='plotly_
dark')
```

3. Total Vaccination Status Across Countries

In [32]:

```
fig = px.choropleth(max_total_vaccinations,locations='country',locationmode='country
names',
```

```
                   color='total_vaccinations',hover_name="country", template='plotly_
dark',
                   title= 'Total Vaccination Status Across Countries',projection='nat
ural earth',
                   labels={'country' : 'Country','total_vaccinations' : 'Total Vaccin
ations'})
```