Naan Mudhalvan IBM project

Applied DataScience(Phase 4- Development)

Topic- covid 19 Vaccine Analysis

By: Sushthi. R(au411521104115)

3.1 Dataset and its detail explanation implementation

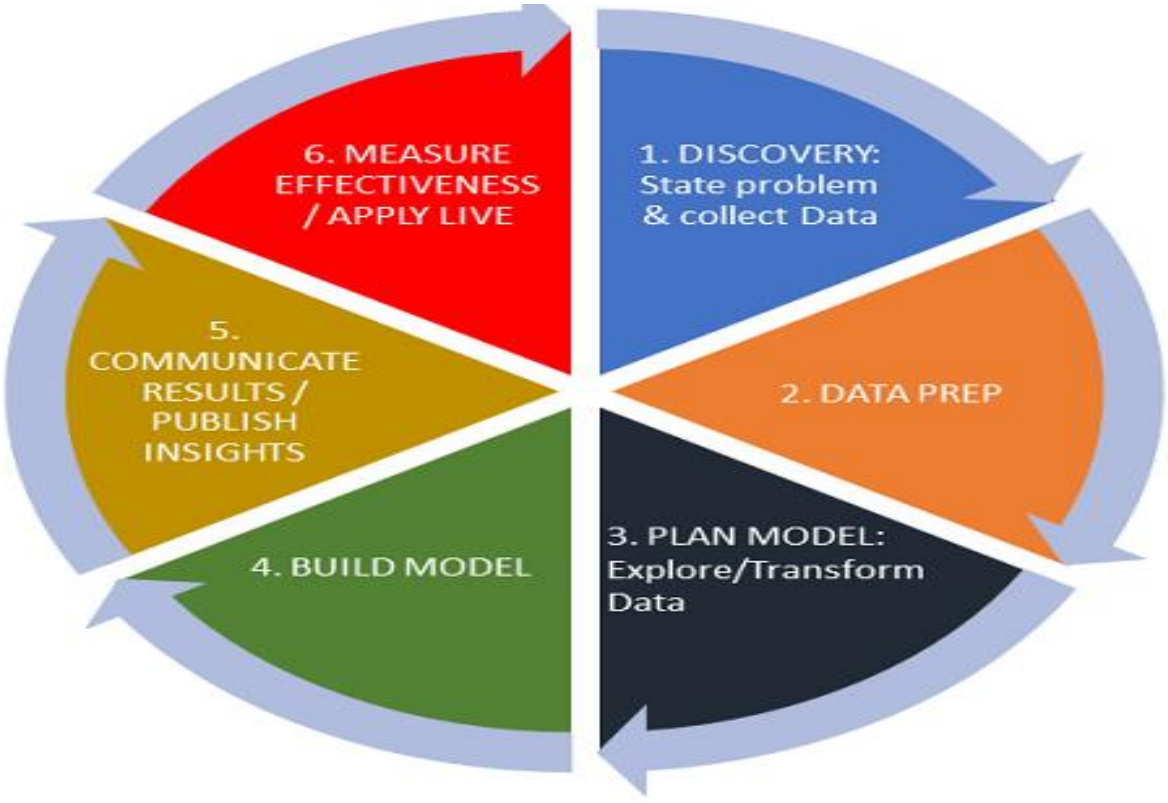
3.1.1 Problem Statement:

This Project mainly aims to find out the trend of the vaccinations around the world for the prevention of the Covid 19 pandemic and how much has been achieved so far.

3.1.2 Design Thinking:

The design thinking process consists of five stages: empathize, define, ideate, prototype, and test. Each step needs to be given appropriate resources and the proper duration to create an end product that reliably meets user needs.

3.1.3 Phase of Development:

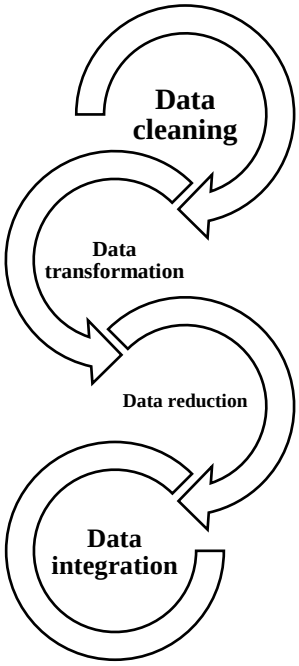


Dataset link: https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress

Data description:

Country level vaccination data is gathered and assembled in one single file. Then, this data file is merged with locations data file to include vaccination sources information. A second file, with manufacturers information, is included.

3.1.4 Data Preprocessing steps:



3.1.5 Model training process

3.1.5.1 Elastic net (ENET):

Elastic Net (ENET) is a penalized linear regression model that incorporates both the L1 and L2 penalties. Combining the L1-norm (lasso) and L2-norm (ridge) penalties, ENET decreases the regression coefficients. ENET arose from criticism of LASSO (Least Absolute Shrinkage and Selection Operator), a variable selection

algorithm that is excessively dependent on data and hence unstable. To obtain the best of both techniques is to mix the penalties of ridge regression and lasso. ENET mathematical equations are as follows:

$$E_{enet}\left(\widehat{\beta}\right) = \frac{\sum i = 1^{n}\left(y_i - x_i\widehat{\beta}\right)^2}{2n} + \gamma\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\widehat{\beta_j}^{\,2} + \alpha\sum_{j=1}^{m}\left|\widehat{\beta_j}\right|\right)$$

3.1.5.2 CUBIST:

Cubist is a rule-based model derived from Quinlan's M5 model tree. Linear regression models are embedded in the terminal leaves of a tree. The predictors used in earlier splits have been utilized to create these models. At each branch of the tree, there are also intermediate linear models. At the tree's terminal node, a prediction is created using the linear regression model, but it is "smoothed" by taking into consideration the preceding node's prediction (which also occurs recursively up the tree). The tree is simplified to a collection of rules, which are originally pathways from top to bottom. CUBIST has the following mathematical equation:

$$C_{cubist} = (1-a) \times \rho\,(p) + a \times \rho(c)$$

where is the current model forecast and is the parent model prediction positioned above it in the tree.

3.1.5.3 Gaussian process (GAUSS)

The Gaussian Processes (GAUSS) model is a probabilistic machine learning framework that is often used for regression and classification issues [31]. The GAUSS model may make predictions based on past data and provide confidence ranges for those predictions. The Gaussian processes model [32] is an approach developed by scientist and statistician. The following are the GAUSS mathematical procedures:

The following is a multivariate Gaussian regression function:

$$P(f|X) = \aleph(f|\mu, k)$$

3.6 Performance measures

Three metrics are used to evaluate prediction performance of daily COVID-19 vaccination: Mean Absolute Scaled Error (MASE), Relative Absolute Error (RAE), Mean Squared Log Error (MSLE).

MASE is given a:

$$\frac{1}{n}\sum_{n=1}^{n}\left(\frac{\left|y_t^n - \hat{y}_t^n\right|}{\frac{1}{n-m}\sum_{n=m+1}^{n}\left|y_t^n - y_{t-m}^n\right|}\right)$$

RAE is defined as follows:

$$\frac{\sqrt{\sum_{n=1}^{n}\left(y_t^n - \hat{y}_t^n\right)^2}}{\sqrt{\sum_{n=1}^{n} y_t^{n\,2}}}$$

MSLE is defined as follows:

$$\frac{1}{n}\sum_{n=0}^{n}(\log{(y_t^n+1)}-\log(\hat{y}_t^n+1))^2$$

Exploratory data analysis:

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

Import libraries:

importing all the required libraries like pandas, NumPy, matplotlib, plotly, seaborn, and word cloud that are required for data analysis. Check the below code to import all the required libraries.

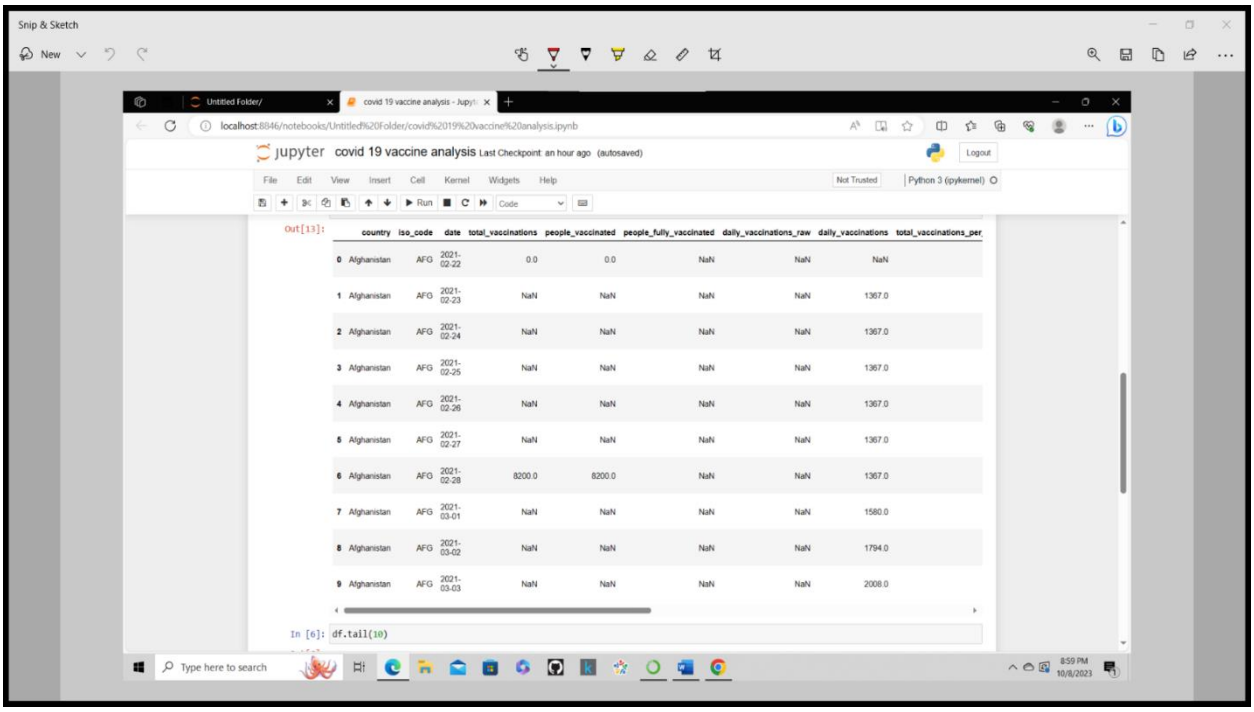
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.patches as mpatches
from plotly.subplots import make_subplots
from wordcloud import WordCloud
import seaborn as sns
sns.set(color_codes = True)
sns.set(style="whitegrid")
import plotly.figure_factory as ff
from plotly.colors import n_colors
```

READ DATA AND BASIC INFORMATION

Read the CSV file using pandas read_csv() function and show the output using head() function.

df=pd.read_csv("country_vaccinations.csv")

df.head(10)



info() function is used to get the overview of data like data type of feature, a number of null values in each column, and many more.

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              86512 non-null  object
 1   iso_code                             86512 non-null  object
 2   date                                 86512 non-null  object
 3   total_vaccinations                   43607 non-null  float64
 4   people_vaccinated                    41294 non-null  float64
```

```
 5   people_fully_vaccinated              38802 non-null  float64
 6   daily_vaccinations_raw               35362 non-null  float64
 7   daily_vaccinations                   86213 non-null  float64
 8   total_vaccinations_per_hundred       43607 non-null  float64
 9   people_vaccinated_per_hundred        41294 non-null  float64
 10  people_fully_vaccinated_per_hundred  38802 non-null  float64
 11  daily_vaccinations_per_million       86213 non-null  float64
 12  vaccines                             86512 non-null  object
 13  source_name                          86512 non-null  object
 14  source_website                       86512 non-null  object
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
```

```
df.fillna(value = 0, inplace = True)
df.total_vaccinations = df.total_vaccinations.astype(int)
df.people_vaccinated = df.people_vaccinated.astype(int)
df.people_fully_vaccinated = df.people_fully_vaccinated.astype(int)
df.daily_vaccinations_raw = df.daily_vaccinations_raw.astype(int)
df.daily_vaccinations = df.daily_vaccinations.astype(int)
df.total_vaccinations_per_hundred = df.total_vaccinations_per_hundred.astype(int)
df.people_fully_vaccinated_per_hundred =
df.people_fully_vaccinated_per_hundred.astype(int)
df.daily_vaccinations_per_million = df.daily_vaccinations_per_million.astype(int)
df.people_vaccinated_per_hundred =
df.people_vaccinated_per_hundred.astype(int)
date = df.date.str.split('-', expand =True)
date
```

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

86512 rows × 3 columns

data.isnull().sum()

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

Statistical analysis:

Statistical analysis is the process of collecting and analyzing large volumes of data in order to identify trends and develop valuable insights.

Explore the mean, min, max

vaccinations_df.mean()

```
total_vaccinations                 2.315117e+07
people_vaccinated                  8.451007e+06
people_fully_vaccinated            6.341251e+06
daily_vaccinations_raw             1.106083e+05
daily_vaccinations                 1.308517e+05
total_vaccinations_per_hundred     4.041962e+01
people_vaccinated_per_hundred      1.953547e+01
people_fully_vaccinated_per_hundred    1.593274e+01
daily_vaccinations_per_million     3.245792e+03
year                               2.021199e+03
month                              6.165711e+00
day                                1.571936e+01
dtype: float64
```

vaccinations_df.min()

```
country
Afghanistan
iso_code
ABW
```

```
date                                                                2020-
12-02 00:00:00
total_vaccinations
0.0
people_vaccinated
0.0
people_fully_vaccinated
0.0
daily_vaccinations_raw
0.0
daily_vaccinations
0.0
total_vaccinations_per_hundred
0.0
people_vaccinated_per_hundred
0.0
people_fully_vaccinated_per_hundred
0.0
daily_vaccinations_per_million
0.0
vaccines                            Abdala, Johnson&Johnson, Oxford/Ast
raZeneca, P...
source_name                         Africa Centres for Disease Control
and Prevention
source_website                      http://103.247.238.92/webportal/pag
es/covid19-...
year
2020
month
1
day
1
dtype: object
```

vaccinations_df.max()

```
country
Zimbabwe
iso_code
ZWE
date                                                                2022-
03-29 00:00:00
total_vaccinations
3263129000.0
people_vaccinated
1275541000.0
```

```
people_fully_vaccinated
1240777000.0
daily_vaccinations_raw
24741000.0
daily_vaccinations
22424286.0
total_vaccinations_per_hundred
345.37
people_vaccinated_per_hundred
124.76
people_fully_vaccinated_per_hundred
122.37
daily_vaccinations_per_million
117497.0
vaccines                                                  Sinopharm/Beij
ing, Sputnik V
source_name                                                  World Healt
h Organization
source_website                          https://www.ssm.gov.mo/docs/19164/1
9164_dd2dfe...
year
2022
month
12
day
31
dtype: object
```

vaccinations_df.country.value_counts()

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

vaccinations_df.country

```
0        Afghanistan
1        Afghanistan
2        Afghanistan
```

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

Visualization:

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Barplot visualization of top countries with most vaccinations

x=df.groupby("country")["Total_vaccinations(count)"].mean().sort_values(ascending= False).head(20)

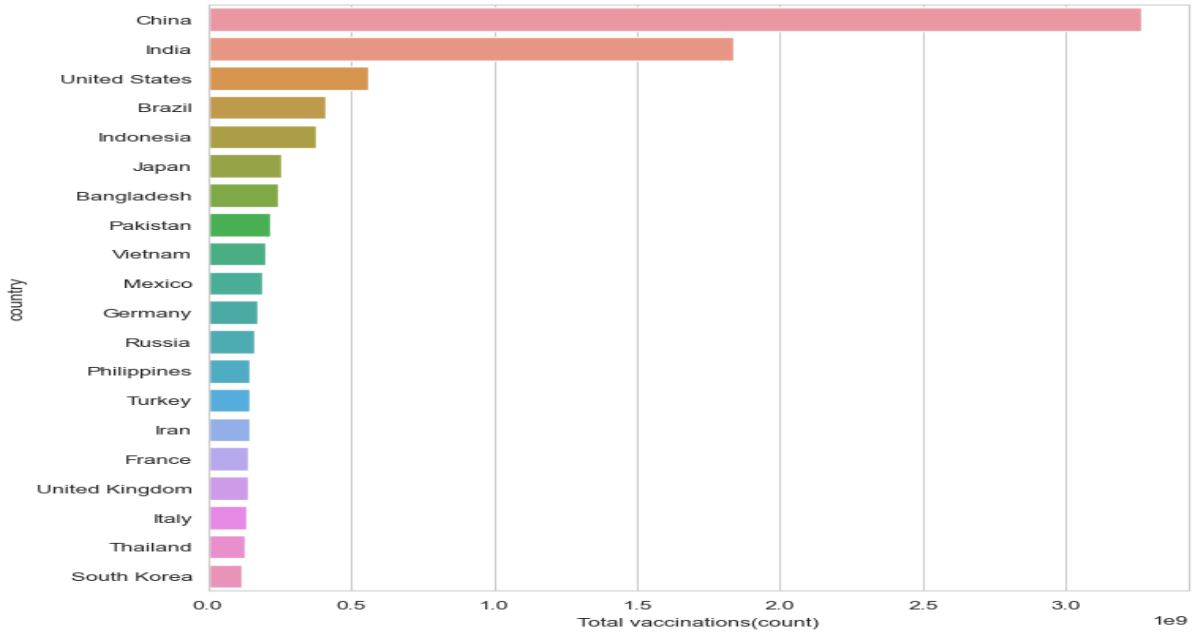
sns.set_style("whitegrid")

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

ax= sns.barplot(x.values,x.index)

ax.set_xlabel("Total vaccinations(count)")

plt.show()



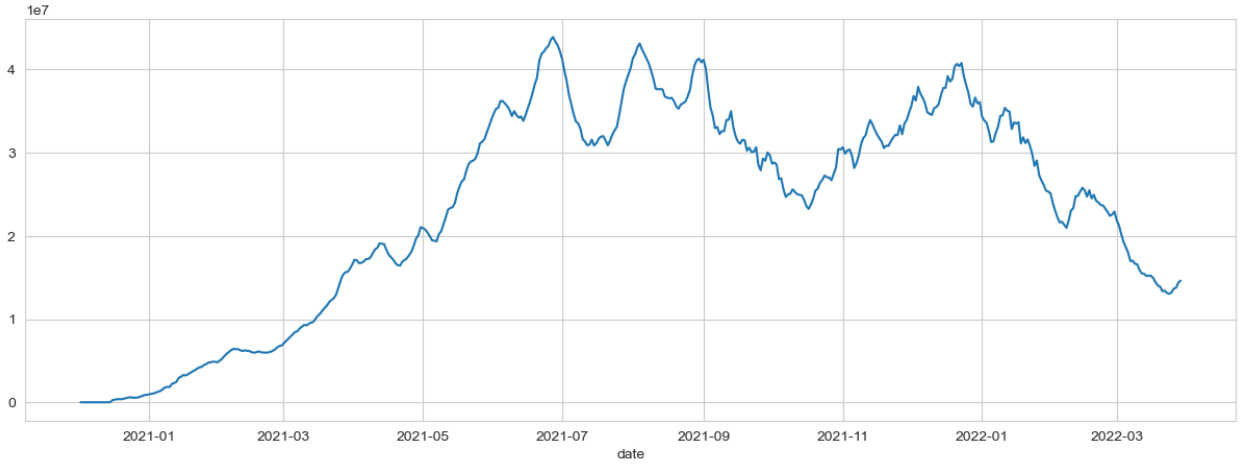
Daily vaccinations

```
x= df.groupby("date").daily_vaccinations.sum()

plt.figure(figsize= (15,5))

sns.lineplot(x.index,x.values)

plt.show()
```

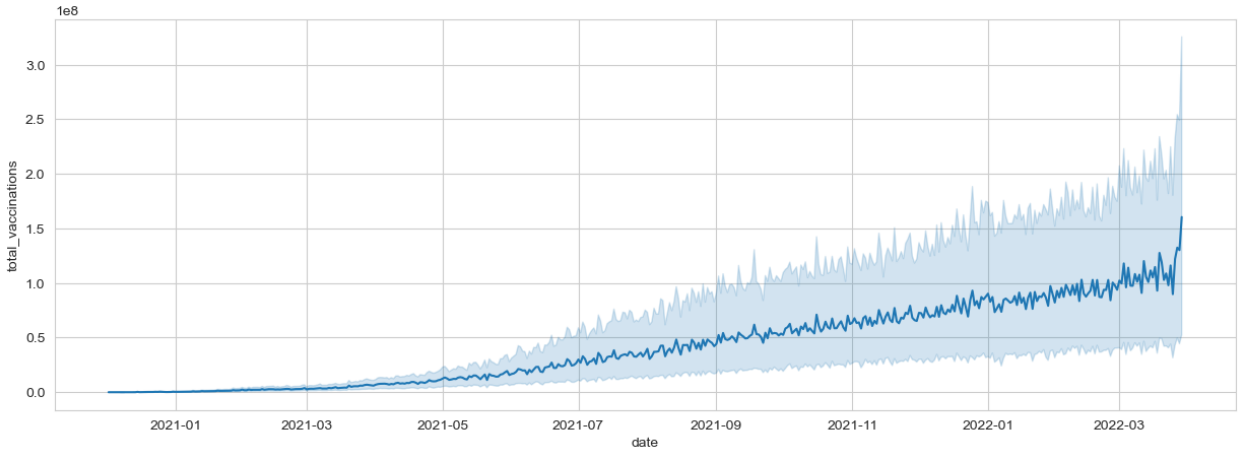


Total vaccinations

plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "total_vaccinations",data= df)

plt.show()

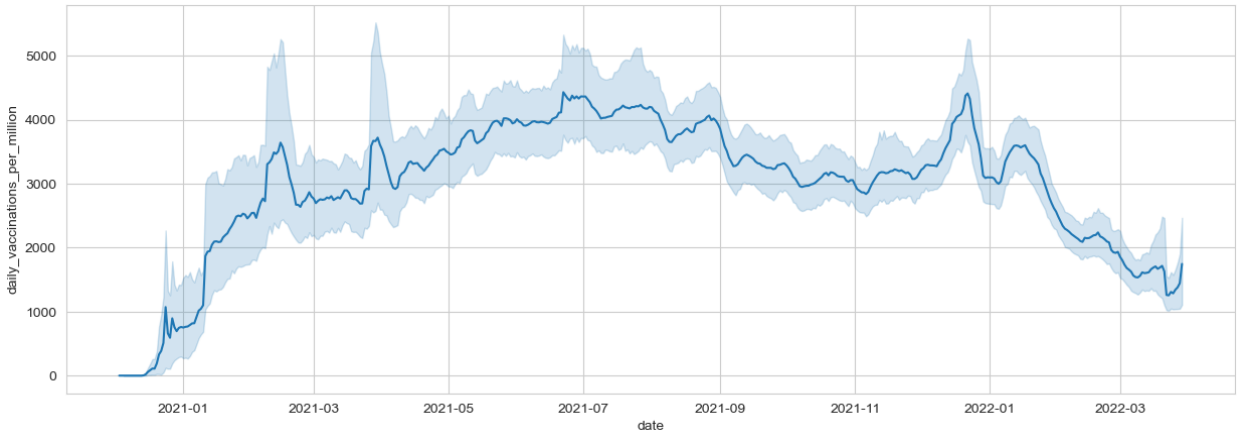


Daily vaccination per million

plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "daily_vaccinations_per_million",data= df)

plt.show()

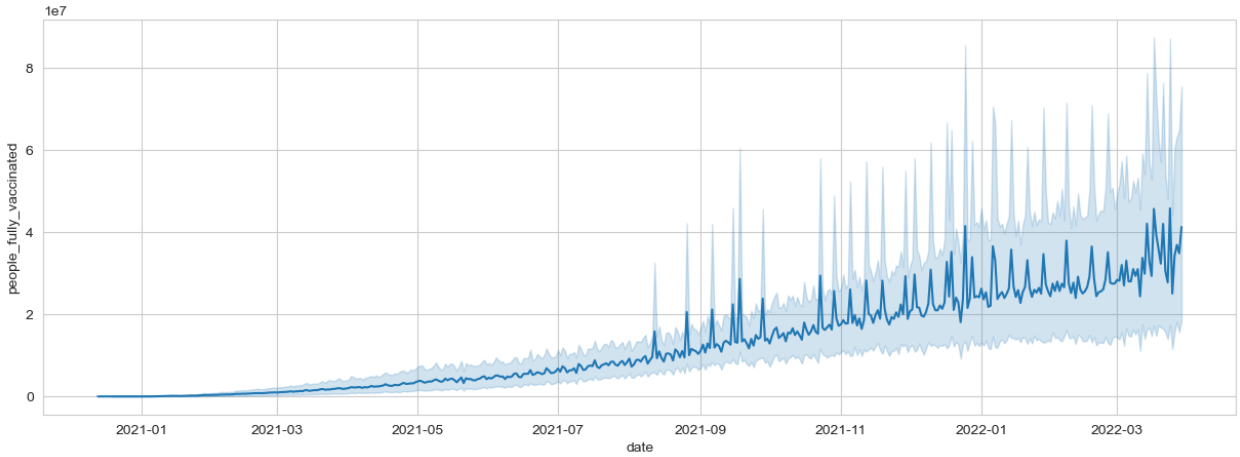


People fully vaccinated

```
plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "people_fully_vaccinated",data= df)

plt.show()
```

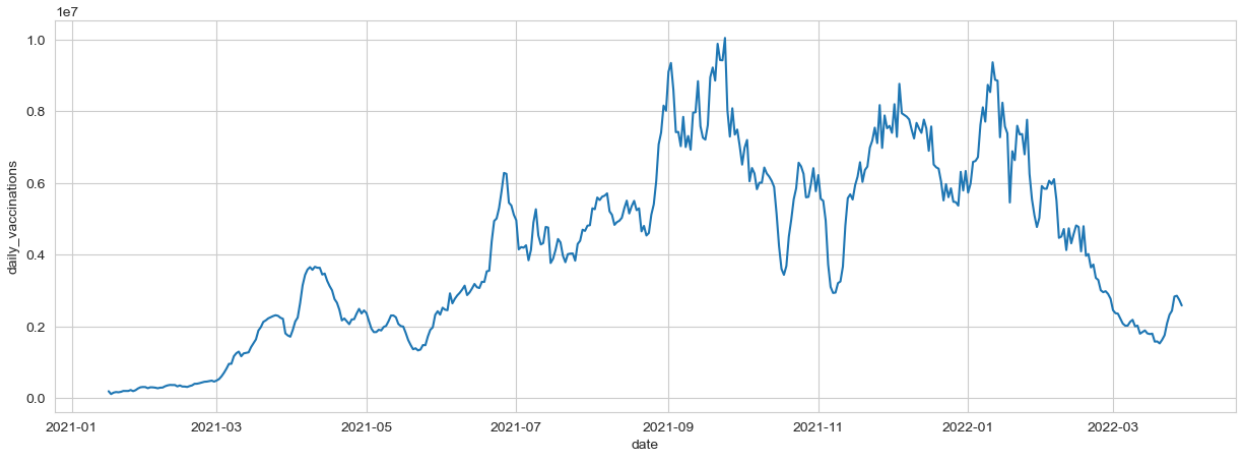


Daily vaccinations in India

```
plt.figure(figsize= (15,5))


sns.lineplot(x= "date",y= "daily_vaccinations",data= df[df.country== "India"])

plt.show()
```

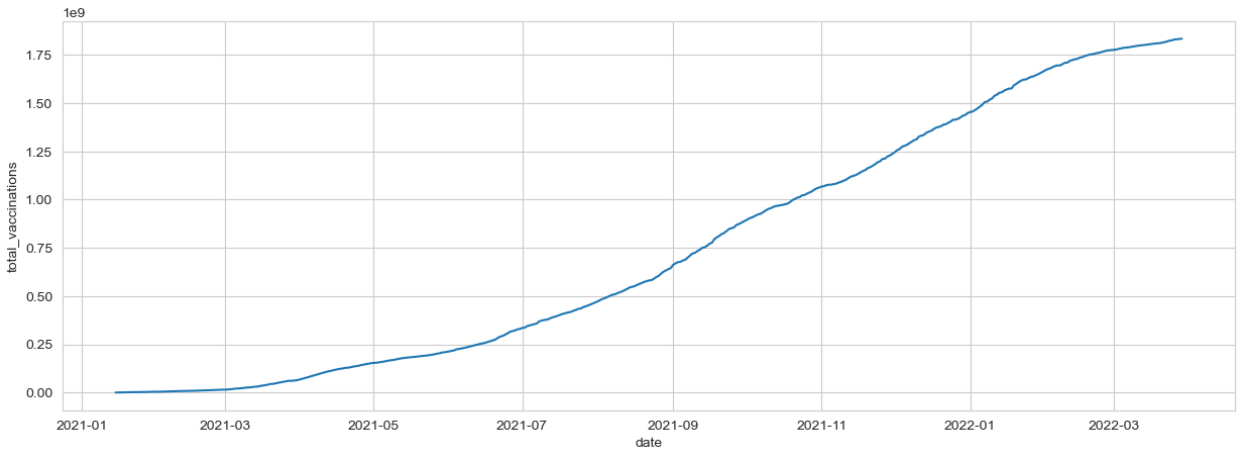


Total vaccinations in India

```
plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "total_vaccinations",data= df[df["country"]=="India"])

plt.show()
```

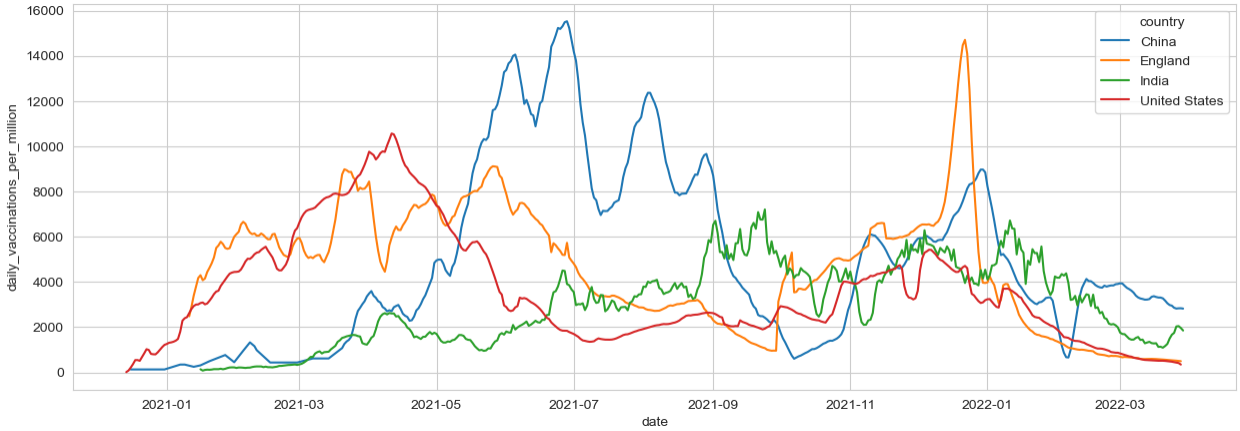


Daily vaccination per million comparison

```
plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "daily_vaccinations_per_million" ,data= x,hue= "country")

plt.show()
```


daily_vaccinations_per_million
16000
14000
12000
10000
8000
6000
4000
2000
0
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
date
country
China
England
India
United States