

```
In [1]: # Data Analysis on country vaccination(2020-2021) using python
```

```
In [2]: import pandas as pd
import numpy as np
```

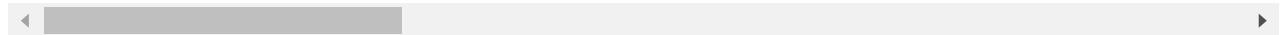
```
In [3]: df=pd.read_csv('C:/Users/SUBINDAS/Desktop/subin/country_vaccinations.csv')
```

```
In [4]: df
```

```
Out[4]:
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_v
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
3	Afghanistan	AFG	25-02-21	NaN	NaN	NaN	NaN
4	Afghanistan	AFG	26-02-21	NaN	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



```
In [5]: #showing first 3 rows with head()
```

```
In [6]: df.head(3)
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
0	Afghanistan	AFG	2021-02-21	22-	0.0	0.0	NaN
1	Afghanistan	AFG	2021-02-21	23-	NaN	NaN	NaN
2	Afghanistan	AFG	2021-02-21	24-	NaN	NaN	NaN

In [7]: # describe columns

In [8]: df.columns

```
Out[8]: 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')
```

In [9]: # Basic information about the dataset

In [10]: 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
```

In [11]: # view bottom 5 records

In [12]: df.tail(5)

Out[12]: country iso_code date total_vaccinations people_vaccinated people_fully_vaccinated daily_va

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

◀ ▶

In [13]: `# describe stastical information on data set`

In [14]: `df.describe()`

Out[14]:

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

◀ ▶

In [15]: `# To check the number of rows and columns in dataset`

In [16]: `df.shape`

Out[16]: (7897, 15)

In [17]: `# To check missing values count`

In [18]: `df.isnull().sum().values.sum()`

Out[18]: 27167

`# getting which column contains null`

In [19]:

In [21]: `df.isnull().any()`

```
Out[21]: 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
```

In [22]: `# To check for the null value count`In [23]: `df.isna().sum()`

```
Out[23]: 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
```

In [24]: `# Check the count of all country`In [25]: `df['country'].value_counts()`

```
Out[25]: England      100
          Canada       100
          Wales         100
          United Kingdom 100
          Scotland      100
          ...
          Taiwan         2
          Equatorial Guinea 1
          Laos           1
          Guinea          1
          Bahamas         1
          Name: country, Length: 151, dtype: int64
```

In [26]: `# Check the count of all date`In [27]: `df['date'].value_counts()`

```
Out[27]: 09-03-21    132  
10-03-21    132  
08-03-21    131  
07-03-21    131  
11-03-21    130  
...  
18-12-20     8  
15-12-20     8  
16-12-20     8  
14-12-20     6  
13-12-20     5  
Name: date, Length: 101, dtype: int64
```

```
In [28]: # Check the count of all vaccines
```

```
In [29]: df['vaccines'].value_counts()
```

```
Out[29]: 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
26
Moderna, Oxford/AstraZeneca
22
Pfizer/BioNTech, Sputnik V
11
Name: vaccines, dtype: int64
```

In [30]: # Qualitative Data Analysis

In [31]: # no 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)
```

In [33]: # no 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',
'Staten 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)
```

In [35]: # To find source_name maximum in dataset

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

Out[36]: 'Zimbabwe'

In [37]: # To find source_name minimum in dataset

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

Out[38]: 'Afghanistan'

```
In [39]: # To find vaccines maximum in dataset
```

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

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

```
In [41]: # To find vaccines minimum in dataset
```

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

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

```
In [43]: # Quantitative Analysis
```

```
In [44]: # countrywise vaccination
```

```
In [45]: df.groupby('country').sum(),['total']
```

```
Out[45]: (   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'])

```

In [46]: # Date wise vaccination

In [47]: df.groupby('date').sum(),['total']

```

Out[47]: (   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                                people_fully_vaccinated_per_hundred  daily_vaccinations_per_million
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                            ...                                 ...
30-01-21                            13.05                               12.87
30-12-20                            290.25                              224.62
31-01-21                            18.33                               18.12
31-02-21                            308.81                              232.35
31-12-20                            18.47                               17.57
.

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

```

In [48]: # Number of daily_vaccinations vaccinated by country

In [49]: print (df.groupby("country")["daily_vaccinations"].sum())

```

country
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

```

In [50]: # Number of people_vaccinated using by source_name

In [51]: print (df.groupby("source_name")["people_vaccinated"].sum())

```

source_name
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 [52]: # Statistical Analysis on Dataset

In [53]: # maximum value per column

In [54]: df.max()

Out[54]: 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

In [55]: # minimum value per column

In [56]: df.min()

Out[56]: 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 Covaxin, Oxford/AstraZeneca
 source_name COVID-19 Malta Public Health Response Team
 source_website <http://covid19.ncema.gov.ae/en>
 dtype: object

In [57]: # median value per column

In [58]: df.median()

Out[58]: 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

```
In [59]: # Standard deviation value per column
```

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

```
Out[60]: 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
```

```
In [61]: # Variance value per column
```

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

```
Out[62]: 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
```

```
In [63]: # Lower quartile/first quartile
```

```
In [64]: df.quantile(0.25)
```

```
Out[64]: 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
```

```
In [65]: # second quartile/median
```

```
In [66]: df.quantile(0.5)
```

```
Out[66]: 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
```

```
In [67]: # upper quartile
```

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

```
Out[68]: 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
```

In [69]: `# Interquartile range (IQR) on dataset`

In [70]: `df.quantile(0.75)-df.quantile(0.25)`

```
Out[70]: 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
```

In [71]: `# corelation values on dataset`

In [72]: `df.corr()`

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
total_vaccinations	1.000000	0.990179	0.947367	
people_vaccinated	0.990179	1.000000	0.893220	
people_fully_vaccinated	0.947367	0.893220	1.000000	
daily_vaccinations_raw	0.878029	0.885668	0.793317	
daily_vaccinations	0.942280	0.950743	0.861147	
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	

In [73]: `# covariance values on dataset`

In [74]: `df.cov()`

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
total_vaccinations	8.539110e+13	6.372826e+13	3.757908e+13	

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
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	

In [75]: # Data Visualization

In [76]: # Data Visualization of country by histogram

In [86]:

```
import matplotlib.pyplot as plt
import seaborn as sb
import matplotlib.pyplot as plt
import seaborn as sns
```

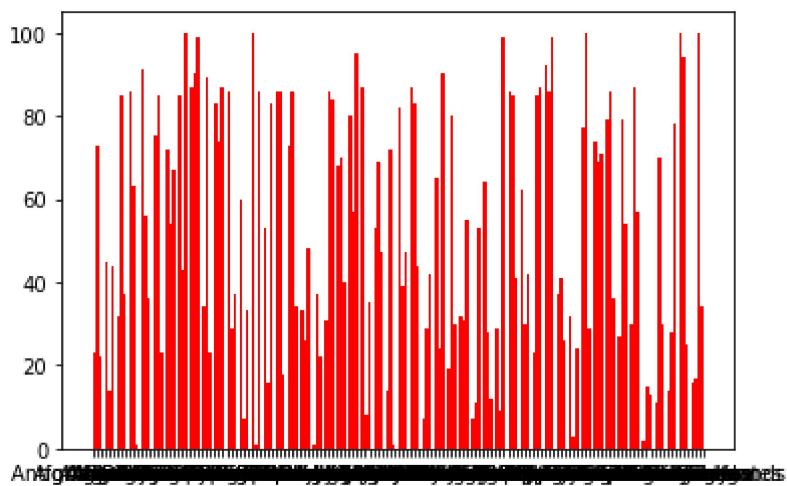
In [78]: plt.hist(df['country'], bins=200, color="red")

Out[78]: (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>

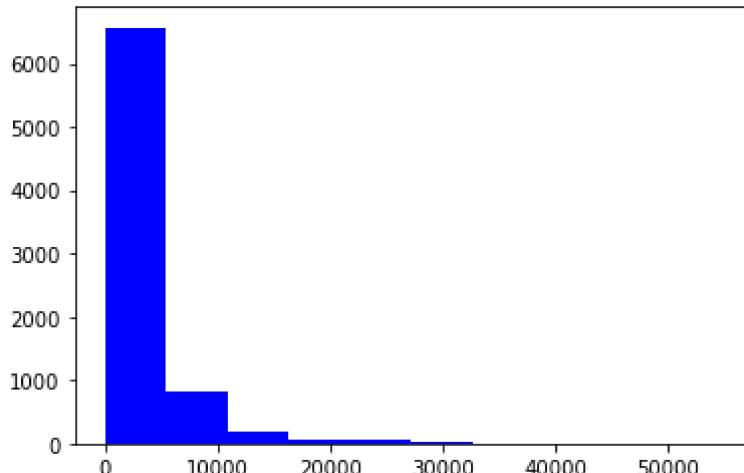
```



```
In [79]: # Data Visualization of daily_vaccinations_per_million by histogram
```

```
In [80]: plt.hist(df["daily_vaccinations_per_million"],bins=10,color="blue")
```

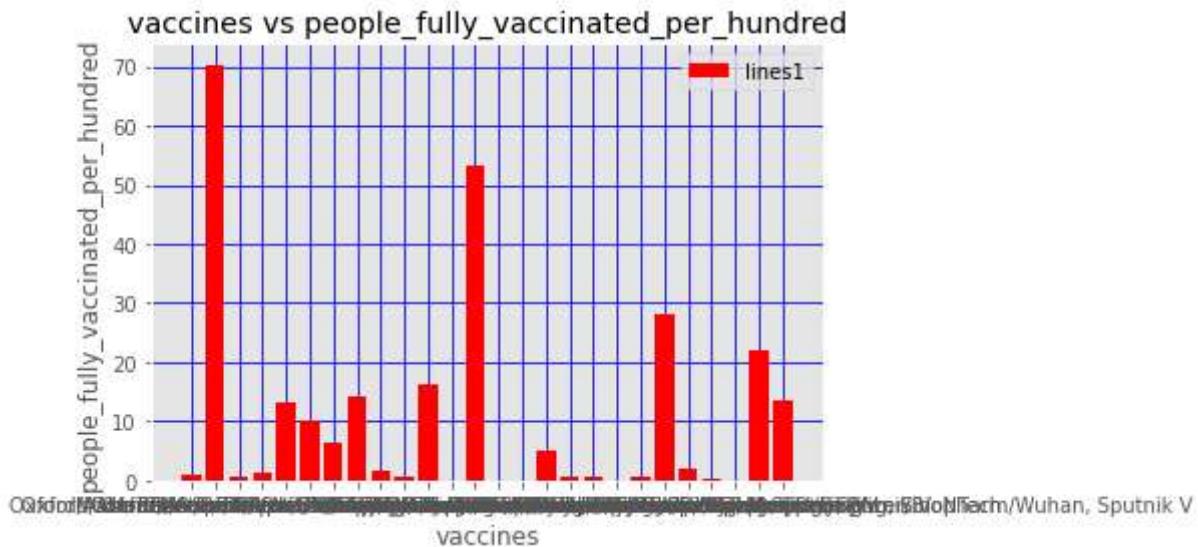
```
Out[80]: (array([6.564e+03, 8.240e+02, 1.990e+02, 5.800e+01, 4.200e+01, 1.100e+01,
       1.000e+00, 1.000e+00, 1.000e+00, 8.000e+00]),
 array([
        0. ,  5426.4, 10852.8, 16279.2, 21705.6, 27132. , 32558.4,
       37984.8, 43411.2, 48837.6, 54264. ]),
<BarContainer object of 10 artists>)
```



```
In [81]: from matplotlib import style
style.use('ggplot')
x=df['vaccines']
y=df['people_fully_vaccinated_per_hundred']
plt.bar(x,y,label='lines1',linewidth=20,color='red')
```

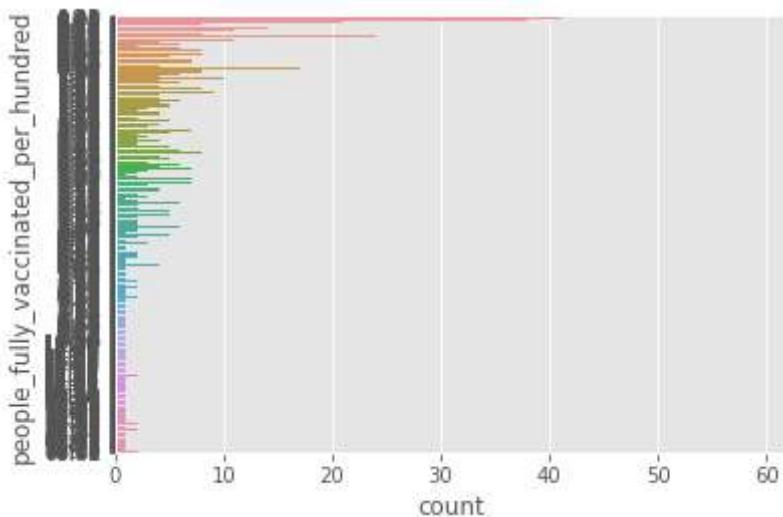
```
plt.title('vaccines vs people_fully_vaccinated_per_hundred')
plt.xlabel('vaccines')
plt.ylabel('people_fully_vaccinated_per_hundred')
plt.legend()
plt.grid(True, color='blue')
plt.show()
```

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.
fig.canvas.print_figure(bytes_io, **kw)



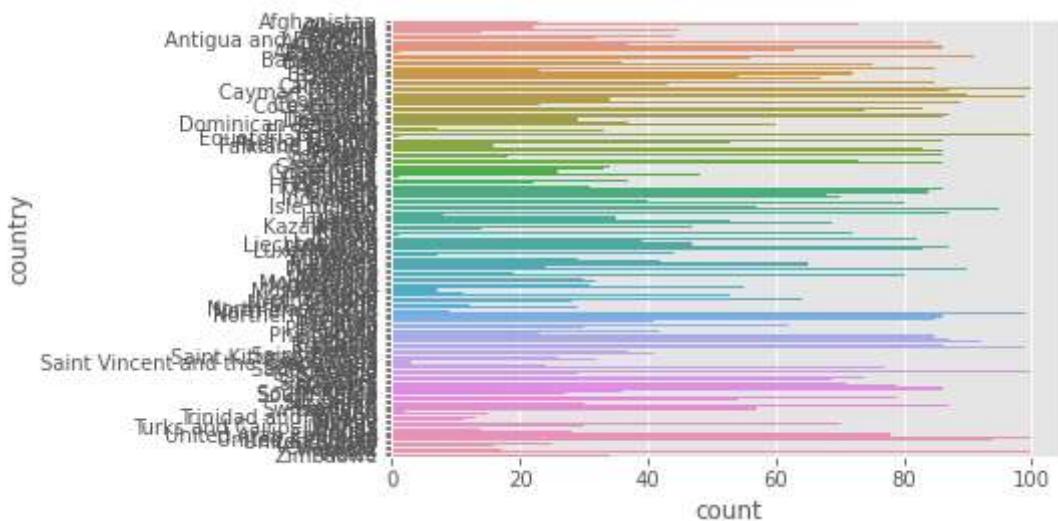
In [83]: `sb.countplot(y='people_fully_vaccinated_per_hundred', data=df)`

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



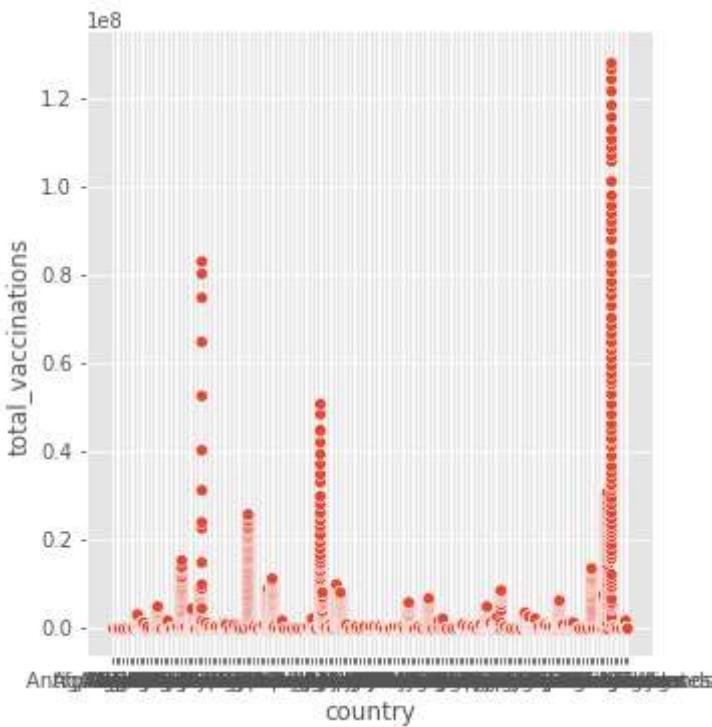
In [84]: `sb.countplot(y='country', data=df)`

Out[84]: <AxesSubplot:xlabel='count', ylabel='country'>



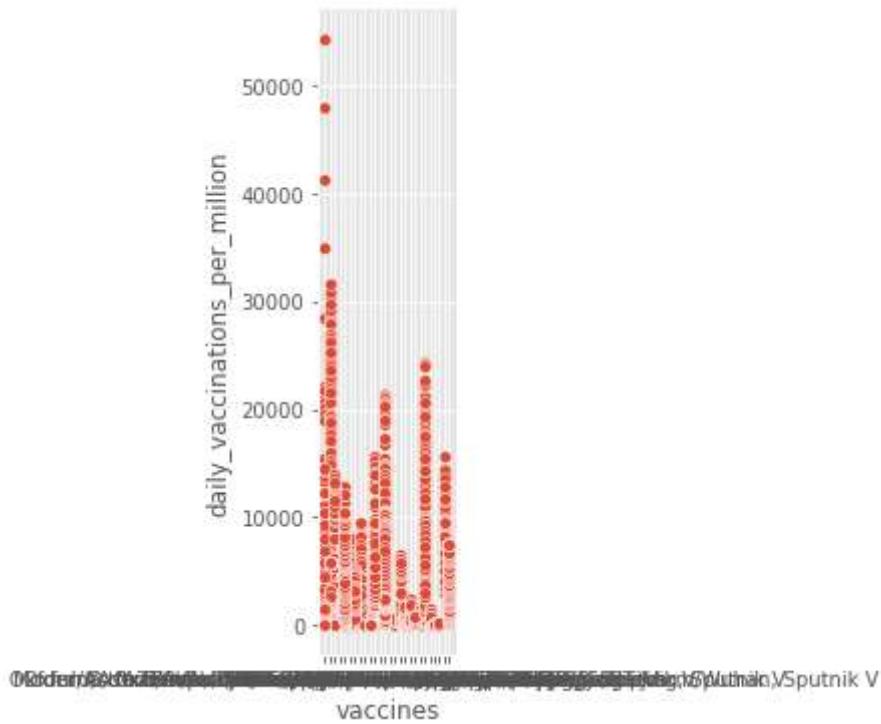
```
In [87]: sns.relplot(x="country",y="total_vaccinations",data=df)
```

```
Out[87]: <seaborn.axisgrid.FacetGrid at 0xb0f21a8>
```



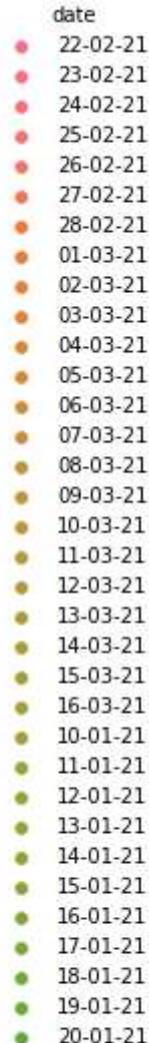
```
In [88]: sns.relplot(x="vaccines",y="daily_vaccinations_per_million",data=df)
```

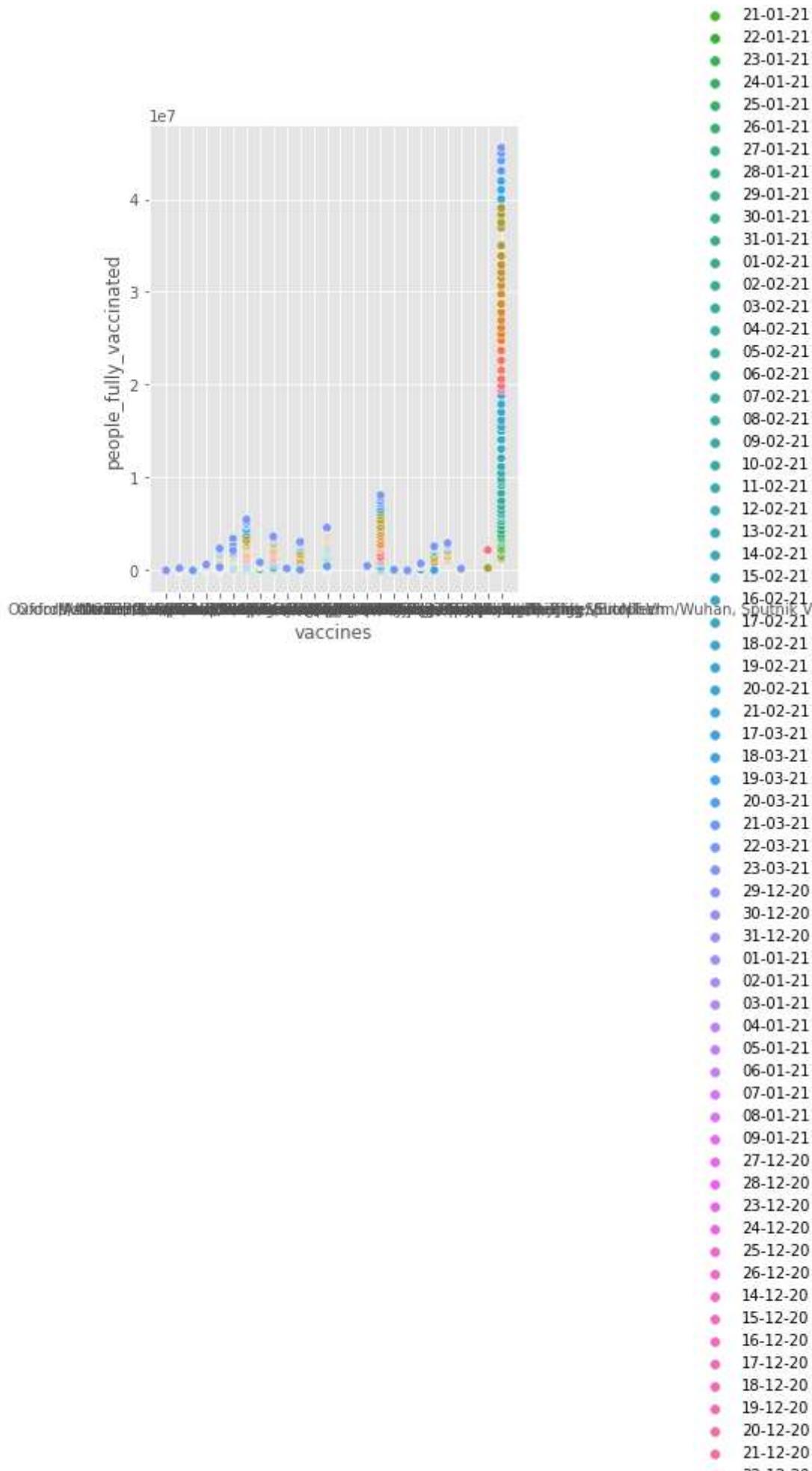
```
Out[88]: <seaborn.axisgrid.FacetGrid at 0x824e058>
```



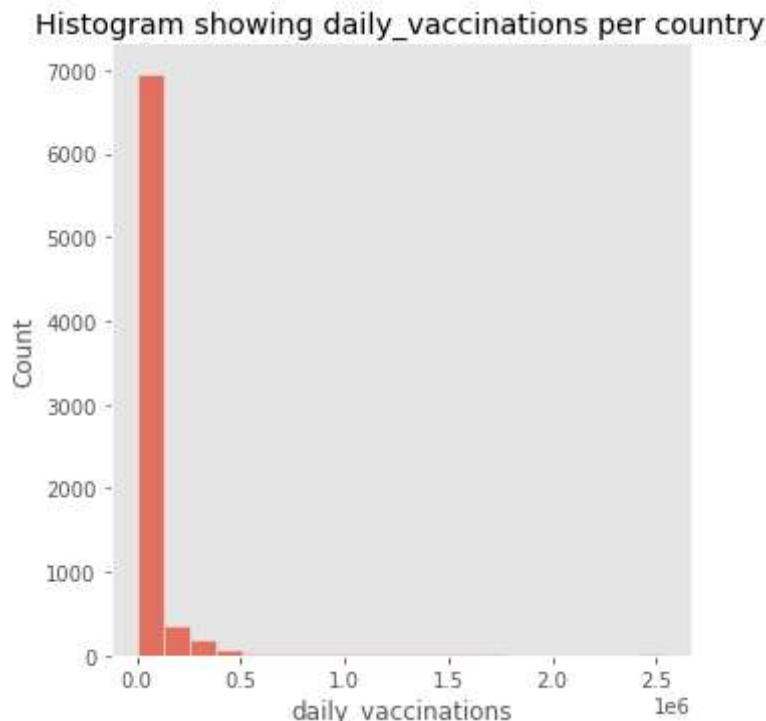
```
In [89]: sns.relplot(x="vaccines",y="people_fully_vaccinated",hue="date",data=df)
```

```
Out[89]: <seaborn.axisgrid.FacetGrid at 0xc372fe8>
```





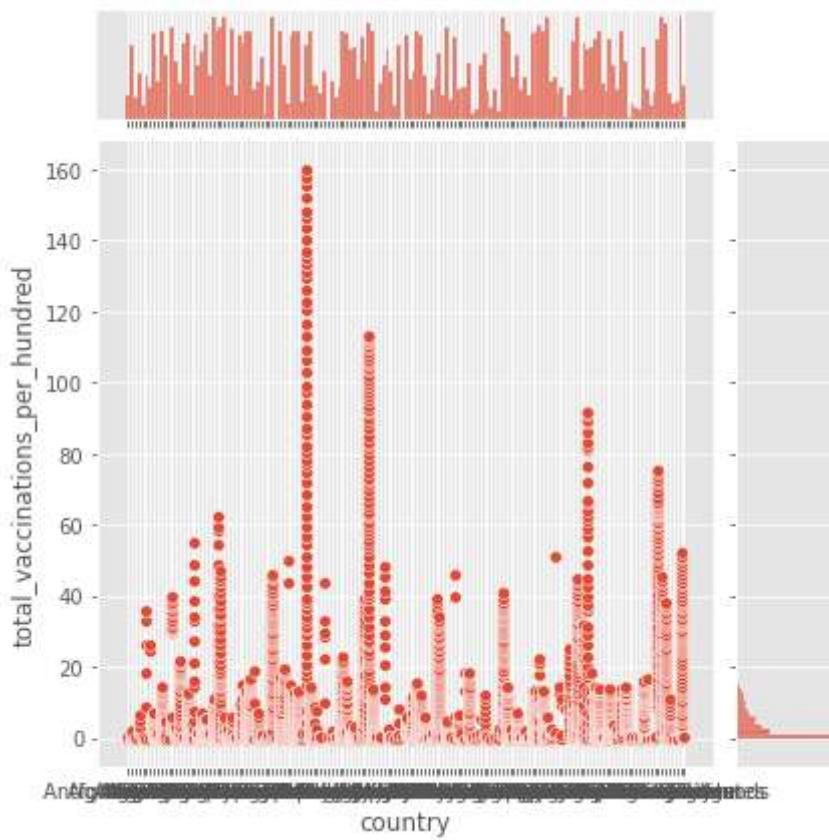
```
In [90]: sns.displot(df["daily_vaccinations"],bins=20)
plt.title("Histogram showing daily_vaccinations per country")
plt.grid()
plt.show()
```



```
In [91]: sns.jointplot(df["country"],df["total_vaccinations_per_hundred"])
```

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[91]: <seaborn.axisgrid.JointGrid at 0x6408490>
```

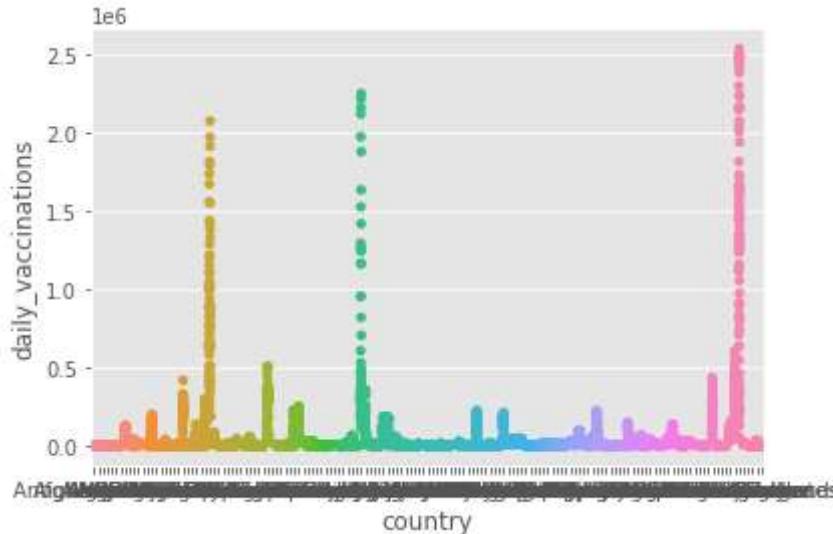


```
In [92]: sns.stripplot(df["country"],df["daily_vaccinations"])
```

```
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 positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
warnings.warn(
```

```
Out[92]: <AxesSubplot:xlabel='country', ylabel='daily_vaccinations'>
```

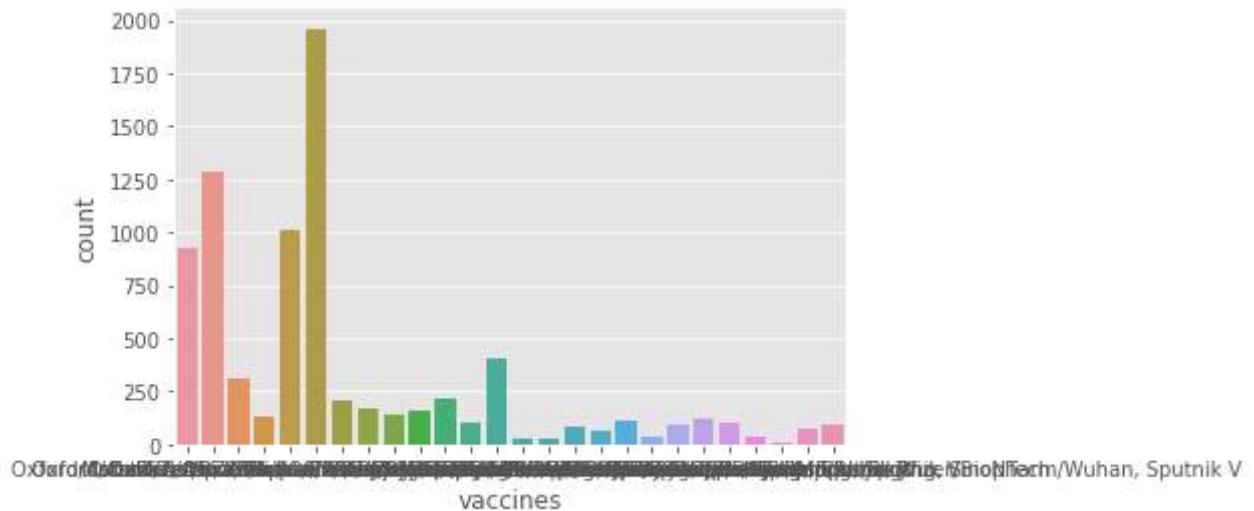


```
In [93]: sns.countplot(df["vaccines"])
```

```
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 w
```

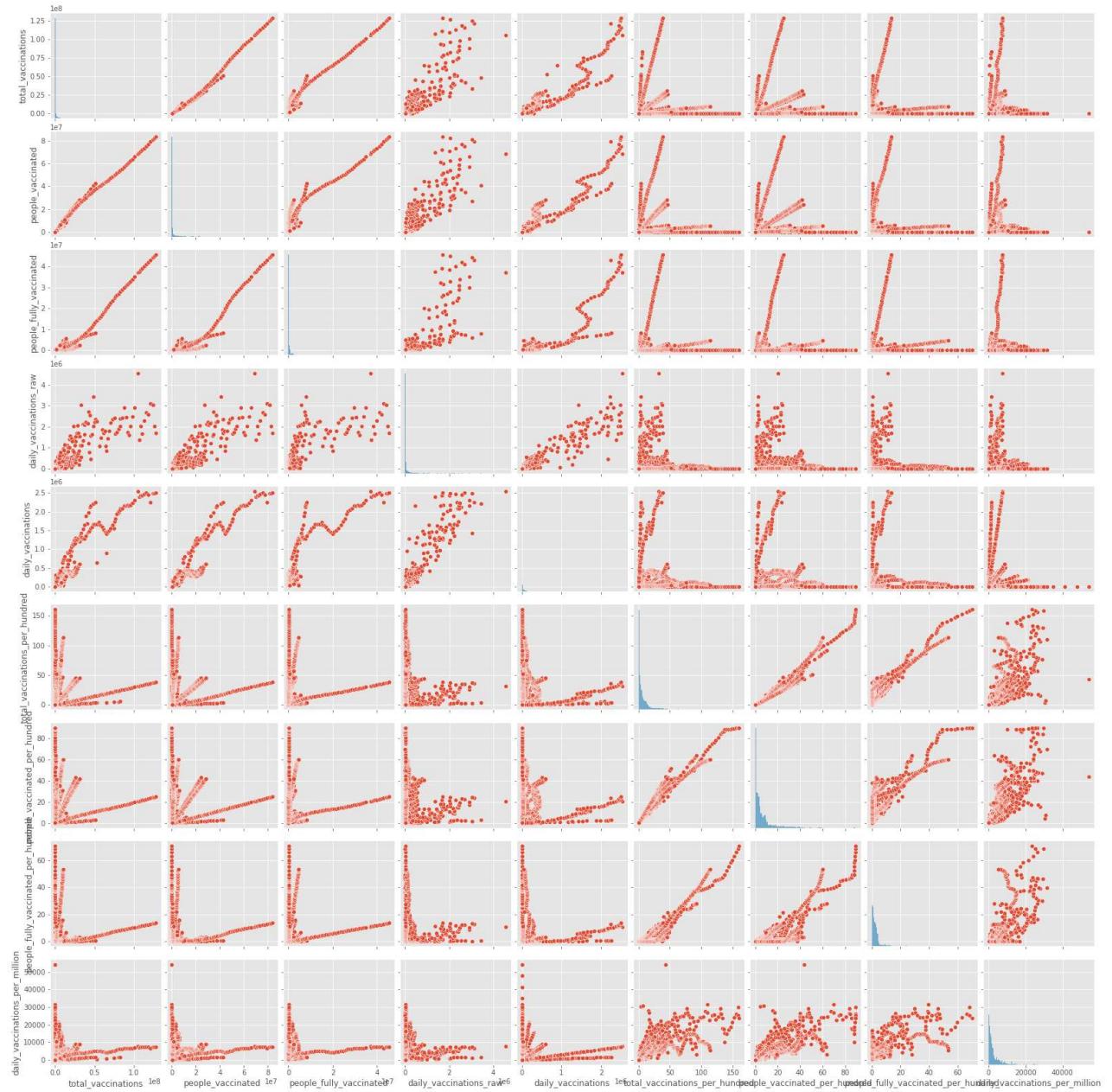
```
ill result in an error or misinterpretation.  
    warnings.warn(
```

```
Out[93]: <AxesSubplot:xlabel='vaccines', ylabel='count'>
```



```
In [97]: sb.pairplot(df)
```

Out[97]: <seaborn.axisgrid.PairGrid at 0xc372040>



In []:

In []:

In []: