COVID-19 VACCINES PROGRESS OVER THE WORLD

Coronaviruses are a group of related RNA viruses that cause diseases in mammals and birds. In humans and birds, they cause respiratory tract infections¶

COVID-19 vaccines are effective.They can keep you from getting and spreading the virus that causes covid-19

This dataset analysis is focused on summarizing how the COVID-19 vaccination is going around the world. More accurately, it is focused on answering the following questions:

- **How many types of vaccine are used? Where a specific vaccine is used in the world?**
- **How many people are vaccinated per hundred?**
- **How many people are fully and partially vaccinated**
- **Where are vaccinated more people per day?**
- **Where is the vaccination program more advanced? When will we have 25% of the population vaccinated?**

importing all the necessary libraries:

In [1]:

```
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('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv
/kaggle/input/covid-world-vaccination-progress/country_vaccinations.csv
/kaggle/input/vaccine/cgWhR4ArheoPsIKgoFRl.jpg
```

In [2]:

```
covid=pd.read_csv('/kaggle/input/covid-world-vaccination-progress/country_vaccinations.cs
v')
```

In [3]:

```
covid.shape
```

Out[3]:

```
(41919, 15)
```

OBSERVATION:

It has 40977 rows and 15 columns

In [4]:

```
covid.columns
```

Out[4]:

```
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')
      dtype='object')
```

OBSERVATION:

The 15 columns are 1)CountryCountry 2)ISO Code 3)Date 4)Total number of vaccinations 5)Total number of people vaccinated 6)Total number of people fully vaccinated 7)Daily vaccinations (raw) 8) Total vaccinations per hundred 9)Total number of people vaccinated per hundred 10)Total number of people fully vaccinated per hundred 11)Number of vaccinations per day 12)Daily vaccinations per million 13)Vaccines used in the country 14)Source name 15)Source website

In [5]:

```
covid.head
```

Out[5]:

```
<bound method NDFrame.head of            country iso_code        date  total_vaccinations
\
0      Afghanistan      AFG  2021-02-22                 0.0
1      Afghanistan      AFG  2021-02-23                 NaN
2      Afghanistan      AFG  2021-02-24                 NaN
3      Afghanistan      AFG  2021-02-25                 NaN
4      Afghanistan      AFG  2021-02-26                 NaN
...            ...      ...         ...                 ...
41914     Zimbabwe      ZWE  2021-08-30           4172657.0
41915     Zimbabwe      ZWE  2021-08-31           4219824.0
41916     Zimbabwe      ZWE  2021-09-01           4270430.0
41917     Zimbabwe      ZWE  2021-09-02           4323735.0
41918     Zimbabwe      ZWE  2021-09-03           4372216.0

       people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  \
0                    0.0                      NaN                     NaN
1                    NaN                      NaN                     NaN
2                    NaN                      NaN                     NaN
3                    NaN                      NaN                     NaN
4                    NaN                      NaN                     NaN
...                  ...                      ...                     ...
41914          2552273.0                1620384.0                 34804.0
41915          2582405.0                1637419.0                 47167.0
41916          2615233.0                1655197.0                 50606.0
41917          2649505.0                1674230.0                 53305.0
41918          2681657.0                1690559.0                 48481.0

       daily_vaccinations  total_vaccinations_per_hundred  \
0                     NaN                            0.00
1                  1367.0                             NaN
2                  1367.0                             NaN
3                  1367.0                             NaN
4                  1367.0                             NaN
...                   ...                             ...
41914             49092.0                           27.65
41915             47200.0                           27.96
41916             36416.0                           28.30
41917             39711.0                           28.65
41918             42317.0                           28.97

       people_vaccinated_per_hundred  people_fully_vaccinated_per_hundred  \
0                               0.00                                  NaN
1                                NaN                                  NaN
2                                NaN                                  NaN
3                                NaN                                  NaN
4                                NaN                                  NaN
...                              ...                                  ...
41914                          16.91                                10.74
41915                          17.11                                10.85
41916                          17.33                                10.97
41917                          17.56                                11.09
41918                          17.77                                11.20

       daily_vaccinations_per_million  \
0                                 NaN
```

```
1                                 34.0
2                                 34.0
3                                 34.0
4                                 34.0
...                                ...
41914                           3253.0
41915                           3127.0
41916                           2413.0
41917                           2631.0
41918                           2804.0

                                                 vaccines  \
0      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
1      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
2      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
3      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
4      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
...                                                  ...
41914  Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
41915  Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
41916  Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
41917  Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
41918  Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

                   source_name  \
0      World Health Organization
1      World Health Organization
2      World Health Organization
3      World Health Organization
4      World Health Organization
...                          ...
41914         Ministry of Health
41915         Ministry of Health
41916         Ministry of Health
41917         Ministry of Health
41918         Ministry of Health

                                           source_website
0      https://app.powerbi.com/view?r=eyJrIjoiYTkyM2V...
1      https://app.powerbi.com/view?r=eyJrIjoiYTkyM2V...
2      https://app.powerbi.com/view?r=eyJrIjoiYTkyM2V...
3      https://app.powerbi.com/view?r=eyJrIjoiYTkyM2V...
4      https://app.powerbi.com/view?r=eyJrIjoiYTkyM2V...
...                                                  ...
41914  https://www.arcgis.com/home/webmap/viewer.html...
41915  https://www.arcgis.com/home/webmap/viewer.html...
41916  https://www.arcgis.com/home/webmap/viewer.html...
41917  https://www.arcgis.com/home/webmap/viewer.html...
41918  https://www.arcgis.com/home/webmap/viewer.html...

[41919 rows x 15 columns]>
```

OBSERVATION:

It shows the first five rows of the file and all the 9 columns, in which some columns has null values.source of the website was also provided in the last column

Here we can observe that some columns has numerical values whereas some have strings

In [6]:

```
missing=covid.isnull().sum()
```

In [7]:

```
missing
```

Out[7]:

```
country                                     0
```

```
iso_code                                  0
date                                      0
total_vaccinations                    18848
people_vaccinated                     19921
people_fully_vaccinated               22828
daily_vaccinations_raw                22975
daily_vaccinations                      237
total_vaccinations_per_hundred        18848
people_vaccinated_per_hundred         19921
people_fully_vaccinated_per_hundred   22828
daily_vaccinations_per_million          237
vaccines                                  0
source_name                               0
source_website                            0
dtype: int64
```

OBSERVATION:

we can observe that country,iso_code, vaccines, source_name, source_website has no null values.

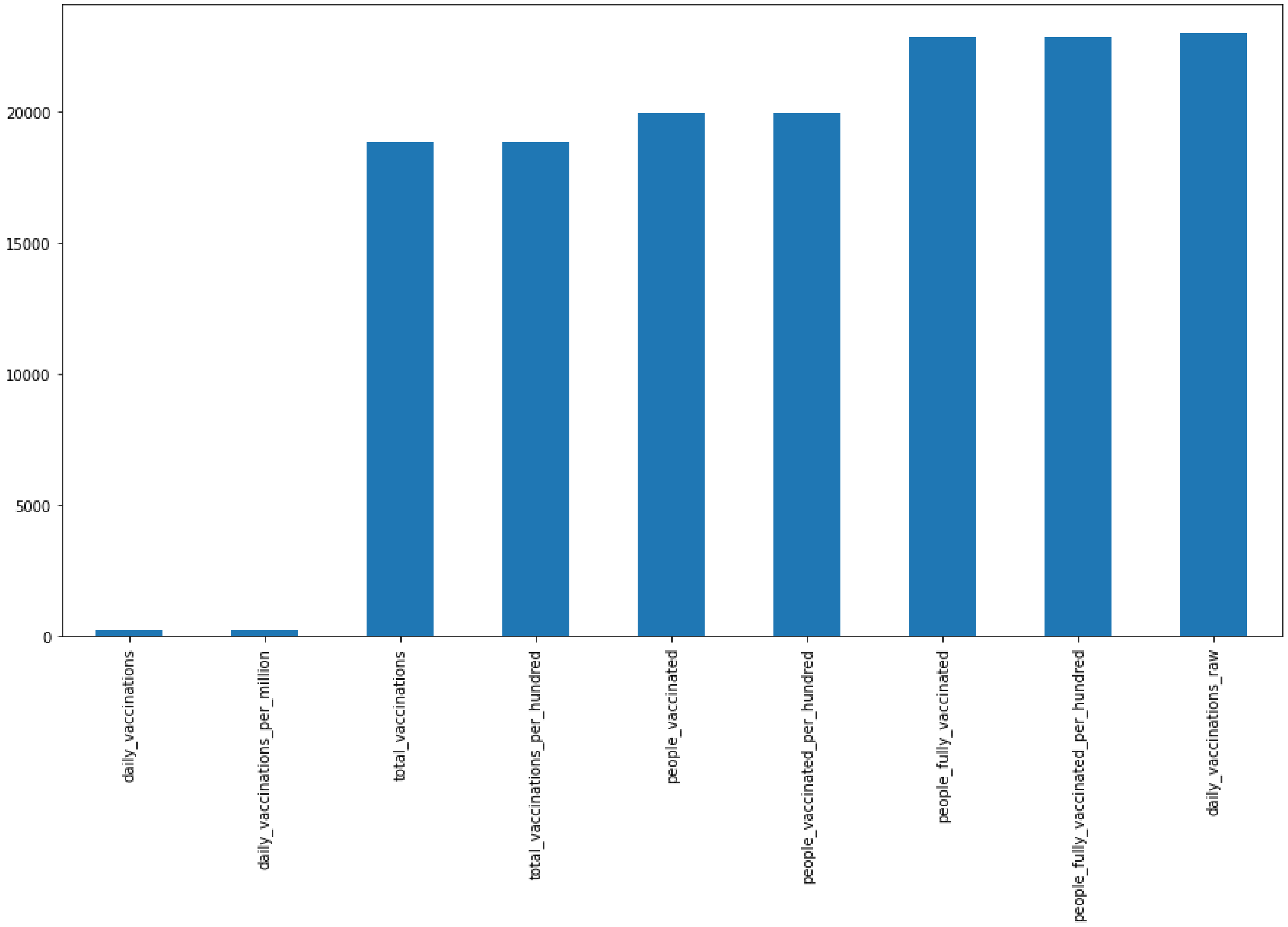
Remaining all the columns have null values. They need to be filled up with appropriate values later on

In [8]:

```
missing=missing[missing>0]
missing.sort_values(inplace=True)
plt.figure(figsize=(15,8))
missing.plot.bar()
```

Out[8]:

```
<AxesSubplot:>
```



In the above bar graph we can see that daily_vaccinations and daily_vaccinations_per_million has very less null values it's nearly in range 200-300 whereas people_fully_vaccinated, people_fully_vaccinated_per_hundred and

daily_vaccinations_raw has more null values

In [9]:

```
covid.dtypes.value_counts()
```

Out[9]:

```
float64    9
object     6
dtype: int64
```

OBSERVATION:

9 columns are Floating point numbers and 6 columns are object data type which is Text or mixed numeric and non-numeric values

In [10]:

```
covid['country'].describe()
```

Out[10]:

```
count       41919
unique        222
top        Norway
freq          275
Name: country, dtype: object
```

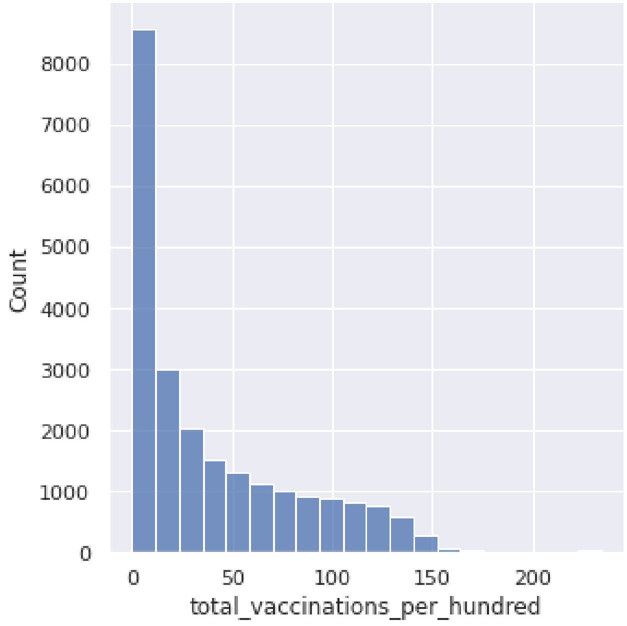
OBSERVATION:

This returns Different stats like count of values, unique values, top and frequency of occurences in this case. Hre the count of values are 40649 ,and it has 222 unique values, and frequency of occurences are 271

Histogram

In [11]:

```
sns.set(rc={'figure.figsize':(12,8)})
sns.displot(covid['total_vaccinations_per_hundred'], kde=False, bins=20) ;
```



OBSERVATION:

total vaccinations per hundred is the ratio (in percent) between population immunized and total population up to

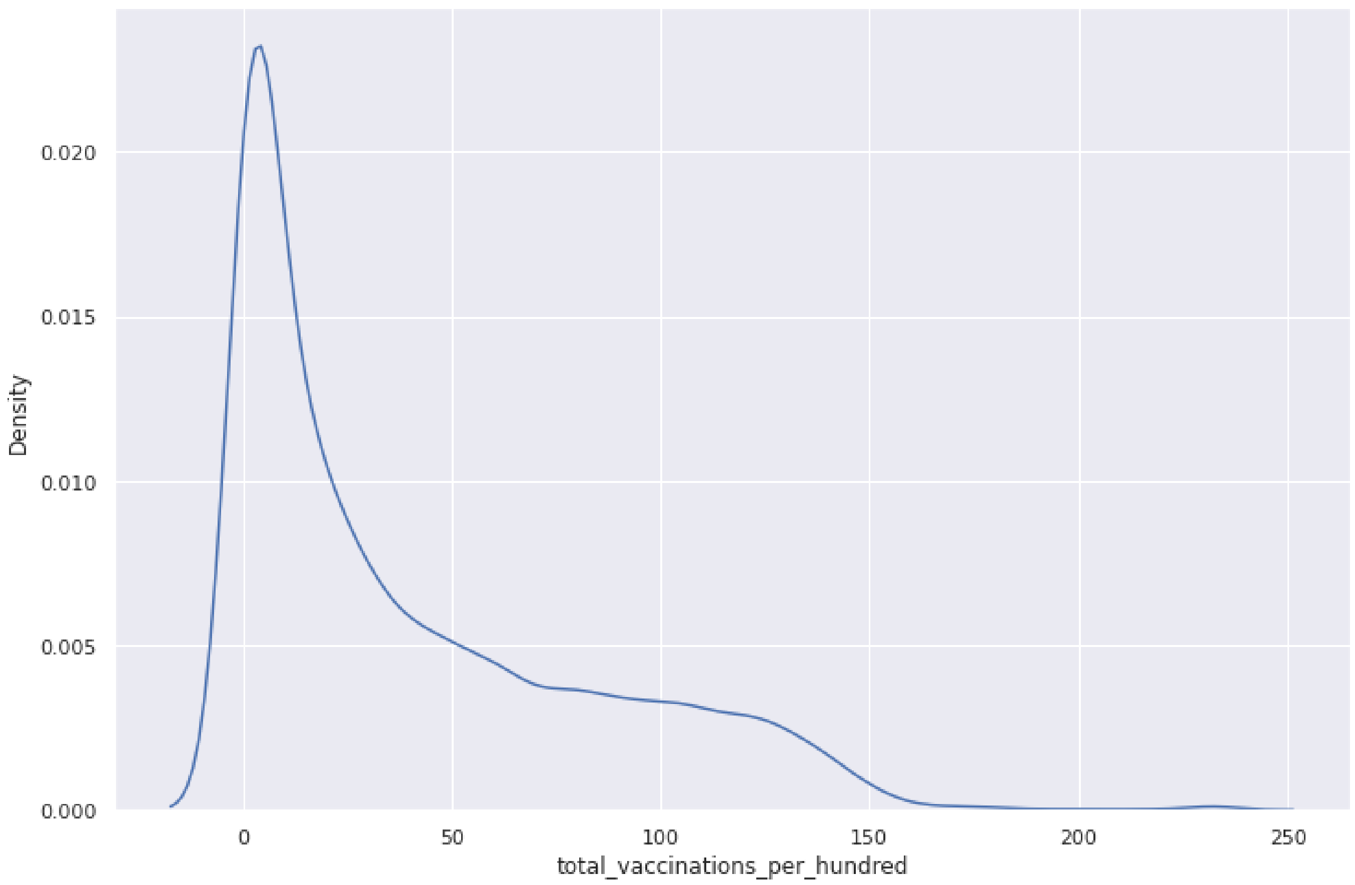
the date in the country. Here we can observe that very few countries have 150 vaccinations per hundred (I.e.half are fully vaccinated and half are partially vaccinated) this may due to less population and most of the countries has need of vaccine very badly

In [12]:

```
sns.kdeplot(covid['total_vaccinations_per_hundred'])
```

Out[12]:

```
<AxesSubplot:xlabel='total_vaccinations_per_hundred', ylabel='Density'>
```



OBSERVATION:

Here the peak point is between 0-5 and the tail part shoelws that very few countries have 150,200 vaccinations per hundred

In [13]:

```
covid['total_vaccinations_per_hundred'].describe()
```

Out[13]:

```
count    23071.000000
mean        40.334616
std         42.815281
min          0.000000
25%          4.690000
50%         23.160000
75%         66.500000
max        234.150000
Name: total_vaccinations_per_hundred, dtype: float64
```

OBSERVATION:

Different stats were returned like count of values, mean, mode, minimum value, maximum value and standard deviation etc

SCATTER PLOT

SCATTER PLOT

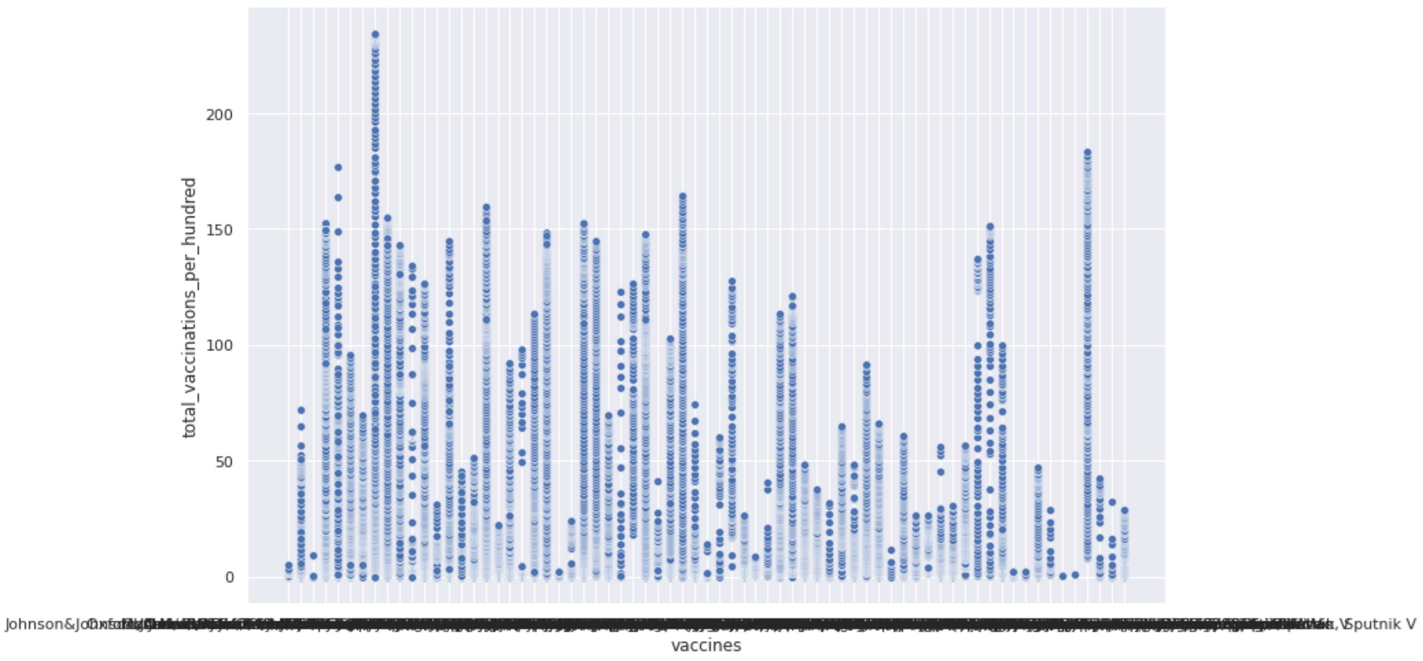
Scatter plots use a collection of points placed using Cartesian coordinates to display values from two variables. By displaying a variable in each axis, we can detect if a relationship or correlation between the two variables exists. Scatter Plots are also great for observing the spread of the data as they retain the exact data values and sample size.

In [14]:

```
sns.scatterplot(x='vaccines', y='total_vaccinations_per_hundred', data=covid)
```

Out[14]:

```
<AxesSubplot:xlabel='vaccines', ylabel='total_vaccinations_per_hundred'>
```



OBSERVATION:

This scatterplot, plots the vaccines on x axis and total number of vaccinations per hundred on years axis. Most of the countries have 50 vaccinations per hundred. only in few countries(roughly 1) people are fully vaccinated that may be due to very less population

CORRELATION

correlation can be calculated only on numerical columns we can't caluculate correlation on non-numeric

In [15]:

```
numeric_features = covid.select_dtypes(include = [np.number])
numeric_features.columns
```

Out[15]:

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

OBSERVATION:

Here numeric columns are stored in variable called number if features and columns are extracted. We can see that nearly 9 columns out of 15 have numerical values

In [16]:

```
numeric_features1 = covid.select_dtypes(exclude = ['O'])
numeric_features1.columns
```

Out[16]:

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

OBSERVATION:

Here numeric columns are extracted excluding strings

In [17]:

```
numeric_features.shape, numeric_features1.shape
```

Out[17]:

```
((41919, 9), (41919, 9))
```

OBSERVATION:

it gives number of numeric columns are there. We can get concluded that nearly 9 colums have numeric values

In [18]:

```
categorical_features = covid.select_dtypes(include = [np.object])
categorical_features.columns
```

Out[18]:

```
Index(['country', 'iso_code', 'date', 'vaccines', 'source_name',
       'source_website'],
      dtype='object')
```

OBSERVATION:

It gives all the string columns. We have nearly 5 string columns

it gives all the string columns.They are 1)country 2)iso code. 3)date 4)vaccines 5) source_name. 6) source website

In [19]:

```
categorical_features.shape
```

Out[19]:

```
(41919, 6)
```

OBSERVATION:

We can observe that we have 6 string columns

In [20]:

```
correlation = numeric_features.corr()
print(correlation['daily_vaccinations'].sort_values(ascending = False), '\n')
```

```
daily_vaccinations                       1.000000
daily_vaccinations_raw                   0.978422
people_vaccinated                        0.886501
```

```
total_vaccinations                       0.863934
people_fully_vaccinated                  0.673754
daily_vaccinations_per_million           0.116170
total_vaccinations_per_hundred           0.076263
people_vaccinated_per_hundred            0.071114
people_fully_vaccinated_per_hundred     -0.013562
Name: daily_vaccinations, dtype: float64
```

OBSERVATION:

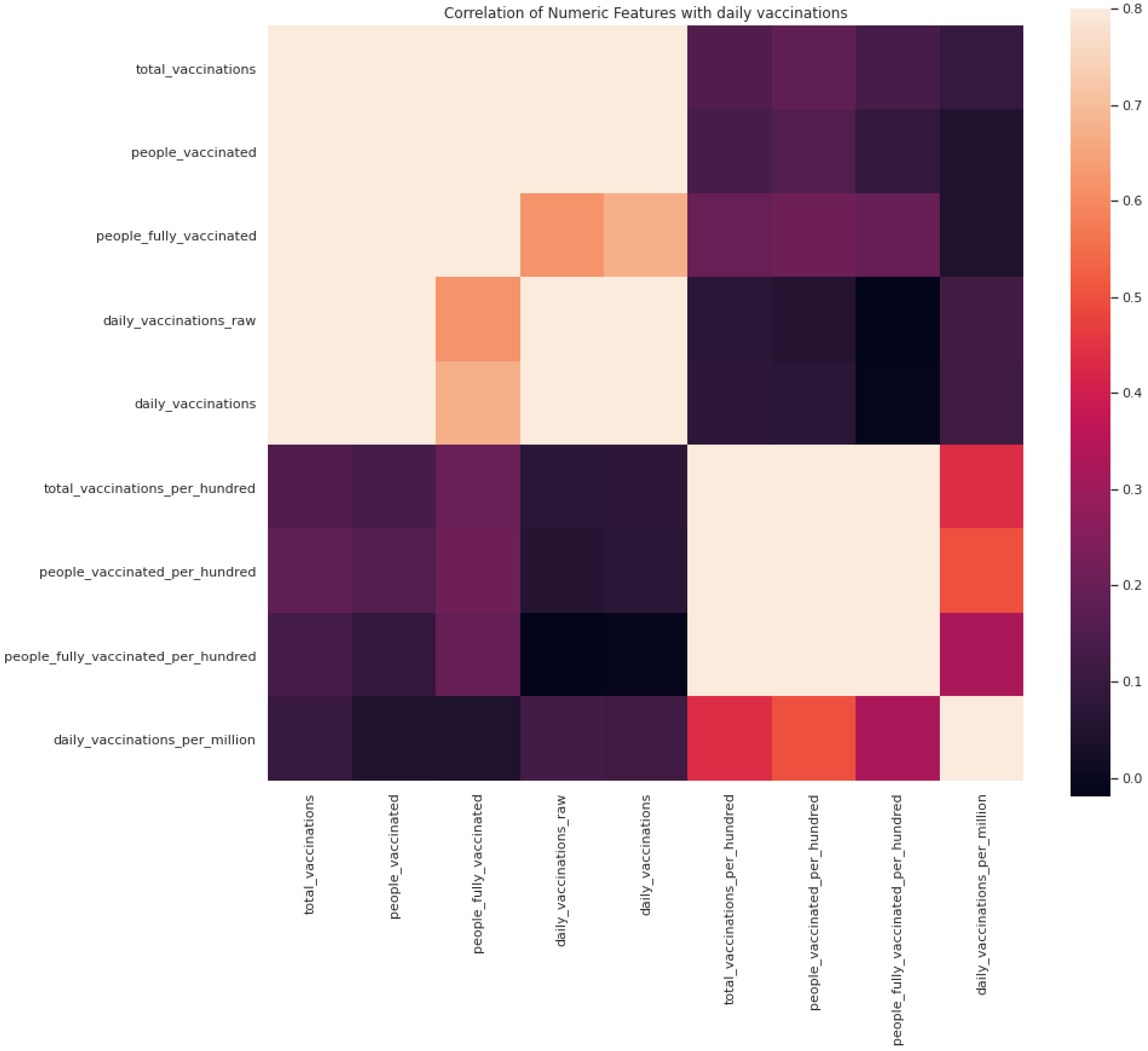
To find the correlation between numerical features we are using corr method

In [21]:

```
f, ax = plt.subplots(figsize = (14, 12))
plt.title('Correlation of Numeric Features with daily vaccinations')
sns.heatmap(correlation, square=True, vmax=0.8)
```

Out[21]:

```
<AxesSubplot:title={'center':'Correlation of Numeric Features with daily vaccinations'}>
```



OBSERVATION:

This is the correlation matrix for all the 9 numerical columns

In [22]:

```
k=5
cols = correlation.nlargest(k, 'daily_vaccinations')['daily_vaccinations'].index
print(cols)
```

```
Index(['daily_vaccinations', 'daily_vaccinations_raw', 'people_vaccinated',
       'total_vaccinations', 'people_fully_vaccinated'],
      dtype='object')
```

OBSERVATION:

We can conclude that top 5 numerical columns are daily_vaccinations, daily_vaccinations_raw, people _vaccinated, total_vaccinations, people_fully_vaccinated

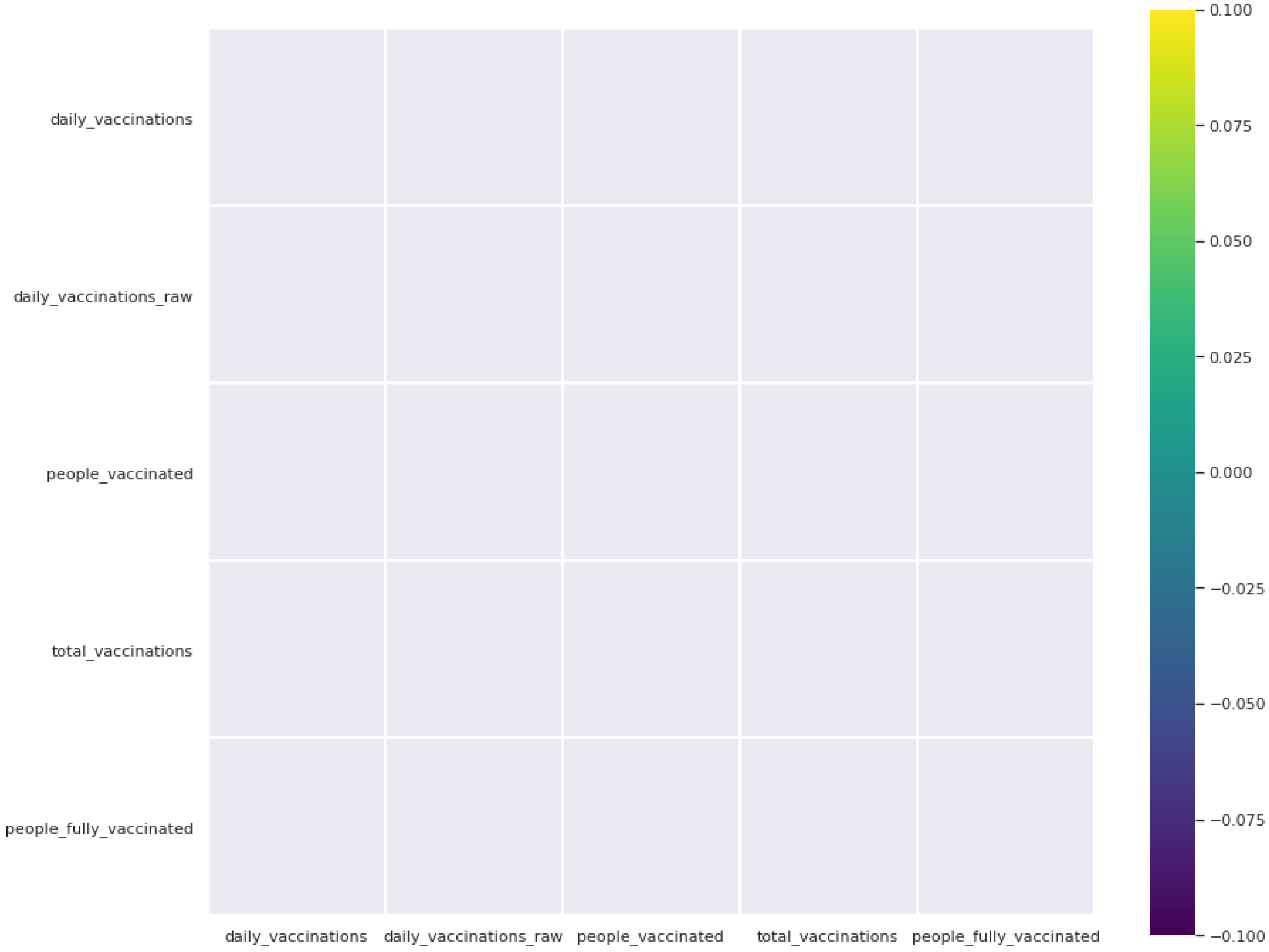
In [23]:

```
cm = np.corrcoef(covid[cols].values.T)
f, ax = plt.subplots(figsize = (14, 12))
sns.heatmap(cm, vmax=0.8, linewidths=0.01, square=True, annot=True, cmap='viridis', line
color='white', xticklabels=cols.values, yticklabels=cols.values)
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/matrix.py:194: RuntimeWarning: All-NaN sli
ce encountered
  vmin = np.nanmin(calc_data)
```

Out[23]:

```
<AxesSubplot:>
```



OBSERVATION:

Here correlation matrix for top 5 columns was plotted

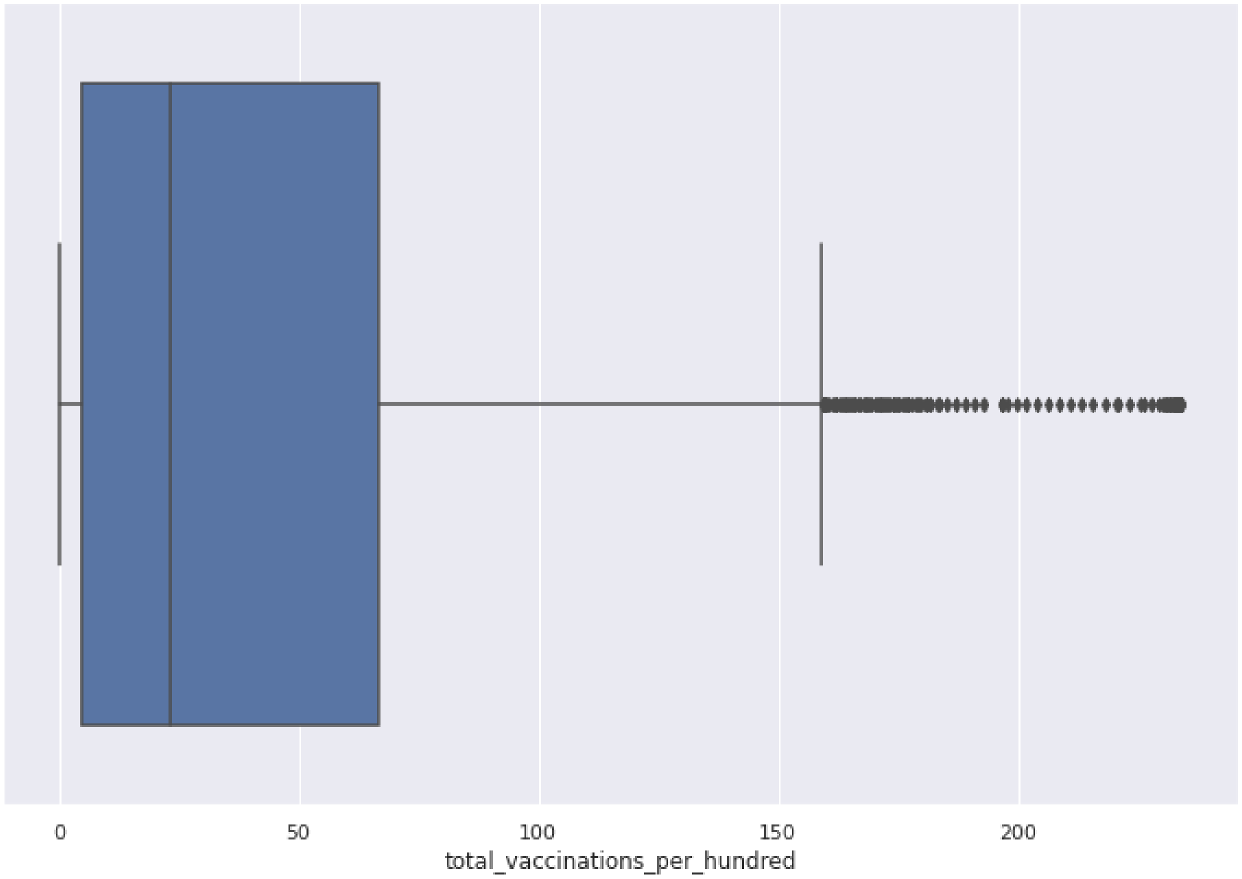
In [24]:

```
sns.boxplot(covid['total_vaccinations_per_hundred'])
```

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

Out[24]:

```
<AxesSubplot:xlabel='total_vaccinations_per_hundred'>
```

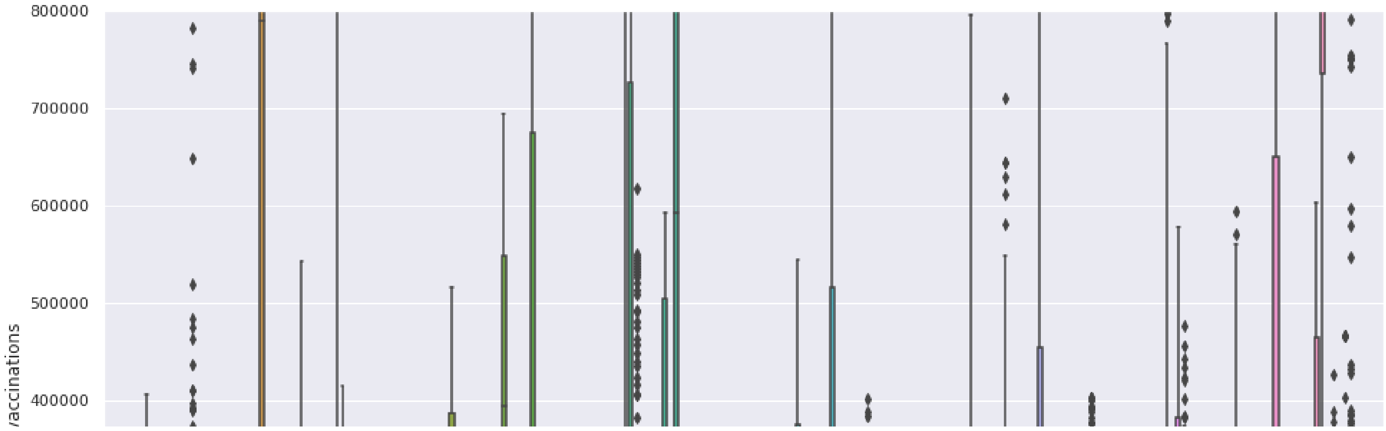


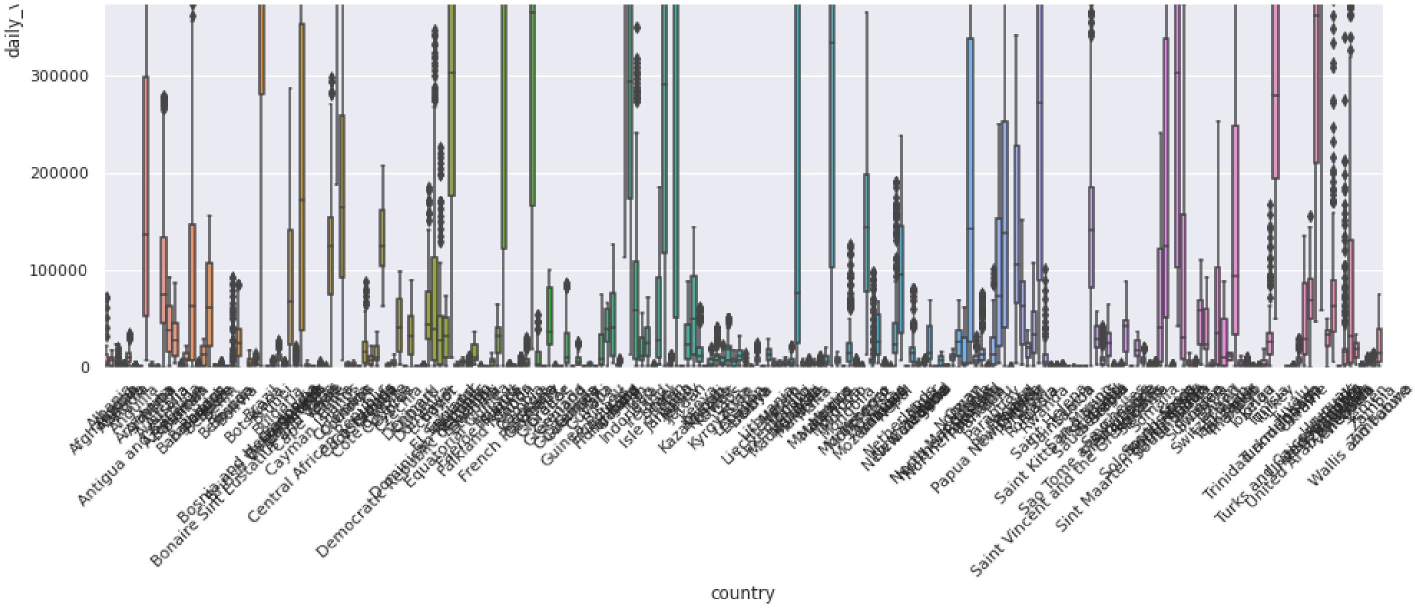
OBSERVATION:

It is the boxplot on total vaccinations per hundred and we can observe that median of total vaccinations per hundred is between 10-20.Here we can observe some outliers which are not fitting the box we can remove those outliers to reduce the difference from mean to median

In [25]:

```
f, ax = plt.subplots(figsize = (16,10))
fig = sns.boxplot(x='country', y='daily_vaccinations', data=covid)
fig.axis(ymin=0, ymax=800000)
xt = plt.xticks(rotation = 45)
```





OBSERVATION:

Here boxplot is plotted between country on x axis and daily_vaccinations on you axis

In [26]:

```
covid['country'].unique()
```

Out[26]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'Andorra', 'Angola',
       'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bermuda', 'Bhutan', 'Bolivia', 'Bonaire Sint Eustatius and Saba',
       'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'British Virgin Islands', 'Brunei', 'Bulgaria', 'Burkina Faso',
       'Cambodia', 'Cameroon', 'Canada', 'Cape Verde', 'Cayman Islands',
       'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
       'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', "Cote d'Ivoire",
       'Croatia', 'Cuba', 'Curacao', 'Cyprus', 'Czechia',
       'Democratic Republic of Congo', 'Denmark', 'Djibouti', 'Dominica',
       'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador', 'England',
       'Equatorial Guinea', 'Estonia', 'Eswatini', 'Ethiopia',
       'Faeroe Islands', 'Falkland Islands', 'Fiji', 'Finland', 'France',
       'French Polynesia', 'Gabon', 'Gambia', 'Georgia', 'Germany',
       'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada',
       'Guatemala', 'Guernsey', 'Guinea', 'Guinea-Bissau', 'Guyana',
       'Haiti', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Isle of Man', 'Israel',
       'Italy', 'Jamaica', 'Japan', 'Jersey', 'Jordan', 'Kazakhstan',
       'Kenya', 'Kiribati', 'Kosovo', 'Kuwait', 'Kyrgyzstan', 'Laos',
       'Latvia', 'Lebanon', 'Lesotho', 'Liberia', 'Libya',
       'Liechtenstein', 'Lithuania', 'Luxembourg', 'Macao', 'Madagascar',
       'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Mauritania',
       'Mauritius', 'Mexico', 'Moldova', 'Monaco', 'Mongolia',
       'Montenegro', 'Montserrat', 'Morocco', 'Mozambique', 'Myanmar',
       'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'New Caledonia',
       'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue',
       'North Macedonia', 'Northern Cyprus', 'Northern Ireland', 'Norway',
       'Oman', 'Pakistan', 'Palestine', 'Panama', 'Papua New Guinea',
       'Paraguay', 'Peru', 'Philippines', 'Pitcairn', 'Poland',
       'Portugal', 'Qatar', 'Romania', 'Russia', 'Rwanda', 'Saint Helena',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and the Grenadines', 'Samoa', 'San Marino',
       'Sao Tome and Principe', 'Saudi Arabia', 'Scotland', 'Senegal',
       'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
       'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia',
       'Solomon Islands', 'Somalia', 'South Africa', 'South Korea',
       'South Sudan', 'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Sweden',
       'Switzerland', 'Syria', 'Taiwan', 'Tajikistan', 'Tanzania',
       'Thailand', 'Timor', 'Togo', 'Tokelau', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
       'Turks and Caicos Islands', 'Tuvalu', 'Uganda', 'Ukraine',
       'United Arab Emirates', 'United Kingdom', 'United States',
       'Uruguay', 'Uzbekistan', 'Vanuatu', 'Venezuela', 'Vietnam',
       'Wales', 'Wallis and Futuna', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

OBSERVATION:

We can observe the array of all the countries that has unique names

In [27]:

```
covid['country'].nunique()
```

Out[27]:

```
222
```

OBSERVATION:

Here we can observe that nearly 222 countries had unique names

In [28]:

```
covid['country'].value_counts()
```

Out[28]:

```
Norway                                275
Latvia                                273
England                               269
Scotland                              269
Canada                                264
                                     ...
Madagascar                             48
Haiti                                  43
Tanzania                               21
Turkmenistan                            1
Bonaire Sint Eustatius and Saba         1
Name: country, Length: 222, dtype: int64
```

OBSERVATION:

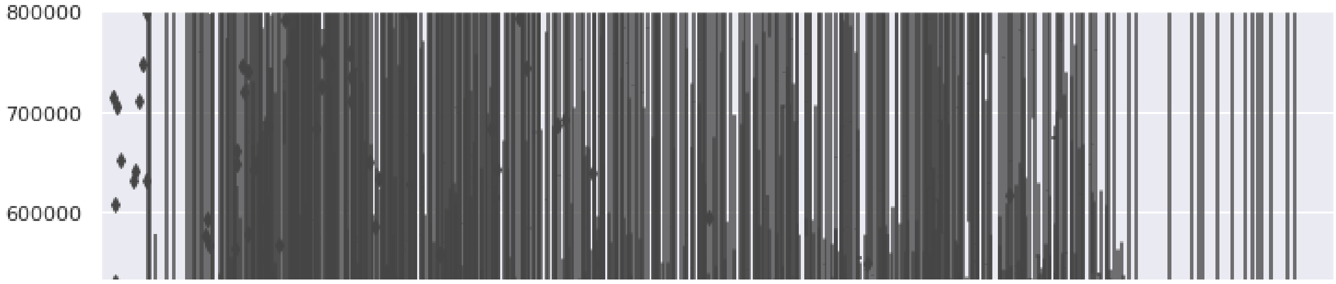
it gives value counts for all the 222 countries.we can observe that Norway has the highest count that is 271 and Bonaire Sint Eustatius and saba,and Turkmenistan has least count I.e. 1

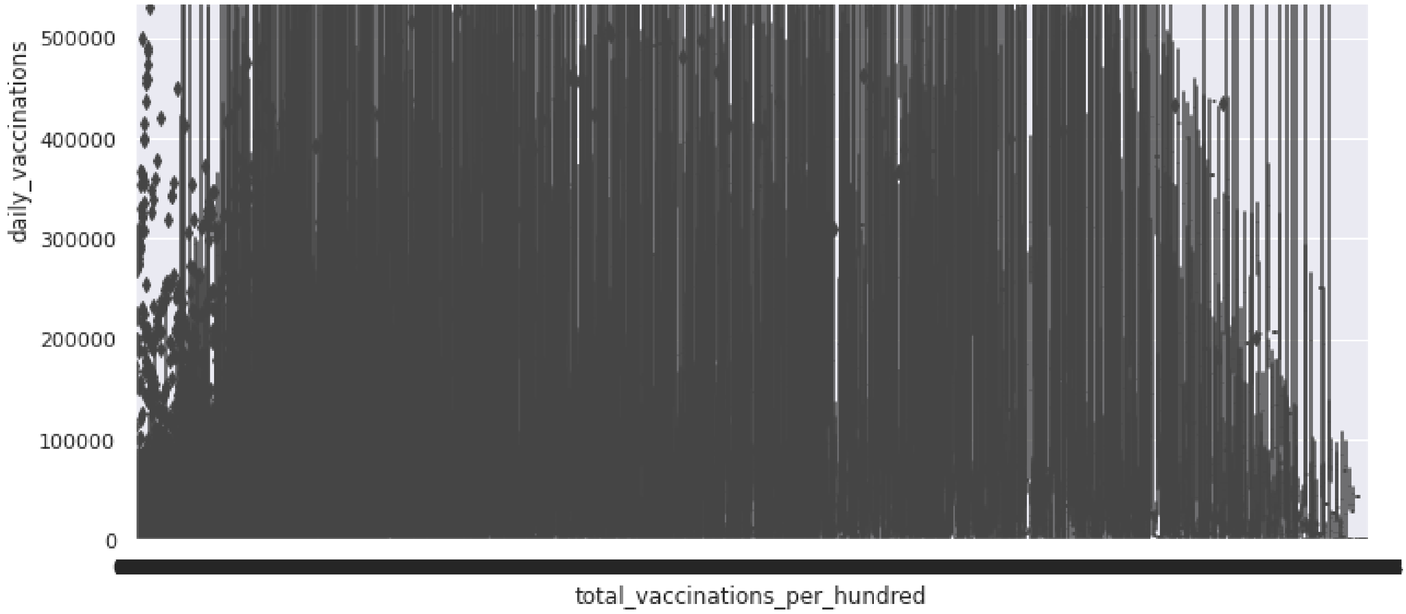
In [29]:

```
f, ax = plt.subplots(figsize = (12,8))
fig = sns.boxplot(x='total_vaccinations_per_hundred', y='daily_vaccinations', data=covid
)
fig.axis(ymin=0, ymax=800000)
```

Out[29]:

```
(-0.5, 9533.5, 0.0, 800000.0)
```





In [30]:

```
covid['total_vaccinations_per_hundred'].value_counts()
```

Out[30]:

```
0.00      244
0.01       95
0.02       67
0.06       56
0.04       51
         ...
31.36       1
139.48      1
17.70       1
141.91      1
55.06       1
Name: total_vaccinations_per_hundred, Length: 9534, dtype: int64
```

CONCLUSION:

Vaccination program in all over the world is going at a high rate. From this dataset we can conclude that, In some countries, they cover there population at a higher rate as comparisons to other countries while in some country the program is started in a month.In some undeveloped country, it is still not started but few people got the vaccinated (may be due to high profile).maybe in the near future we will witness day by day improvements when it comes to the availability of the vaccine and the number of people vaccinated in a day. Also, the fact that more types of vaccines are tested, accepted and used in the world can strengthen our hope to go back to a normal life.

The above graphs shows how slowly but surely, the vaccines are being administered in increasingly large numbers each day. If we look carefully, we can also identify a slight downward trend in the number of new cases each day, as the vaccinations progress. Humanity is on its way to victory!hi

COVID-19 has taken a heavy toll on mankind. We have lost far too many people and suffered too much for too long. Now is the time to fight back. Let 2021 be the year we reclaim what 2020 took from us. Regardless of what people might say, always wear a mask when out in public and maintain social distancing. DO NOT give in hearsay! Only when all of the graphs plotted inthe innumerable notebooks posted by the talented people on Kaggle point heavily in our favour, having dwarfed the damage this pandemic has already done, will be be able to call it a victory.

Until then, take care, don’t forget to wear a mask and hold on, because the end of this pandemic may be closer than we imagine. I hope the notebook was insightful, and as I am new to Kaggle, I would really appreciate some feedback

In [ ]: