Naan Mudhalvan

IBM project

Applied DataScience(Phase 3- Development)

Topic- covid 19 Vaccine Analysis

By: Sushthi. R(au411521104115)

3.1 Dataset and its detail explanation implementation

3.1.1 Basic Libraries:

This dataset contains 35310 rows and 15 columns which is really informaive to analysis. In this project,an attempt has been made to analyze various information of COVID-19 World Vaccination Progress such as country, total_Vaccinations, people_vaccinated, daily_vaccinations total_vaccinations_per_hundred, people_vaccinated_per_hundred, people_fully_vaccinated_per_hundred, vaccines and many more.

Library Used:-

- **Pandas-** Pandas is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data

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

- **Seaborn**-Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics

3.1.2 Data Preparation and Cleaning:

- Load the dataset into a data frame using Pandas
- Explore the number of rows & columns, ranges of values etc.
- Handle missing, incorrect and invalid data

```
import pandas as pd
vaccinations_df = pd.read_csv('../input/covid-world-vaccination-progress/country_vaccinations.csv')
vaccinations_df
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows × 15 columns

vaccinations_df.info()

```
<code                                  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_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
```

vaccinations_df.columns

```
Index(['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'],
      dtype='object')
```

vaccinations_df.shape

```
(86512, 15)
```

vaccinations_df.describe()

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinat
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313055e+05	80.188543	
std	2.246004e+08	7.078731e+07	5.713920e+07	1.212427e+06	7.682388e+05	67.913577	
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	
25%	5.264100e+05	3.494642e+05	2.439622e+05	4.668000e+03	9.000000e+02	16.050000	
50%	3.590096e+06	2.187310e+06	1.722140e+06	2.530900e+04	7.343000e+03	67.520000	
75%	1.701230e+07	9.152520e+06	7.559870e+06	1.234925e+05	4.409800e+04	132.735000	
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	

vaccinations_df.isnull().sum()

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                 42905
people_vaccinated                  45218
people_fully_vaccinated            47710
daily_vaccinations_raw             51150
daily_vaccinations                   299
total_vaccinations_per_hundred     42905
people_vaccinated_per_hundred      45218
people_fully_vaccinated_per_hundred    47710
daily_vaccinations_per_million       299
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

vaccinations_df.fillna(value=0, inplace=True)

	0	1	2
0	2021	02	22
1	2021	02	23
2	2021	02	24
3	2021	02	25
4	2021	02	26
...	...	...	...
86507	2022	03	25
86508	2022	03	26
86509	2022	03	27
86510	2022	03	28
86511	2022	03	29

86512 rows × 3 columns

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

vaccinations_df['month'] = date[1]

vaccinations_df['day'] = date[2]

vaccinations_df.year = pd.to_numeric(vaccinations_df.year)

vaccinations_df.month = pd.to_numeric(vaccinations_df.month)

vaccinations_df.day = pd.to_numeric(vaccinations_df.day)

vaccinations_df.date = pd.to_datetime(vaccinations_df.date)

vaccinations_df.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per
0	Afghanistan	AFG	2021-02-22	0.0	0.0	0.0	0.0	0.0	
1	Afghanistan	AFG	2021-02-23	0.0	0.0	0.0	0.0	1367.0	
2	Afghanistan	AFG	2021-02-24	0.0	0.0	0.0	0.0	1367.0	
3	Afghanistan	AFG	2021-02-25	0.0	0.0	0.0	0.0	1367.0	
4	Afghanistan	AFG	2021-02-26	0.0	0.0	0.0	0.0	1367.0	

```
vaccinations_df.info()
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 18 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   country                               86512 non-null  object
 1   iso_code                              86512 non-null  object
 2   date                                  86512 non-null  datetime64[ns]
 3   total_vaccinations                    86512 non-null  float64
 4   people_vaccinated                     86512 non-null  float64
 5   people_fully_vaccinated               86512 non-null  float64
 6   daily_vaccinations_raw                86512 non-null  float64
 7   daily_vaccinations                    86512 non-null  float64
 8   total_vaccinations_per_hundred        86512 non-null  float64
 9   people_vaccinated_per_hundred         86512 non-null  float64
 10  people_fully_vaccinated_per_hundred   86512 non-null  float64
 11  daily_vaccinations_per_million        86512 non-null  float64
 12  vaccines                              86512 non-null  object
 13  source_name                           86512 non-null  object
 14  source_website                        86512 non-null  object
 15  year                                  86512 non-null  int64
 16  month                                 86512 non-null  int64
 17  day                                   86512 non-null  int64
dtypes: datetime64[ns](1), float64(9), int64(3), object(5)
memory usage: 11.9+ MB
```

Exploratory Analysis and Visualization:

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
# Explore the mean, min, max
```

```
vaccinations_df.mean()
```

```
total_vaccinations                    2.315117e+07
people_vaccinated                     8.451007e+06
people_fully_vaccinated               6.341251e+06
daily_vaccinations_raw                1.106083e+05
daily_vaccinations                    1.308517e+05
total_vaccinations_per_hundred        4.041962e+01
```

```
people_vaccinated_per_hundred          1.953547e+01
people_fully_vaccinated_per_hundred    1.593274e+01
daily_vaccinations_per_million         3.245792e+03
year                                   2.021199e+03
month                                  6.165711e+00
day                                    1.571936e+01
dtype: float64
```

vaccinations_df.min()

```
country
Afghanistan
iso_code
ABW
date12-02 00:00:00
total_vaccinations
0.0
people_vaccinated
0.0
people_fully_vaccinated
0.0
daily_vaccinations_raw
0.0
daily_vaccinations
0.0
total_vaccinations_per_hundred
0.0
people_vaccinated_per_hundred
0.0
people_fully_vaccinated_per_hundred
0.0
daily_vaccinations_per_million
0.0
vaccines                               Abdala, Johnson&Johnson, Oxford/Ast
raZeneca, P...
source_name                            Africa Centres for Disease Control
and Prevention
source_website                         http://103.247.238.92/webportal/pag
es/covid19-...
year
2020
month
1
day
1
dtype: object
```

vaccinations_df.max()

```
country
Zimbabwe
iso_code
ZWE
date2022-03-29 00:00:00
total_vaccinations
3263129000.0
people_vaccinated
1275541000.0
people_fully_vaccinated
1240777000.0
daily_vaccinations_raw
24741000.0
daily_vaccinations
22424286.0
total_vaccinations_per_hundred
345.37
people_vaccinated_per_hundred
124.76
people_fully_vaccinated_per_hundred
122.37
daily_vaccinations_per_million
117497.0
vaccines                                                  Sinopharm/Beij
ing, Sputnik V
source_name                                                  World Healt
h Organization
source_website                          https://www.ssm.gov.mo/docs/19164/1
9164_dd2dfe...
year
2022
month
12
day
31
dtype: object
```

Explore the country Coloumn

vaccinations_df.country.value_counts()

```
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: 223, dtype: int64
```

vaccinations_df.country

```
0         Afghanistan
1         Afghanistan
2         Afghanistan
3         Afghanistan
4         Afghanistan
             ...
86507        Zimbabwe
86508        Zimbabwe
86509        Zimbabwe
86510        Zimbabwe
86511        Zimbabwe
Name: country, Length: 86512, dtype: object
```

vaccinations_df.country.nunique()

223

Explore the min and max of fully vacnated people.

vaccinations_df.people_fully_vaccinated.min()

```
0.0
```

vaccinations_df.people_fully_vaccinated.max()

```
1240777000.0
```

#Explore the min and max date.

vaccinations_df.date.min()

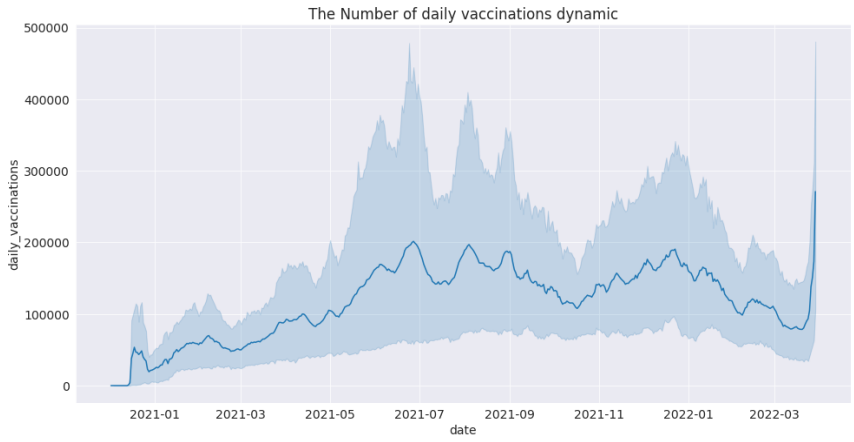
```
Timestamp('2020-12-02 00:00:00')
```

vaccinations_df.date.max()

```
Timestamp('2022-03-29 00:00:00')
```

Explore The Number of daily vaccinations dynamic

```
plt.figure(figsize=(16,8))
sns.lineplot(x=vaccinations_df.date, y=vaccinations_df.daily_vaccinations)
plt.title('The Number of daily vaccinations dynamic')
plt.show()
```



Explore the Vaccination procedure go on rapidly from which date.

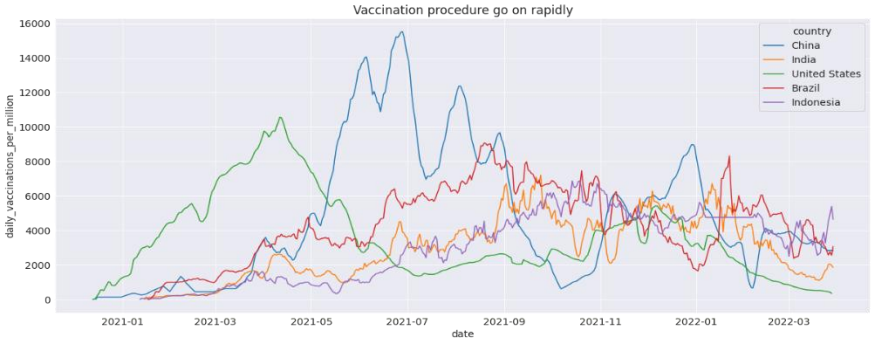
countries = vaccinations_df.groupby('country')['total_vaccinations'].max().sort_values(ascending= False)[:5].indextop_countries = pd.DataFrame(columns= vaccinations_df.columns)

for country in countries: top_countries = top_countries.append(vaccinations_df.loc[vaccinations_df['country'] == country])

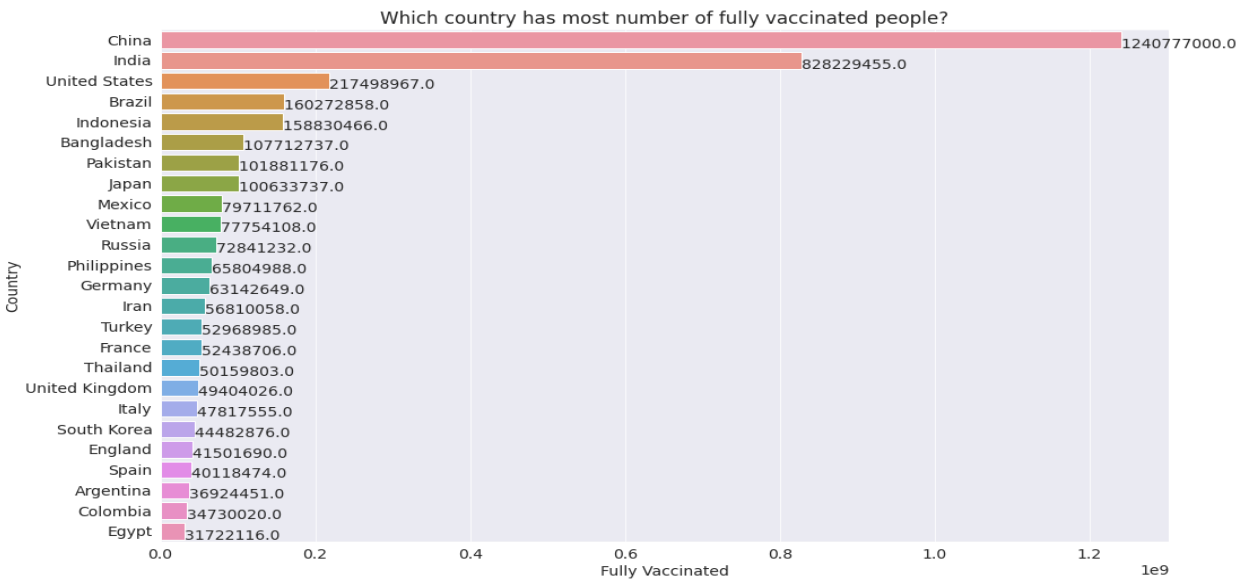
plt.figure(figsize=(20,8))

```
sns.lineplot(top_countries['date'], top_countries['daily_vaccinations_per_million'], hue= top_countries['country'], ci= False)
plt.title('Vaccination procedure go on rapidly');
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation. FutureWarning
```




```
plt.figure(figsize=(16,10))

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

plt.xlabel("Fully Vaccinated")

plt.ylabel("Country");

plt.title('Which country has most number of fully vaccinated people?');

for patch in ax.patches:

    width = patch.get_width()

    height = patch.get_height()

    x = patch.get_x()

    y = patch.get_y()

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

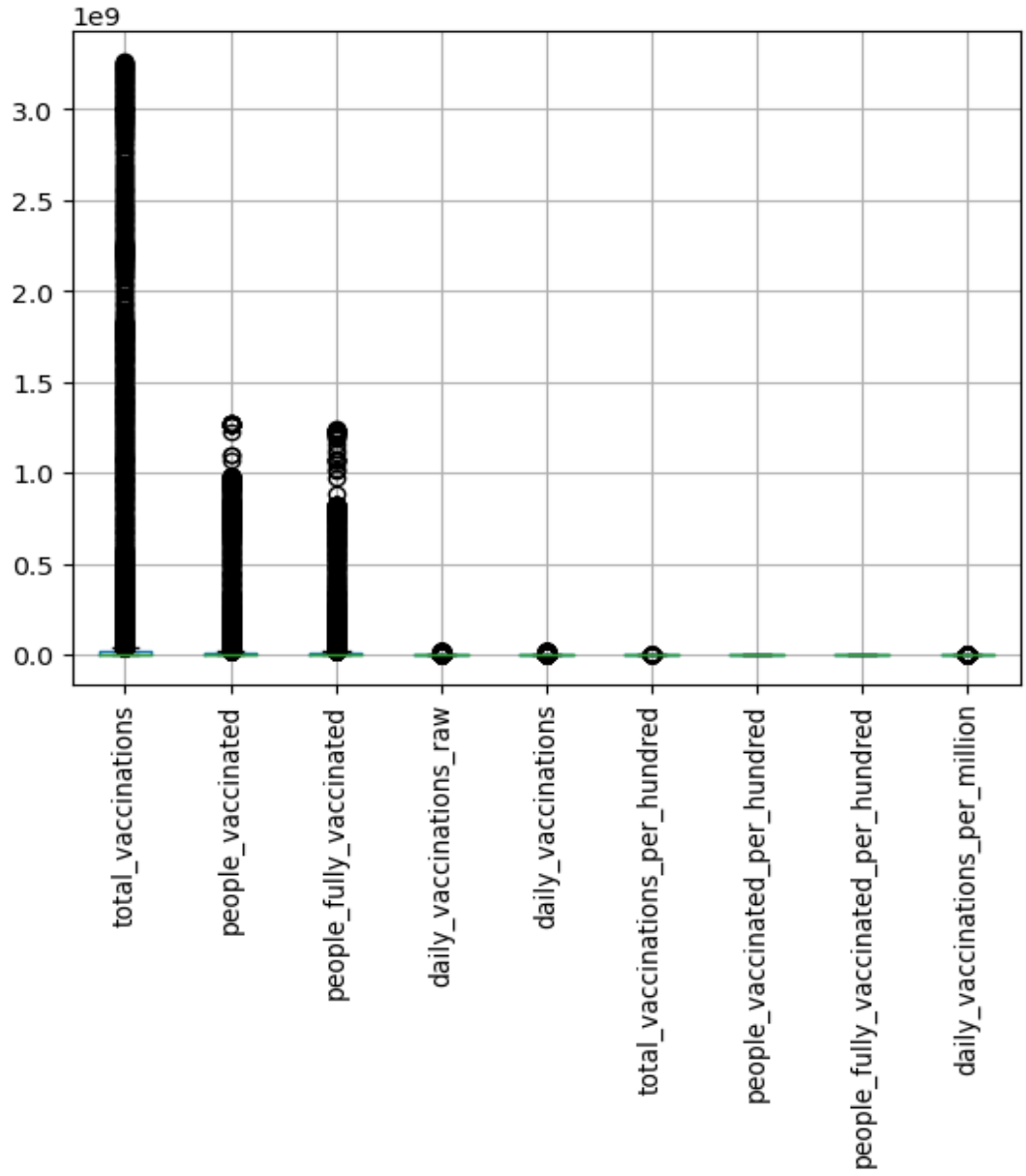


df.boxplot()

plt.xticks(rotation=90)

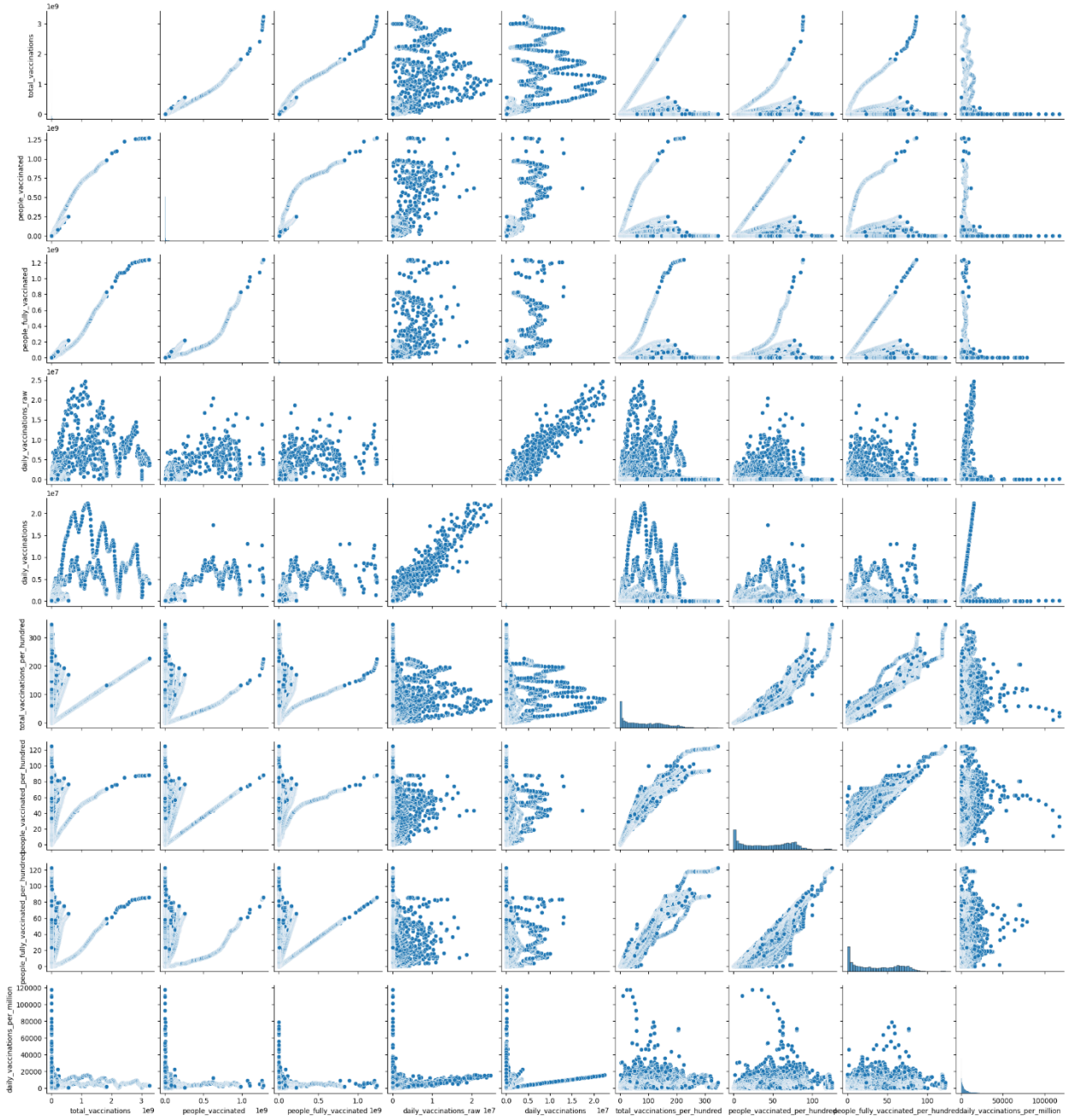
```
(array([1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(1, 0, 'total_vaccinations'),
  Text(2, 0, 'people_vaccinated'),
  Text(3, 0, 'people_fully_vaccinated'),
  Text(4, 0, 'daily_vaccinations_raw'),
  Text(5, 0, 'daily_vaccinations'),
  Text(6, 0, 'total_vaccinations_per_hundred'),
  Text(7, 0, 'people_vaccinated_per_hundred'),
  Text(8, 0, 'people_fully_vaccinated_per_hundred'),
  Text(9, 0, 'daily_vaccinations_per_million')]) (array([1, 2, 3, 4, 5, 6, 7,
8, 9]),
 [Text(1, 0, 'total_vaccinations'),
  Text(2, 0, 'people_vaccinated'),
  Text(3, 0, 'people_fully_vaccinated'),
  Text(4, 0, 'daily_vaccinations_raw'),
  Text(5, 0, 'daily_vaccinations'),
  Text(6, 0, 'total_vaccinations_per_hundred'),
  Text(7, 0, 'people_vaccinated_per_hundred'),
  Text(8, 0, 'people_fully_vaccinated_per_hundred'),
  Text(9, 0, 'daily_vaccinations_per_million')])
```

In [22]:

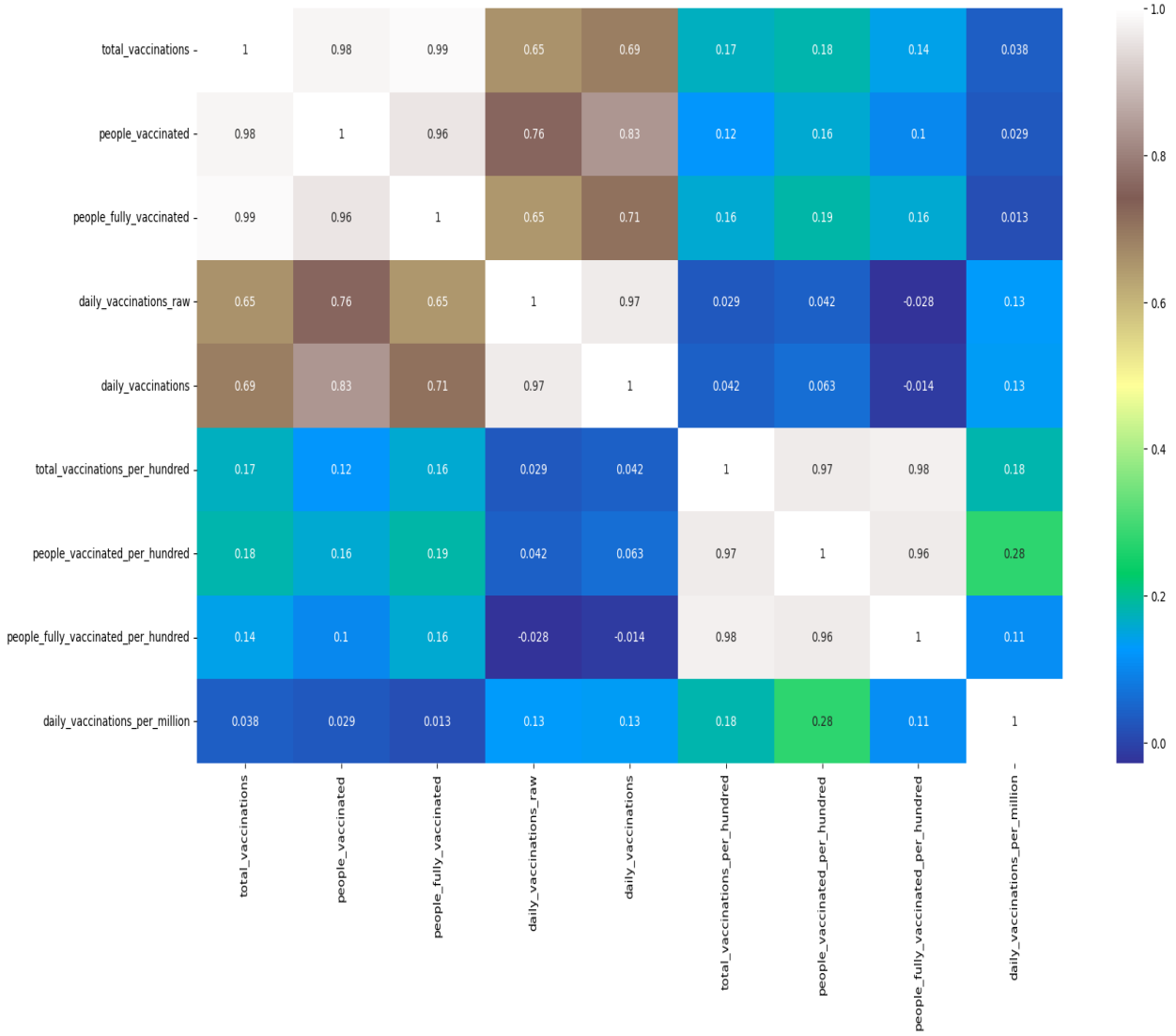


sns.pairplot(data=df)

```
<seaborn.axisgrid.PairGrid at 0x239b98d5700>
```




```
  import matplotlib.pyplot as plt
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True,cmap='terrain')
```



Conclusion:

Here is the analysis of the covid-19 vaccinations data. In future we work more analysis on this data.

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