

# Data Analysis on Covid-19 country vaccination(2020-2021)

Importing Libraries

```
In [73]: import pandas as pd
import numpy as np
```

Reading the data

```
In [4]: df=pd.read_csv('C:/Users/SUBINDAS/Desktop/subin/country_vaccinations.csv')
```

To View all data

```
In [5]: df
```

```
Out[5]:
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_v
<b>0</b>	Afghanistan	AFG	22-02-21	0.0	0.0	NaN	NaN
<b>1</b>	Afghanistan	AFG	23-02-21	NaN	NaN	NaN	NaN
<b>2</b>	Afghanistan	AFG	24-02-21	NaN	NaN	NaN	NaN
<b>3</b>	Afghanistan	AFG	25-02-21	NaN	NaN	NaN	NaN
<b>4</b>	Afghanistan	AFG	26-02-21	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...
<b>7892</b>	Zimbabwe	ZWE	19-03-21	41687.0	41687.0	41687.0	NaN
<b>7893</b>	Zimbabwe	ZWE	20-03-21	42210.0	42210.0	42210.0	NaN
<b>7894</b>	Zimbabwe	ZWE	21-03-21	42729.0	42729.0	42729.0	NaN
<b>7895</b>	Zimbabwe	ZWE	22-03-21	43294.0	43294.0	43294.0	NaN
<b>7896</b>	Zimbabwe	ZWE	23-03-21	44135.0	44135.0	44135.0	NaN

7897 rows × 15 columns

showing first 3 rows with head()

In [7]: `df.head(3)`

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vacci
0	Afghanistan	AFG	22-02-21	0.0	0.0	0.0	NaN
1	Afghanistan	AFG	23-02-21	NaN	NaN	NaN	NaN
2	Afghanistan	AFG	24-02-21	NaN	NaN	NaN	NaN

view bottom 5 records

In [9]: `df.tail(5)`

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_va
7892	Zimbabwe	ZWE	19-03-21	41687.0	41687.0	41687.0	NaN
7893	Zimbabwe	ZWE	20-03-21	42210.0	42210.0	42210.0	NaN
7894	Zimbabwe	ZWE	21-03-21	42729.0	42729.0	42729.0	NaN
7895	Zimbabwe	ZWE	22-03-21	43294.0	43294.0	43294.0	NaN
7896	Zimbabwe	ZWE	23-03-21	44135.0	44135.0	44135.0	NaN

In [10]: `# describe columns`

In [11]: `df.columns`

```
Out[11]: 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')
```

Basic information about the dataset

In [13]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7897 entries, 0 to 7896
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   country          7897 non-null    object  
 1   iso_code          7897 non-null    object  
 2   date              7897 non-null    object  
 3   total_vaccinations 4889 non-null    float64 
 4   people_vaccinated 4338 non-null    float64 
 5   people_fully_vaccinated 2961 non-null    float64 
 6   daily_vaccinations_raw 4112 non-null    float64 
 7   daily_vaccinations 7709 non-null    float64 
 8   total_vaccinations_per_hundred 4889 non-null    float64 
 9   people_vaccinated_per_hundred 4338 non-null    float64 
 10  people_fully_vaccinated_per_hundred 2961 non-null    float64 
 11  daily_vaccinations_per_million 7709 non-null    float64 
 12  vaccines          7897 non-null    object  
 13  source_name        7897 non-null    object  
 14  source_website     7897 non-null    object  
dtypes: float64(9), object(6)
memory usage: 740.4+ KB
```

To check the number of rows and columns in dataset

In [15]: df.shape

Out[15]: (7897, 15)

To check missing values count

In [17]: df.isnull().sum().values.sum()

Out[17]: 27167

getting which column contains null

In [19]: df.isnull().any()

```
Out[19]: country          False
iso_code          False
date              False
total_vaccinations  True
people_vaccinated  True
people_fully_vaccinated  True
daily_vaccinations_raw  True
daily_vaccinations  True
total_vaccinations_per_hundred  True
people_vaccinated_per_hundred  True
people_fully_vaccinated_per_hundred  True
daily_vaccinations_per_million  True
vaccines          False
source_name        False
source_website     False
dtype: bool
```

To check for the null value count

In [22]: df.isna().sum()

```
Out[22]: country          0
iso_code          0
date              0
total_vaccinations 3008
```

```
people_vaccinated           3559
people_fully_vaccinated     4936
daily_vaccinations_raw      3785
daily_vaccinations          188
total_vaccinations_per_hundred 3008
people_vaccinated_per_hundred 3559
people_fully_vaccinated_per_hundred 4936
daily_vaccinations_per_million 188
vaccines                      0
source_name                    0
source_website                 0
dtype: int64
```

describe stastical information on data set

In [24]: `df.describe()`

Out[24]:

	<b>total_vaccinations</b>	<b>people_vaccinated</b>	<b>people_fully_vaccinated</b>	<b>daily_vaccinations_raw</b>	<b>daily_vaccin</b>
<b>count</b>	4.889000e+03	4.338000e+03	2.961000e+03	4.112000e+03	7.70900
<b>mean</b>	2.564337e+06	2.046262e+06	8.760931e+05	9.064633e+04	6.10332
<b>std</b>	9.240731e+06	6.713298e+06	3.526087e+06	2.854173e+05	2.12677
<b>min</b>	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	1.00000
<b>25%</b>	3.967500e+04	3.635900e+04	1.901900e+04	2.708750e+03	1.03600
<b>50%</b>	2.725540e+05	2.479645e+05	1.095010e+05	1.374850e+04	6.24100
<b>75%</b>	1.285437e+06	9.996468e+05	4.820470e+05	5.653450e+04	2.79820
<b>max</b>	1.282170e+08	8.393050e+07	4.553396e+07	4.575496e+06	2.54159

Check the count of all country

In [26]: `df['country'].value_counts()`

Out[26]:

England	100
United Kingdom	100
Wales	100
Canada	100
Scotland	100
...	
Taiwan	2
Guinea	1
Bahamas	1
Laos	1
Equatorial Guinea	1

Name: country, Length: 151, dtype: int64

Check the count of all date

In [28]: `df['date'].value_counts()`

Out[28]:

10-03-21	132
09-03-21	132
07-03-21	131
08-03-21	131
12-03-21	130
...	
17-12-20	8
15-12-20	8
18-12-20	8

```
14-12-20      6
13-12-20      5
Name: date, Length: 101, dtype: int64
```

Check the count of all vaccines

```
In [30]: df['vaccines'].value_counts()
```

```
Out[30]: Moderna, Oxford/AstraZeneca, Pfizer/BioNTech          19
61
Pfizer/BioNTech                                         12
84
Oxford/AstraZeneca, Pfizer/BioNTech                      10
08
Oxford/AstraZeneca                                         9
29
Moderna, Pfizer/BioNTech                                     4
10
Sputnik V                                                 3
09
Pfizer/BioNTech, Sinovac                                2
18
Sinovac                                                 2
03
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V 1
65
Sinopharm/Beijing                                         1
63
Oxford/AstraZeneca, Sinovac                           1
42
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V        1
26
Oxford/AstraZeneca, Sinopharm/Beijing                     1
24
Pfizer/BioNTech, Sinopharm/Beijing                      1
13
Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac            1
99
EpiVacCorona, Sputnik V                               1
99
Johnson&Johnson, Moderna, Pfizer/BioNTech           1
94
Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V       1
90
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V 1
86
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V 1
78
Covaxin, Oxford/AstraZeneca                         1
68
Johnson&Johnson                                 1
36
Sinopharm/Beijing, Sputnik V                        1
33
Moderna                                         1
26
Moderna, Oxford/AstraZeneca                       1
22
Pfizer/BioNTech, Sputnik V                        1
11
Name: vaccines, dtype: int64
```

## Qualitative Data Analysis

## Number of country

In [32]: df['country'].unique()

```
Out[32]: array(['Afghanistan', 'Albania', 'Algeria', 'Andorra', 'Angola',
   'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Australia',
   'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
   'Barbados', 'Belarus', 'Belgium', 'Belize', 'Bermuda', 'Bolivia',
   'Brazil', 'Bulgaria', 'Cambodia', 'Canada', 'Cayman Islands',
   'Chile', 'China', 'Colombia', 'Costa Rica', "Cote d'Ivoire",
   'Croatia', 'Cyprus', 'Czechia', 'Denmark', 'Dominica',
   'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador', 'England',
   'Equatorial Guinea', 'Estonia', 'Faeroe Islands',
   'Falkland Islands', 'Finland', 'France', 'Germany', 'Ghana',
   'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guatemala',
   'Guernsey', 'Guinea', 'Guyana', 'Honduras', 'Hong Kong', 'Hungary',
   'Iceland', 'India', 'Indonesia', 'Iran', 'Ireland', 'Isle of Man',
   'Israel', 'Italy', 'Jamaica', 'Japan', 'Jersey', 'Jordan',
   'Kazakhstan', 'Kenya', 'Kuwait', 'Laos', 'Latvia', 'Lebanon',
   'Liechtenstein', 'Lithuania', 'Luxembourg', 'Macao', 'Malawi',
   'Malaysia', 'Maldives', 'Malta', 'Mauritius', 'Mexico', 'Moldova',
   'Monaco', 'Mongolia', 'Montenegro', 'Montserrat', 'Morocco',
   'Mozambique', 'Myanmar', 'Nepal', 'Netherlands', 'New Zealand',
   'Nigeria', 'North Macedonia', 'Northern Cyprus',
   'Northern Ireland', 'Norway', 'Oman', 'Pakistan', 'Panama',
   'Paraguay', 'Peru', 'Philippines', 'Poland', 'Portugal', 'Qatar',
   'Romania', 'Russia', 'Rwanda', 'Saint Helena',
   'Saint Kitts and Nevis', 'Saint Lucia',
   'Saint Vincent and the Grenadines', 'San Marino', 'Saudi Arabia',
   'Scotland', 'Senegal', 'Serbia', 'Seychelles', 'Singapore',
   'Slovakia', 'Slovenia', 'South Africa', 'South Korea', 'Spain',
   'Sri Lanka', 'Suriname', 'Sweden', 'Switzerland', 'Taiwan',
   'Thailand', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
   'Turks and Caicos Islands', 'Uganda', 'Ukraine',
   'United Arab Emirates', 'United Kingdom', 'United States',
   'Uruguay', 'Venezuela', 'Vietnam', 'Wales', 'Zimbabwe'],
  dtype=object)
```

## number of source\_name

In [34]: df['source\_name'].unique()

```
Out[34]: array(['Government of Afghanistan', 'Ministry of Health',
   'Government of Andorra',
   'Government of Australia via covidlive.com.au',
   'Government of Azerbaijan', 'Government of the Bahamas',
   'Directorate General of Health Services', 'Sciensano',
   'Government of Bermuda',
   'Official data from provinces via covid19tracker.ca',
   'Cayman Islands Government',
   'Department of Statistics and Health Information',
   'National Health Commission',
   'Ministry of Health via Universidad de La Sabana',
   'Costa Rican Social Security Fund',
   'Ministry of Health and Public Hygiene', 'Government of Cyprus',
   'Statens Serum Institut', 'Government of Dominica',
   'Ministry of Public Health', 'Government of Ecuador via Ecuacovid',
   'Government of the United Kingdom', 'National Health Board',
   'Government of the Faeroe Islands',
   'Government of the Falkland Islands',
   'Finnish Institute for Health and Welfare', 'Public Health France',
   'Robert Koch Institut', 'Government of Ghana',
   'Government of Gibraltar', 'Government of Greenland',
   'Government of Grenada', 'Government of Guernsey',
   'National Health Security Agency', 'Government of Honduras',
```

```
'Government of Hong Kong', 'Government of Hungary',
'Directorate of Health', 'Government of Iran',
'Health Service Executive', 'Isle of Man Government',
'Government of Israel',
'Extraordinary commissioner for the Covid-19 emergency',
'Government of Jamaica', 'Ministry of Health, Labour and Welfare',
'Government of Jersey', 'Government of Jordan',
'Government of Kazakhstan', 'Government of Laos',
'National Health Service', 'Government of Lebanon',
'Federal Office of Public Health', 'Government of Luxembourg',
'Government of Macao', 'Government of Malaysia',
'Presidency of the Maldives',
'COVID-19 Malta Public Health Response Team',
'National Communication Committee on COVID-19',
'Secretary of Health', 'National Council',
'Ministry of Health via ikon.mn', 'Government of Montenegro',
'Government ofMontserrat',
'Deputy National Director of Public Health', 'Government of Nepal',
'National Institute for Public Health and the Environment',
'Government of Nigeria', 'Government of North Macedonia',
'Norwegian Institute of Public Health',
'National Command and Operation Centre', 'Government of Paraguay',
'Government of the Philippines',
'General Directorate of Health via Data Science for Social Good',
'Government of Romania',
'Official data from local governments via gogov.ru',
'Government of Saint Helena',
'Government of Saint Kitts and Nevis', 'Social Security Institute',
'Saudi Health Council', 'Government of Serbia',
'Extended Programme for Immunisation',
'National Institute of Public Health, via Sledilnik',
'Korea Centers for Disease Control and Prevention',
'Government of Suriname', 'Public Health Agency of Sweden',
'Taiwan Centers for Disease Control', 'Government of Thailand',
'COVID-19 Vaccine Information Platform',
'National Emergency Crisis and Disaster Management Authority',
'Centers for Disease Control and Prevention',
'Government of Venezuela', 'Government of Vietnam'], dtype=object)
```

To find source\_name maximum in dataset

```
In [36]: df['country'].max()
```

```
Out[36]: 'Zimbabwe'
```

To find source\_name minimum in dataset

```
In [38]: df["country"].min()
```

```
Out[38]: 'Afghanistan'
```

To find vaccines maximum in dataset

```
In [40]: df['vaccines'].max()
```

```
Out[40]: 'Sputnik V'
```

To find vaccines minmum in dataset

```
In [42]: df['vaccines'].min()
```

```
Out[42]: 'Covaxin, Oxford/AstraZeneca'
```

# Quantitative Analysis

countrywise vaccination

```
In [44]: df.groupby('country').sum()['total']
```

```
Out[44]: (   total_vaccinations  people_vaccinated  people_fully_vaccinated \
country
Afghanistan      62200.0          62200.0              0.0
Albania          239753.0         14501.0            2316.0
Algeria           75030.0             0.0              0.0
Andorra           20107.0          15145.0            2436.0
Angola            55169.0          55169.0              0.0
...
Uruguay          4103340.0        4103340.0              0.0
Venezuela         12351.0          12351.0              0.0
Vietnam           289709.0         289709.0              0.0
Wales              55567674.0       49549552.0        6018122.0
Zimbabwe          895175.0          895175.0              0.0

   daily_vaccinations_raw  daily_vaccinations \
country
Afghanistan          0.0          49509.0
Albania               6358.0        42547.0
Algeria                30.0        69070.0
Andorra                 0.0        3357.0
Angola                 0.0        42971.0
...
Uruguay              376322.0       329386.0
Venezuela               0.0          8704.0
Vietnam                37911.0       33233.0
Wales                  1530045.0      1551942.0
Zimbabwe              42821.0          43211.0

   total_vaccinations_per_hundred  people_vaccinated_per_hundred \
country
Afghanistan            0.16            0.16
Albania                 8.35            0.50
Algeria                  0.17            0.00
Andorra                  26.02           19.60
Angola                   0.17            0.17
...
Uruguay                 118.10           118.10
Venezuela                  0.04            0.04
Vietnam                   0.29            0.29
Wales                     1762.41          1571.55
Zimbabwe                  6.02            6.02

   people_fully_vaccinated_per_hundred \
country
Afghanistan                0.00
Albania                      0.08
Algeria                      0.00
Andorra                      3.16
Angola                      0.00
...
Uruguay                      0.00
Venezuela                     0.00
Vietnam                      0.00
Wales                        190.87
Zimbabwe                      0.00

   daily_vaccinations_per_million
country
```

Afghanistan	1277.0
Albania	14781.0
Algeria	1568.0
Andorra	43446.0
Angola	1308.0
...	...
Uruguay	94822.0
Venezuela	305.0
Vietnam	343.0
Wales	492235.0
Zimbabwe	2909.0

[151 rows x 9 columns],  
['total'])

Date wise vaccination

In [45]: df.groupby('date').sum()[['total']]

Out[45]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	\
date				
01-01-21	1611555.0	1505278.0	0.0	
01-02-21	86154870.0	67407904.0	11871526.0	
01-03-21	225110282.0	164802060.0	52974001.0	
02-01-21	6867460.0	6758728.0	0.0	
02-02-21	88637838.0	69667120.0	12353047.0	
...	...	...	...	
29-12-20	1012502.0	939830.0	0.0	
30-01-21	78140717.0	62258022.0	10225901.0	
30-12-20	4126238.0	4039682.0	0.0	
31-01-21	106527384.0	65452620.0	11131846.0	
31-12-20	6092550.0	1452371.0	0.0	
	daily_vaccinations_raw	daily_vaccinations	\	
date				
01-01-21	118823.0	940739.0		
01-02-21	3113523.0	4888111.0		
01-03-21	4445974.0	6533377.0		
02-01-21	180547.0	1018936.0		
02-02-21	2868788.0	5074628.0		
...	...	...	...	
29-12-20	261199.0	759495.0		
30-01-21	4235414.0	4693542.0		
30-12-20	305909.0	840175.0		
31-01-21	2981337.0	4662341.0		
31-12-20	268865.0	878520.0		
	total_vaccinations_per_hundred	people_vaccinated_per_hundred	\	
date				
01-01-21	18.44	17.98		
01-02-21	341.73	261.43		
01-03-21	866.54	576.16		
02-01-21	21.47	21.18		
02-02-21	369.46	282.97		
...	...	...	...	
29-12-20	13.05	12.87		
30-01-21	290.25	224.62		
30-12-20	18.33	18.12		
31-01-21	308.81	232.35		
31-12-20	18.47	17.57		
	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million		
date				
01-01-21	0.00	34168.0		
01-02-21	46.40	211262.0		

		total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
01-03-21		184.39			345161.0
02-01-21		0.00			34518.0
02-02-21		52.41			212078.0
...		...			...
29-12-20		0.00			27549.0
30-01-21		37.83			201276.0
30-12-20		0.00			30353.0
31-01-21		44.58			204535.0
31-12-20		0.00			34326.0

[101 rows x 9 columns],  
['total'])

Number of daily\_vaccinations vaccinated by country

```
In [46]: print (df.groupby("country")["daily_vaccinations"].sum())
```

country	daily_vaccinations
Afghanistan	49509.0
Albania	42547.0
Algeria	69070.0
Andorra	3357.0
Angola	42971.0
...	...
Uruguay	329386.0
Venezuela	8704.0
Vietnam	33233.0
Wales	1551942.0
Zimbabwe	43211.0

Name: daily\_vaccinations, Length: 151, dtype: float64

Number of people\_vaccinated using by source\_name

```
In [47]: print (df.groupby("source_name")["people_vaccinated"].sum())
```

source_name	people_vaccinated
COVID-19 Malta Public Health Response Team	3.035972e+06
COVID-19 Vaccine Information Platform	2.660271e+08
Cayman Islands Government	2.340910e+05
Centers for Disease Control and Prevention	2.876310e+09
Costa Rican Social Security Fund	1.143586e+06
...	...
Sciensano	3.240023e+07
Secretary of Health	9.163766e+07
Social Security Institute	7.759500e+04
Statens Serum Institut	2.438263e+07
Taiwan Centers for Disease Control	1.578000e+03

Name: people\_vaccinated, Length: 92, dtype: float64

```
In [50]: country_vaccinations=pd.DataFrame(df)
```

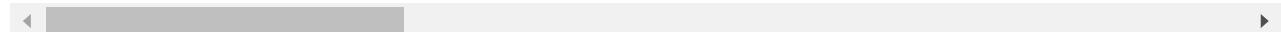
```
In [51]: df
```

```
Out[51]:
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
0	Afghanistan	AFG	22-02-21	0.0	0.0	NaN	NaN
1	Afghanistan	AFG	23-02-21	NaN	NaN	NaN	NaN

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_v
2	Afghanistan	AFG	24-02-21		NaN	NaN	NaN
3	Afghanistan	AFG	25-02-21		NaN	NaN	NaN
4	Afghanistan	AFG	26-02-21		NaN	NaN	NaN
...	...	...	...	...	...	...	...
7892	Zimbabwe	ZWE	19-03-21	41687.0	41687.0	41687.0	NaN
7893	Zimbabwe	ZWE	20-03-21	42210.0	42210.0	42210.0	NaN
7894	Zimbabwe	ZWE	21-03-21	42729.0	42729.0	42729.0	NaN
7895	Zimbabwe	ZWE	22-03-21	43294.0	43294.0	43294.0	NaN
7896	Zimbabwe	ZWE	23-03-21	44135.0	44135.0	44135.0	NaN

7897 rows × 15 columns



Query to extract the record of india in the country.

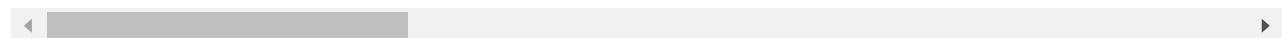
In [59]: df\_India=country\_vaccinations.loc[country\_vaccinations["country"]=="India"]

In [58]: df\_India

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vacc
3320	India	IND	15-01-21	0.0	0.0	0.0	NaN
3321	India	IND	16-01-21	191181.0	191181.0	191181.0	NaN
3322	India	IND	17-01-21	224301.0	224301.0	224301.0	NaN
3323	India	IND	18-01-21	454049.0	454049.0	454049.0	NaN

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vacc
3324	India	IND	19-01-21	674835.0	674835.0		Nan
...	...	...	...	...	...	...	...
3383	India	IND	19-03-21	42063392.0	34842030.0	7221362.0	
3384	India	IND	20-03-21	44603841.0	37125187.0	7478654.0	
3385	India	IND	21-03-21	45065998.0	37574302.0	7491696.0	
3386	India	IND	22-03-21	48494594.0	40631153.0	7863441.0	
3387	India	IND	23-03-21	50841286.0	42731952.0	8109334.0	

68 rows × 15 columns



## Statistical Analysis on Dataset

maximum value per column

In [60]: `df.max()`

```
Out[60]: country                               Zimbabwe
          iso_code                                ZWE
          date                                 31-12-20
          total_vaccinations  1.28217e+08
          people_vaccinated   8.39305e+07
          people_fully_vaccinated  4.5534e+07
          daily_vaccinations_raw  4.5755e+06
          daily_vaccinations    2.5416e+06
          total_vaccinations_per_hundred  159.87
          people_vaccinated_per_hundred  89.55
          people_fully_vaccinated_per_hundred  70.32
          daily_vaccinations_per_million  54264
          vaccines                                Sputnik V
          source_name                             Taiwan Centers for Disease Control
          source_website  https://www.youtube.com/watch?v=XmZfFCD2Wfw
          dtype: object
```

minimum value per column

In [61]: `df.min()`

```
Out[61]: country                               Afghanistan
          iso_code                                AFG
          date                                 01-01-21
          total_vaccinations  0
```

```

people_vaccinated          0
people_fully_vaccinated    1
daily_vaccinations_raw     0
daily_vaccinations         1
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                 Covaxin, Oxford/AstraZeneca
source_website               COVID-19 Malta Public Health Response Team
                             http://covid19.ncema.gov.ae/en
dtype: object

```

median value per column

In [62]: `df.median()`

```

Out[62]: total_vaccinations      272554.00
          people_vaccinated       247964.50
          people_fully_vaccinated  109501.00
          daily_vaccinations_raw  13748.50
          daily_vaccinations      6241.00
          total_vaccinations_per_hundred 3.72
          people_vaccinated_per_hundred 3.19
          people_fully_vaccinated_per_hundred 1.47
          daily_vaccinations_per_million 1291.00
          dtype: float64

```

Standard deviation value per column

In [63]: `df.std()`

```

Out[63]: total_vaccinations      9.240731e+06
          people_vaccinated       6.713298e+06
          people_fully_vaccinated  3.526087e+06
          daily_vaccinations_raw  2.854173e+05
          daily_vaccinations      2.126771e+05
          total_vaccinations_per_hundred 1.764818e+01
          people_vaccinated_per_hundred 1.196153e+01
          people_fully_vaccinated_per_hundred 7.695151e+00
          daily_vaccinations_per_million 4.183033e+03
          dtype: float64

```

Variance value per column

In [65]: `df.var()`

```

Out[65]: total_vaccinations      8.539110e+13
          people_vaccinated       4.506837e+13
          people_fully_vaccinated  1.243329e+13
          daily_vaccinations_raw  8.146305e+10
          daily_vaccinations      4.523154e+10
          total_vaccinations_per_hundred 3.114583e+02
          people_vaccinated_per_hundred 1.430781e+02
          people_fully_vaccinated_per_hundred 5.921536e+01
          daily_vaccinations_per_million 1.749777e+07
          dtype: float64

```

lower quartile/first quartile

In [66]: `df.quantile(0.25)`

```

Out[66]: total_vaccinations      39675.00
          people_vaccinated       36359.00
          people_fully_vaccinated  19019.00
          daily_vaccinations_raw  2708.75
          daily_vaccinations      1036.00

```

```
total_vaccinations_per_hundred      0.78
people_vaccinated_per_hundred      0.80
people_fully_vaccinated_per_hundred 0.40
daily_vaccinations_per_million     365.00
Name: 0.25, dtype: float64
```

second quartile/median

In [67]: `df.quantile(0.5)`

```
total_vaccinations           272554.00
people_vaccinated            247964.50
people_fully_vaccinated       109501.00
daily_vaccinations_raw       13748.50
daily_vaccinations           6241.00
total_vaccinations_per_hundred 3.72
people_vaccinated_per_hundred 3.19
people_fully_vaccinated_per_hundred 1.47
daily_vaccinations_per_million 1291.00
Name: 0.5, dtype: float64
```

upper quartile

In [68]: `df.quantile(0.75)`

```
total_vaccinations           1285437.00
people_vaccinated             999646.75
people_fully_vaccinated        482047.00
daily_vaccinations_raw        56534.50
daily_vaccinations            27982.00
total_vaccinations_per_hundred 10.82
people_vaccinated_per_hundred 8.46
people_fully_vaccinated_per_hundred 3.20
daily_vaccinations_per_million 3117.00
Name: 0.75, dtype: float64
```

Interquartile range (IQR) on dataset

In [69]: `df.quantile(0.75)-df.quantile(0.25)`

```
total_vaccinations           1245762.00
people_vaccinated             963287.75
people_fully_vaccinated        463028.00
daily_vaccinations_raw        53825.75
daily_vaccinations            26946.00
total_vaccinations_per_hundred 10.04
people_vaccinated_per_hundred 7.66
people_fully_vaccinated_per_hundred 2.80
daily_vaccinations_per_million 2752.00
dtype: float64
```

corelation values on dataset

In [70]: `df.corr()`

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
<b>total_vaccinations</b>	1.000000	0.990179	0.947367	
<b>people_vaccinated</b>	0.990179	1.000000	0.893220	
<b>people_fully_vaccinated</b>	0.947367	0.893220	1.000000	
<b>daily_vaccinations_raw</b>	0.878029	0.885668	0.793317	
<b>daily_vaccinations</b>	0.942280	0.950743	0.861147	

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
total_vaccinations_per_hundred	0.173721	0.174280	0.129563	
people_vaccinated_per_hundred	0.188284	0.201812	0.110935	
people_fully_vaccinated_per_hundred	0.087326	0.058450	0.146625	
daily_vaccinations_per_million	0.115324	0.121638	0.098386	

covariance values on dataset

In [71]: `df.cov()`

Out[71]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
total_vaccinations	8.539110e+13	6.372826e+13	3.757908e+13	
people_vaccinated	6.372826e+13	4.506837e+13	2.511152e+13	
people_fully_vaccinated	3.757908e+13	2.511152e+13	1.243329e+13	
daily_vaccinations_raw	2.435739e+12	1.924818e+12	1.007604e+12	
daily_vaccinations	2.156216e+12	1.617999e+12	8.855444e+11	
total_vaccinations_per_hundred	2.833073e+07	2.099663e+07	9.195902e+06	
people_vaccinated_per_hundred	2.155438e+07	1.620575e+07	5.206887e+06	
people_fully_vaccinated_per_hundred	7.559364e+06	3.586032e+06	3.978491e+06	
daily_vaccinations_per_million	4.411471e+09	3.414601e+09	1.427984e+09	

## Data Visualization

In [80]: `import seaborn as sns  
import matplotlib.pyplot as plt  
import seaborn as sb`

Data Visualization of country wise covid vaccine doses administered by histogram

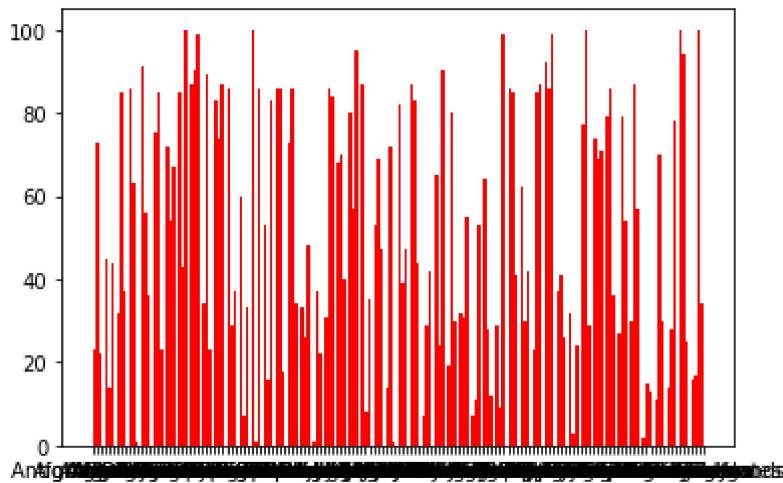
In [75]: `plt.hist(df['country'], bins=200, color="red")`

Out[75]: `(array([ 23., 73., 22., 0., 45., 14., 44., 0., 32., 85., 37., 0., 86., 63., 1., 0., 91., 56., 36., 0., 75., 85., 23., 0., 72., 54., 67., 0., 85., 43., 100., 0., 87., 90., 99., 0., 34., 89., 23., 0., 83., 74., 87., 0., 86., 29., 37., 0., 60., 7., 33., 0., 100., 1., 86., 0., 53., 16., 83., 0., 86., 86., 18., 0., 73., 86., 34., 0., 33., 26., 48., 0., 1., 37., 22., 0., 31., 86., 84., 0., 68., 70., 40., 0., 80., 57., 95., 0., 87., 8., 35., 0., 53., 69., 47., 0., 14., 72., 1., 0., 82., 39., 47., 0., 87., 83., 44., 0., 7., 29., 42., 0., 65., 24., 90., 0., 19., 80., 30., 0., 32., 31., 55., 0., 7., 11., 53., 0., 64., 28., 12., 0., 29., 9., 99., 0., 86., 85., 41., 0., 62., 30., 42., 0., 23., 85., 87., 0., 92., 86., 99., 0., 37., 41., 26., 0., 32., 3., 24., 0., 77., 100., 29., 0., 74., 69., 71., 0., 79., 86., 36., 0., 27., 79., 54., 0.],`

```

    30., 87., 57., 0., 2., 15., 13., 0., 11., 70., 30.,
    0., 14., 28., 78., 0., 100., 94., 25., 0., 16., 17.,
    100., 34.]),
array([
  0. , 0.75, 1.5 , 2.25, 3. , 3.75, 4.5 , 5.25,
  6. , 6.75, 7.5 , 8.25, 9. , 9.75, 10.5 , 11.25,
  12. , 12.75, 13.5 , 14.25, 15. , 15.75, 16.5 , 17.25,
  18. , 18.75, 19.5 , 20.25, 21. , 21.75, 22.5 , 23.25,
  24. , 24.75, 25.5 , 26.25, 27. , 27.75, 28.5 , 29.25,
  30. , 30.75, 31.5 , 32.25, 33. , 33.75, 34.5 , 35.25,
  36. , 36.75, 37.5 , 38.25, 39. , 39.75, 40.5 , 41.25,
  42. , 42.75, 43.5 , 44.25, 45. , 45.75, 46.5 , 47.25,
  48. , 48.75, 49.5 , 50.25, 51. , 51.75, 52.5 , 53.25,
  54. , 54.75, 55.5 , 56.25, 57. , 57.75, 58.5 , 59.25,
  60. , 60.75, 61.5 , 62.25, 63. , 63.75, 64.5 , 65.25,
  66. , 66.75, 67.5 , 68.25, 69. , 69.75, 70.5 , 71.25,
  72. , 72.75, 73.5 , 74.25, 75. , 75.75, 76.5 , 77.25,
  78. , 78.75, 79.5 , 80.25, 81. , 81.75, 82.5 , 83.25,
  84. , 84.75, 85.5 , 86.25, 87. , 87.75, 88.5 , 89.25,
  90. , 90.75, 91.5 , 92.25, 93. , 93.75, 94.5 , 95.25,
  96. , 96.75, 97.5 , 98.25, 99. , 99.75, 100.5 , 101.25,
  102. , 102.75, 103.5 , 104.25, 105. , 105.75, 106.5 , 107.25,
  108. , 108.75, 109.5 , 110.25, 111. , 111.75, 112.5 , 113.25,
  114. , 114.75, 115.5 , 116.25, 117. , 117.75, 118.5 , 119.25,
  120. , 120.75, 121.5 , 122.25, 123. , 123.75, 124.5 , 125.25,
  126. , 126.75, 127.5 , 128.25, 129. , 129.75, 130.5 , 131.25,
  132. , 132.75, 133.5 , 134.25, 135. , 135.75, 136.5 , 137.25,
  138. , 138.75, 139.5 , 140.25, 141. , 141.75, 142.5 , 143.25,
  144. , 144.75, 145.5 , 146.25, 147. , 147.75, 148.5 , 149.25,
  150. ]),
<BarContainer object of 200 artists>

```



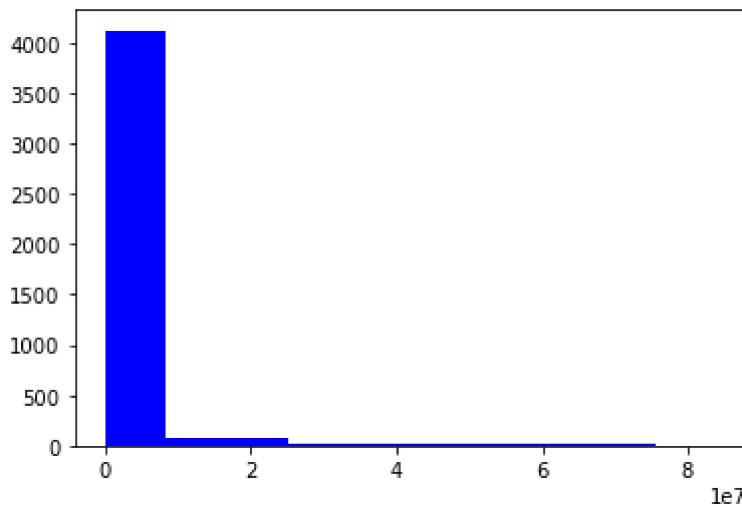
Data Visualization of people\_vaccinated by histogram

```
In [77]: plt.hist(df["people_vaccinated"], bins=10, color="blue")
```

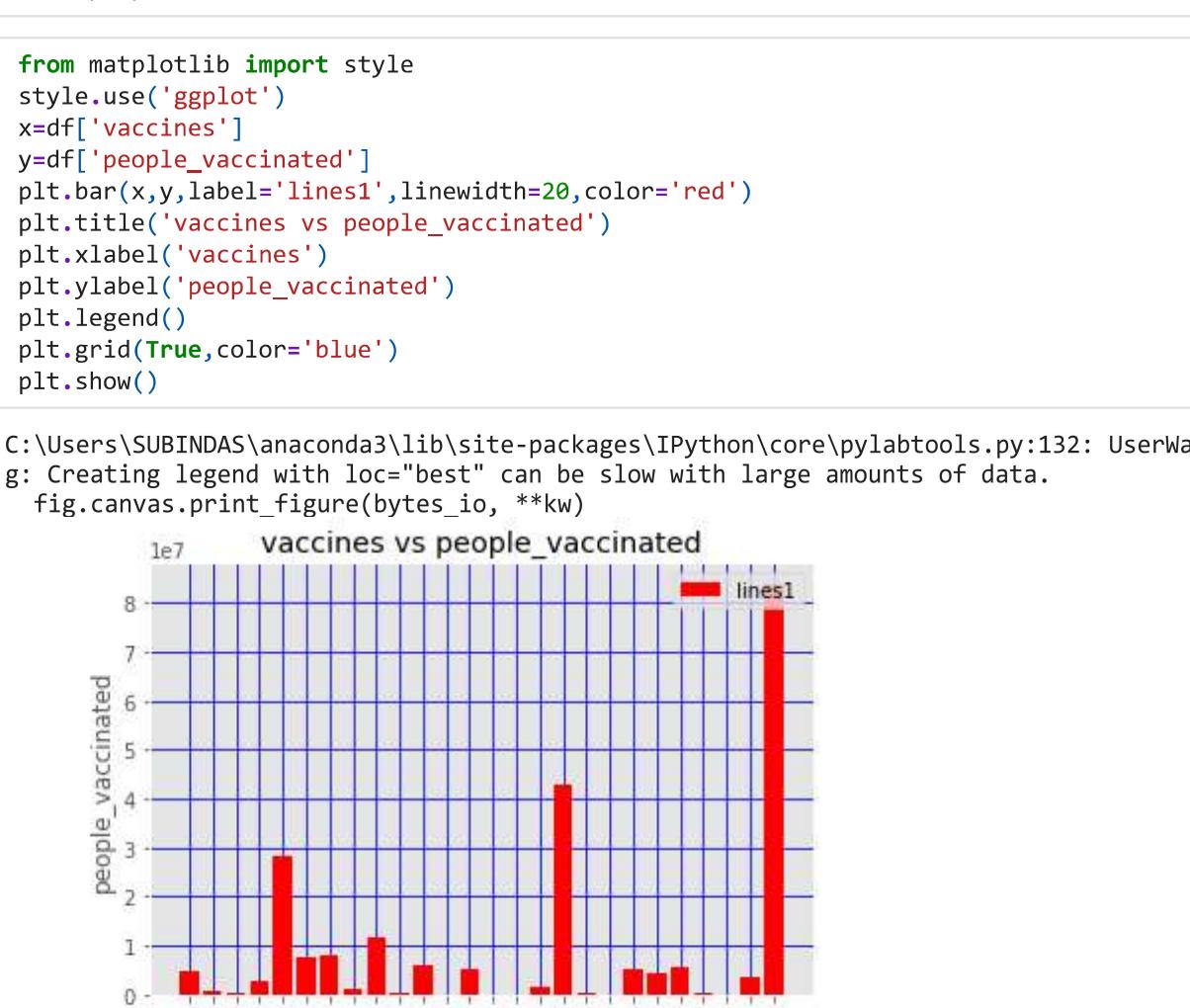
```

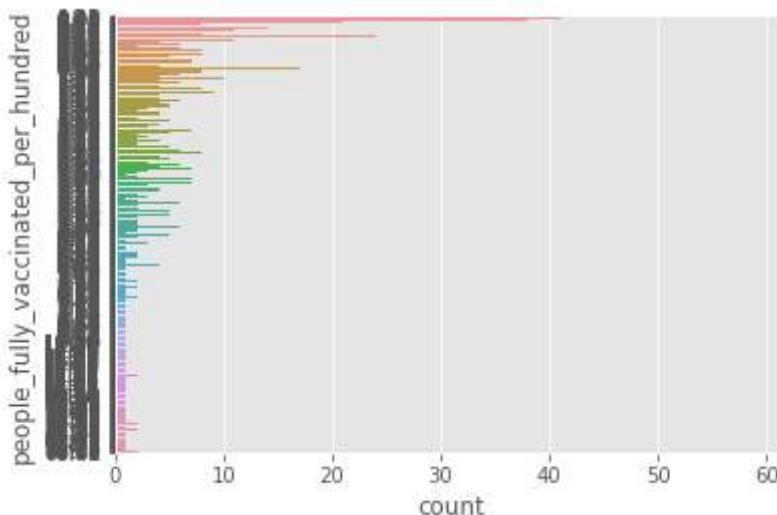
Out[77]: (array([4123., 81., 67., 21., 12., 11., 6., 6., 6.,
      5.]),
array([
  0. , 8393049.5, 16786099. , 25179148.5, 33572198. ,
  41965247.5, 50358297. , 58751346.5, 67144396. , 75537445.5,
  83930495. ]),
<BarContainer object of 10 artists>

```



C:\Users\SUBINDAS\anaconda3\lib\site-packages\IPython\core\pylabtools.py:132: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.

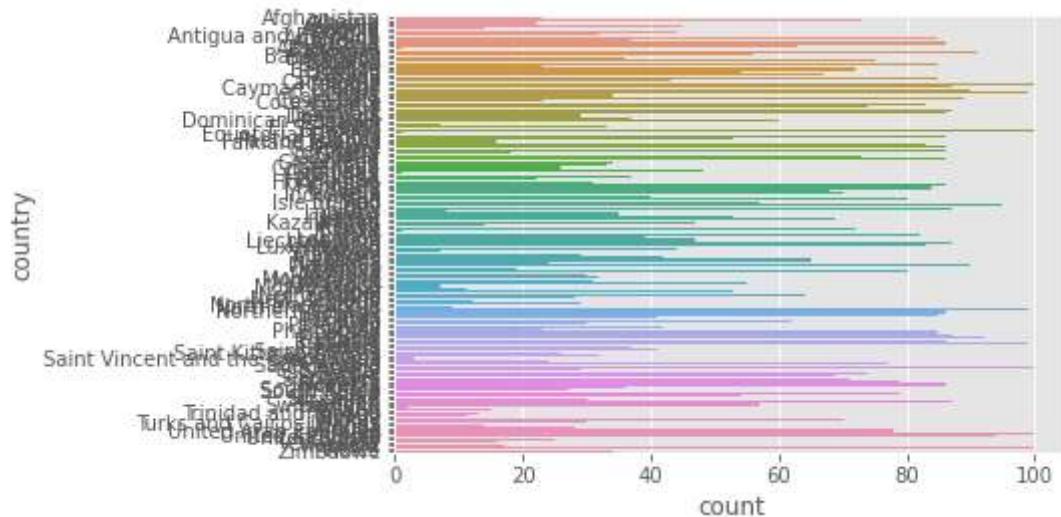




To count the total people\_vaccinated by country

```
In [82]: sb.countplot(y='country', data=df)
```

```
Out[82]: <AxesSubplot:xlabel='count', ylabel='country'>
```

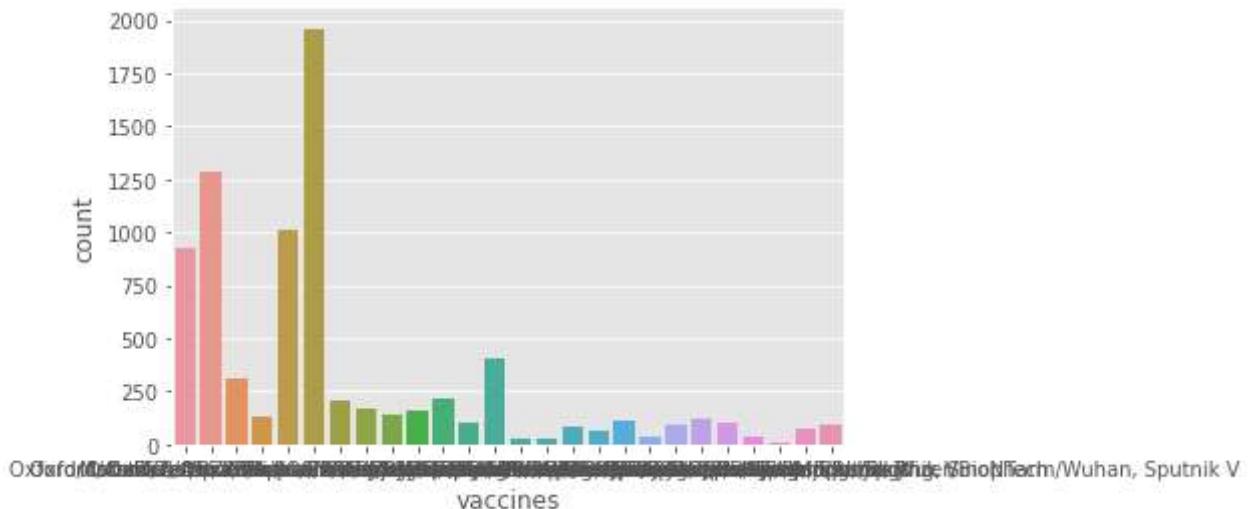


Most popular vaccines

```
In [83]: sns.countplot(df["vaccines"])
```

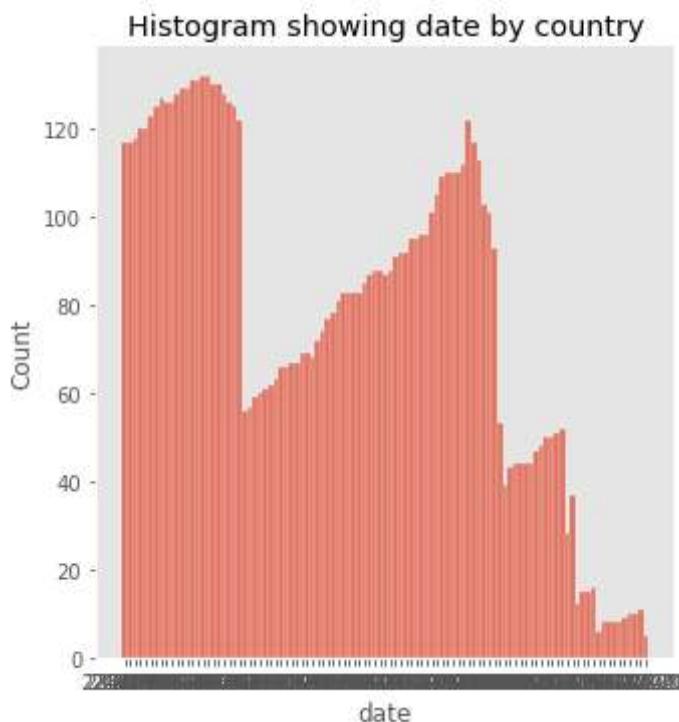
```
C:\Users\SUBINDAS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
    warnings.warn(
```

```
Out[83]: <AxesSubplot:xlabel='vaccines', ylabel='count'>
```



Histogram showing date by country

```
In [85]: sns.displot(df["date"], bins=20)
plt.title("Histogram showing date by country")
plt.grid()
plt.show()
```



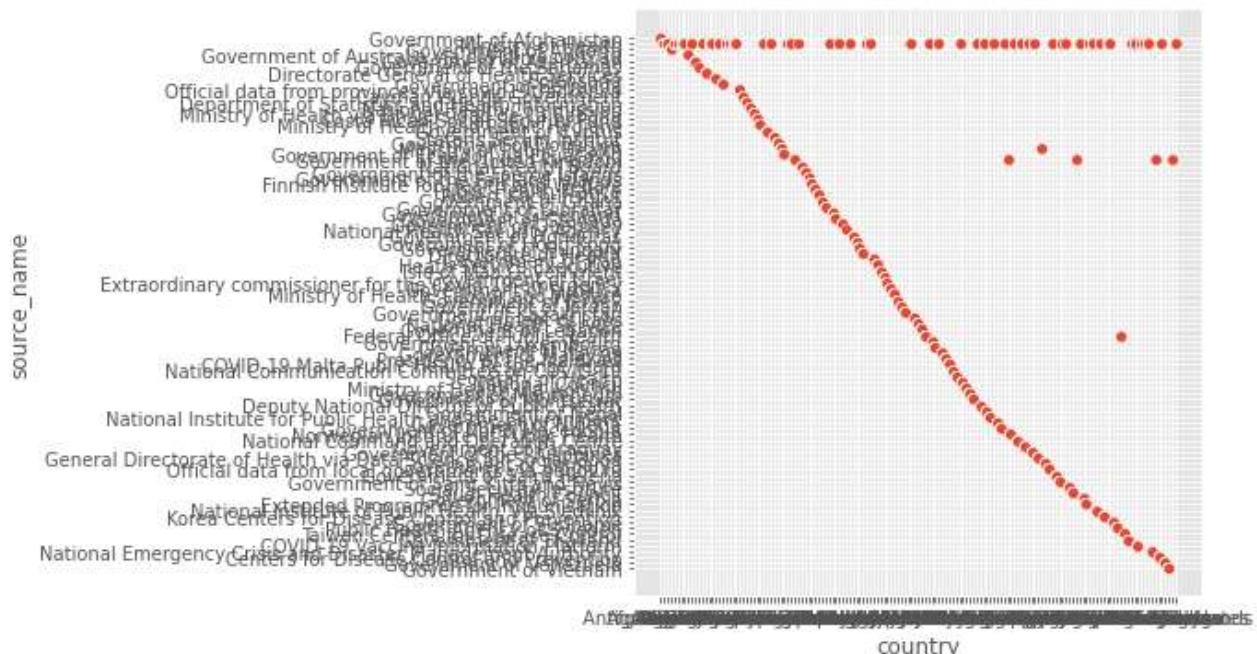
Scatterplot showing source\_name by country

```
In [87]: sns.relplot(x="country", y="source_name", data=df)
```

C:\Users\SUBINDAS\anaconda3\lib\site-packages\seaborn\axisgrid.py:64: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorations.

```
    self.fig.tight_layout(*args, **kwargs)
```

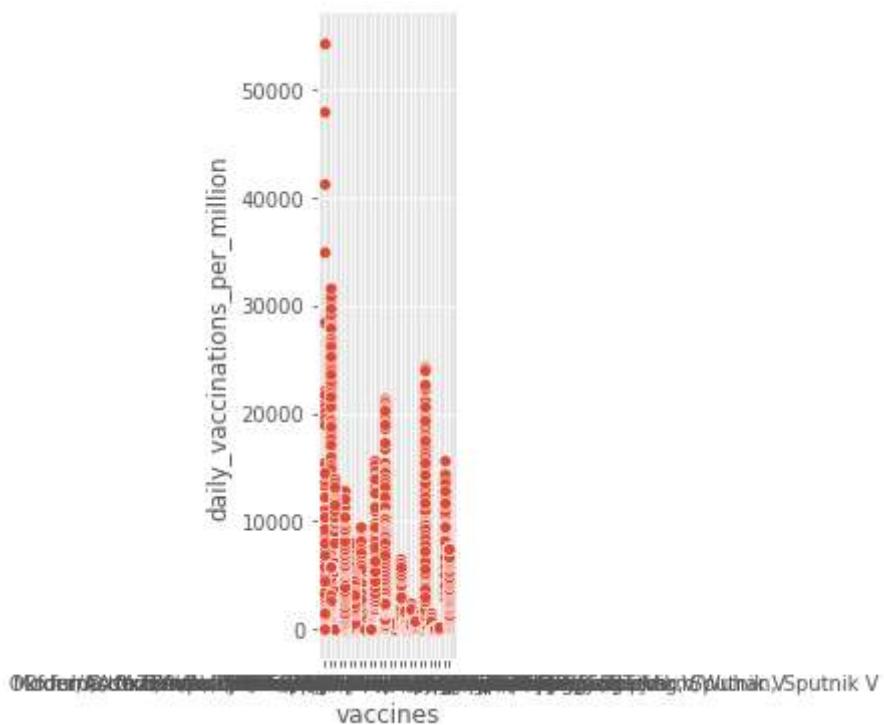
```
Out[87]: <seaborn.axisgrid.FacetGrid at 0xc977d30>
```



Scatterplot showing daily\_vaccinations\_per\_million using vaccines

```
In [89]: sns.relplot(x="vaccines",y="daily_vaccinations_per_million",data=df)
```

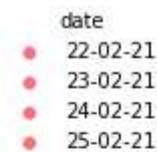
```
Out[89]: <seaborn.axisgrid.FacetGrid at 0xc4879a0>
```

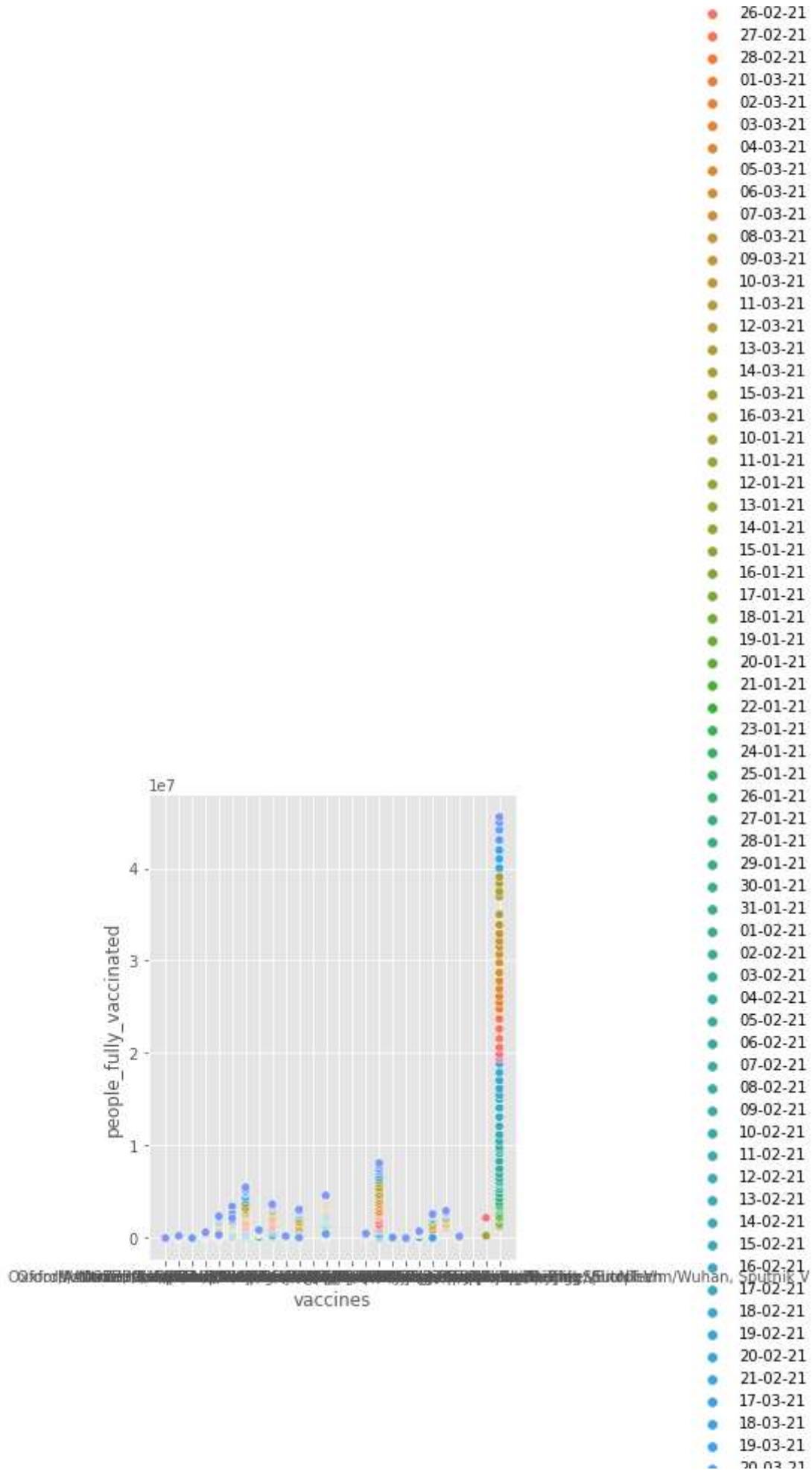


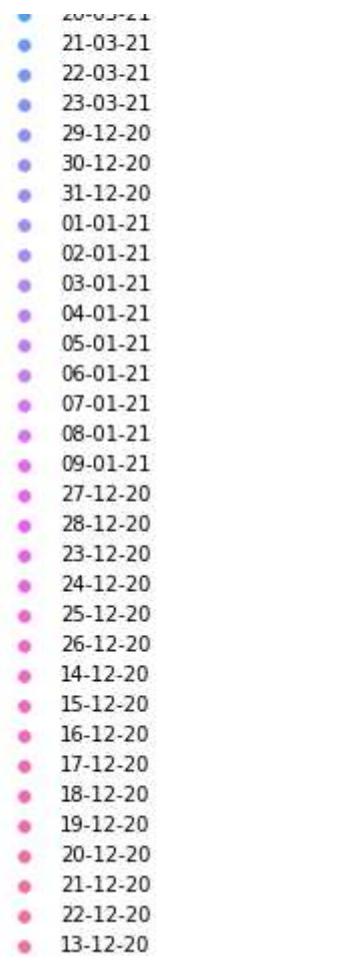
Scatterplot showing people\_fully\_vaccinated using vaccines by date

```
In [92]: sns.relplot(x="vaccines",y="people_fully_vaccinated",hue="date",data=df)
```

```
Out[92]: <seaborn.axisgrid.FacetGrid at 0xc119460>
```



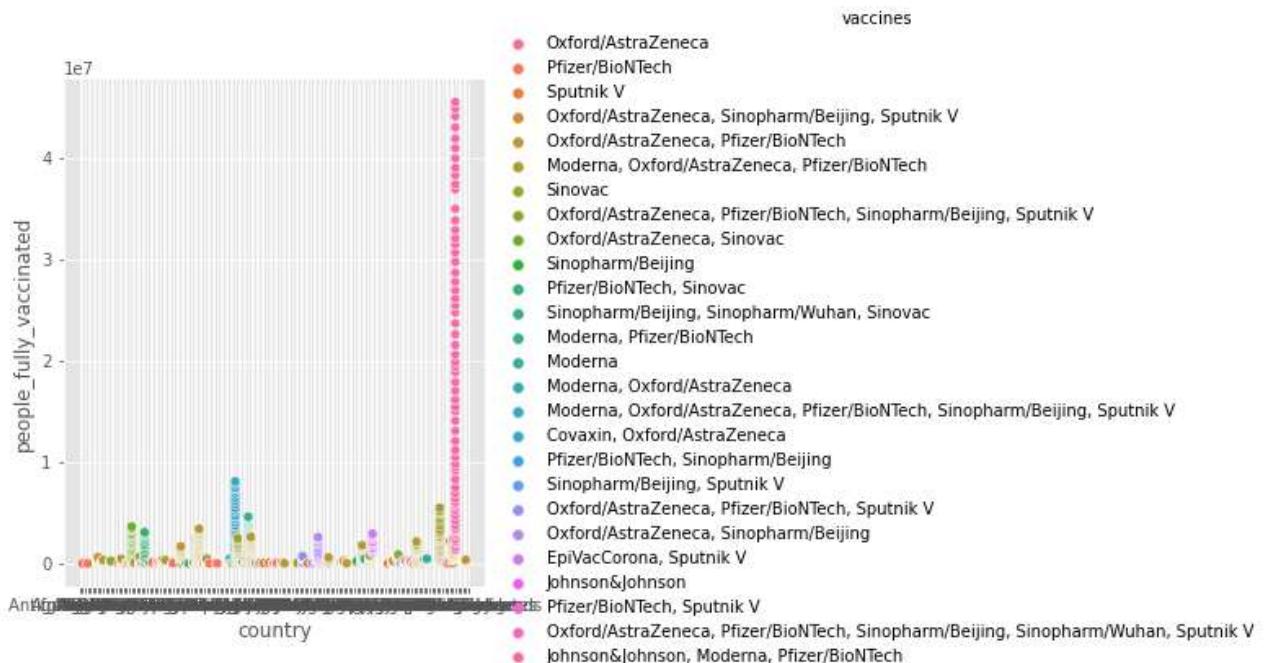




Scatterplot showing people\_fully\_vaccinated using country by vaccines

```
In [95]: sns.relplot(x="country",y="people_fully_vaccinated",hue="vaccines",data=df)
```

```
Out[95]: <seaborn.axisgrid.FacetGrid at 0xca6cb08>
```



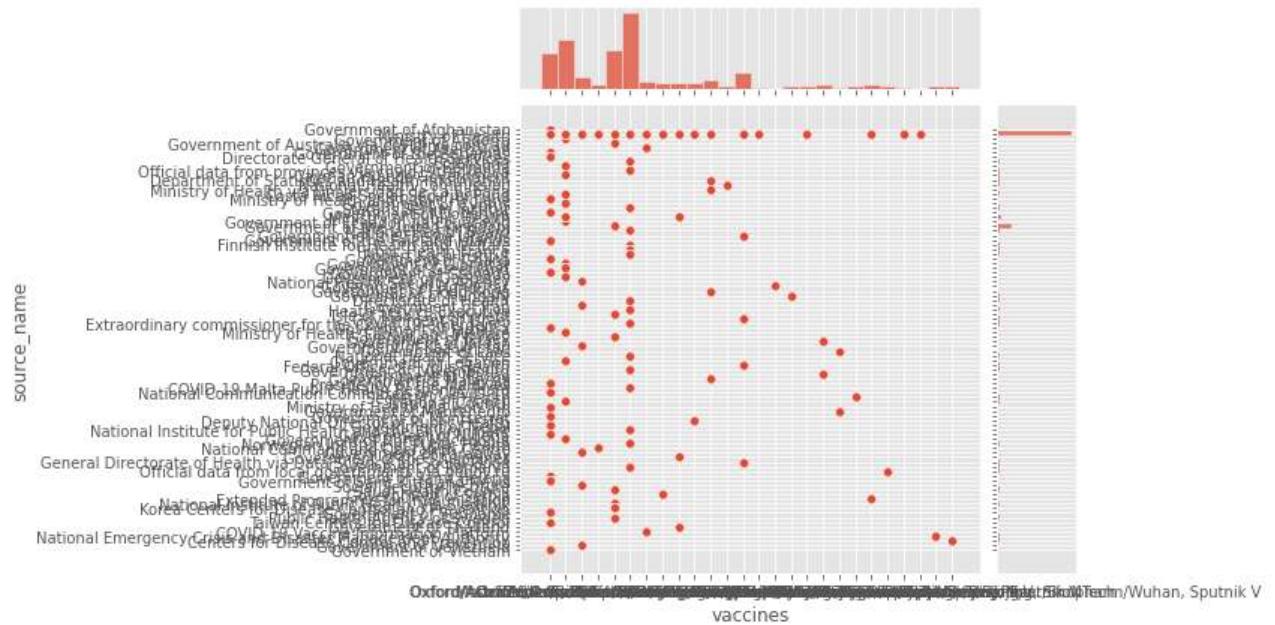
Data Visualization of source\_name by vaccines

```
In [96]: sns.jointplot(df["vaccines"],df["source_name"])
```

```
C:\Users\SUBINDAS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid po-
sitional argument will be `data`, and passing other arguments without an explicit keywor-
d will result in an error or misinterpretation.

warnings.warn(
```

```
Out[96]: <seaborn.axisgrid.JointGrid at 0xa7e9310>
```



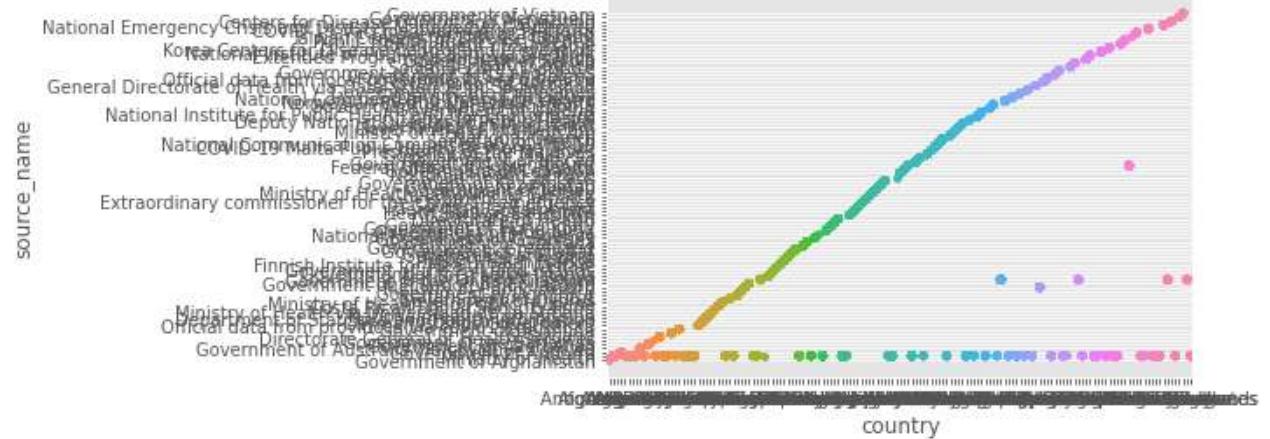
Data Visualization of source\_name by country

```
In [97]: sns.stripplot(df["country"], df["source_name"])
```

```
C:\Users\SUBINDAS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid po-
sitional argument will be `data`, and passing other arguments without an explicit keywor-
d will result in an error or misinterpretation.

warnings.warn(
```

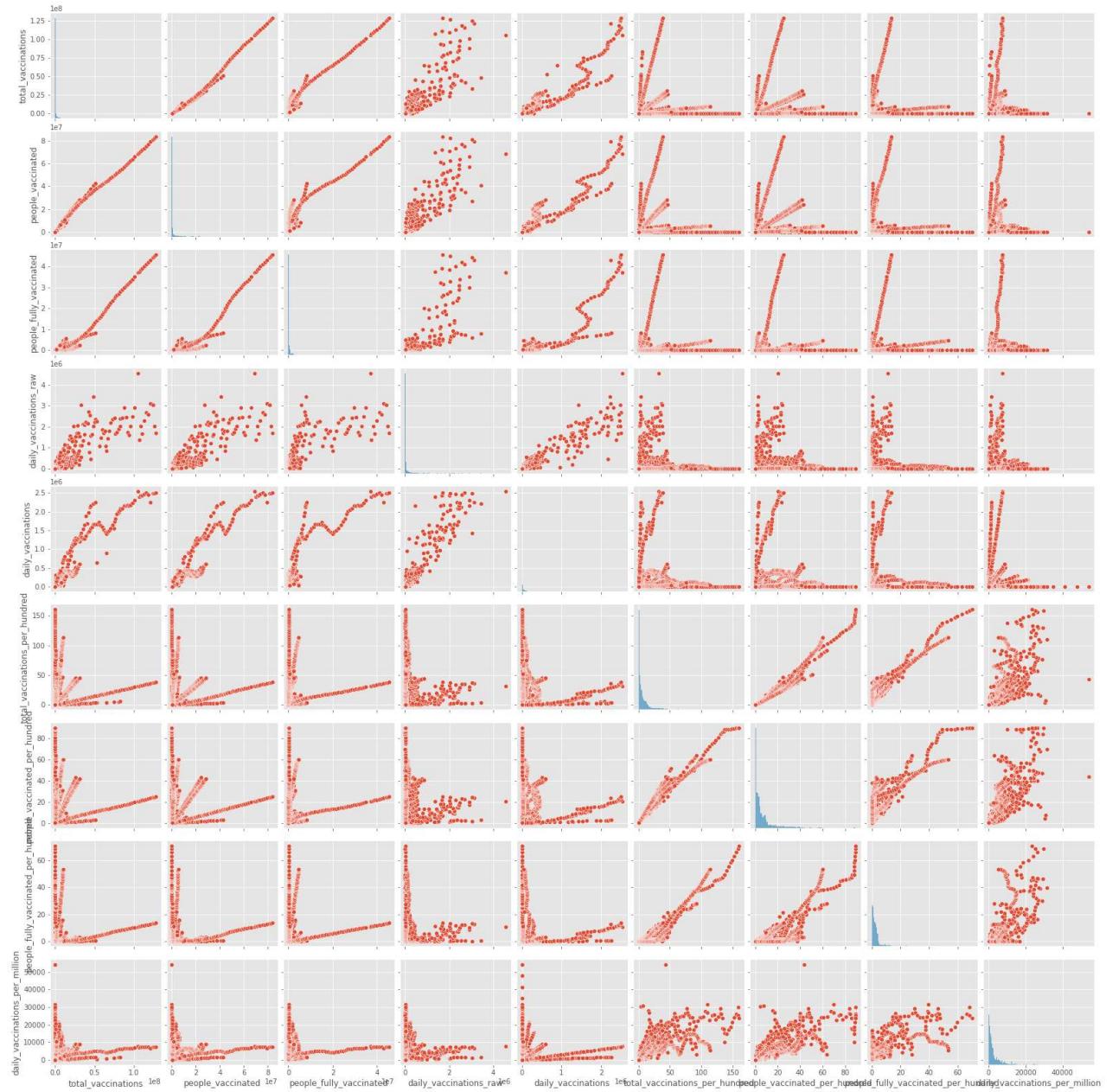
```
Out[97]: <AxesSubplot:xlabel='country', ylabel='source_name'>
```



Data Visualization of dataset

```
In [99]: sns.pairplot(df)
```

```
Out[99]: <seaborn.axisgrid.PairGrid at 0x10065310>
```



In [ ]: