Project: Covid Vaccines Analysis

Empathize and Understand the Problem:

- Understanding the significance of analyzing COVID-19 vaccine data in a specific region.

- Identify the key challenges and concerns related to vaccine distribution, effectiveness, and public perception.

- Gather insights from healthcare experts, public health authorities, and individuals receiving or hesitant about vaccines.

Defining Clear Objectives:

Objective 1: Analyze historical COVID-19 vaccination data to identify vaccination trends and patterns.

Objective 2: Identify regions or vaccination centers with consistently high or low vaccination rates.

Objective 3: Develop a predictive model to estimate vaccine coverage based on demographics and vaccine type.

Ideation and Analysis Approach:

- Data Collection: Identify sources of COVID-19 vaccine data, which may include government health agencies, vaccination centers, and research institutions.

- Data Pre-processing: Clean and preprocess the data, addressing missing values, outliers, and data quality issues.

- Data Analysis: Utilize statistical analysis and visualization techniques to uncover trends and patterns in vaccination data.

- Vaccination Rate Hotspot Detection: Develop criteria or algorithms to identify areas with consistently high or low vaccination rates.

- Predictive Modeling: Select suitable machine learning algorithms to build predictive models for vaccine coverage.

- Evaluation: Define evaluation metrics to assess the performance of predictive models.

Prototype and Visualization Selection:

- Utilize data visualization libraries like Matplotlib, Seaborn, or Plotly for visualizations.

- Use line charts to illustrate vaccination trends over time.

- Heatmaps or geographical maps to pinpoint regions with varying vaccination rates.

- Scatter plots or regression plots to visualize relationships between demographics and vaccine coverage.

Build and Implement:

- Develop the full data analysis and visualization pipeline based on the refined approach.

Test and Iterate:

- Continuously test and refine the analysis and visualization based on feedback and new insights.

Deliver Insights:

- Present findings and insights in a clear and understandable manner.

- Use visualizations to communicate vaccination trends, hotspot areas, and the predictive model's performance.

- Address public concerns and contribute to informed decision-making regarding COVID-19 vaccination strategies.

This adapted approach will enable you to analyze COVID-19 vaccine data effectively and provide valuable insights for public health efforts.

INNOVATION:

Designing an innovation for COVID vaccine analysis is a multi-step process that involves careful planning, development, testing, and implementation. Here's a detailed overview of the steps to transform your design into a practical solution for COVID vaccine analysis:

1. **Clarify Objectives:** Clearly define the objectives of your innovation. What problem does it aim to solve in COVID vaccine analysis? Is it improving efficacy, safety monitoring, or distribution?
2. **Concept Development:** Begin by brainstorming and refining your design concept. Consider all aspects, including the technology, data analysis, and tools involved.
3. **Feasibility Study:** Assess the feasibility of your innovation. What resources will be required, and do they align with your available budget and time frame? Investigate any legal or regulatory requirements.
4. **Prototype Creation:** Develop a prototype of your innovation. Depending on your design, this could be software, hardware, or a combination of both. The prototype should demonstrate how your solution works.
5. **Data and Technology Integration:** Identify the data sources and technology components required for your innovation. Consider how data will be collected, stored, and analyzed.
6. **Testing and Validation:** Conduct rigorous testing to ensure your innovation works as intended. Test for accuracy, efficiency, and reliability. Validate your results against existing methods or data.
7. **Iterative Improvement:** Based on the test results, make necessary improvements to your innovation. This might involve refining algorithms, improving user interfaces, or enhancing data collection methods.

8. **Regulatory Compliance:** If your innovation is intended for use in clinical settings or for regulatory purposes, ensure it complies with relevant standards and regulations. Seek approvals or certifications if necessary.
9. **Data Security and Privacy:** Implement robust data security and privacy measures to protect sensitive information. Encryption, access controls, and anonymization of data may be necessary.
10. **Scalability Planning**: Consider how your innovation will scale as the demand for COVID vaccine analysis grows. This might involve optimizing software for large datasets or manufacturing more hardware components.
11. **User Training and Documentation:** Develop user manuals and provide training for individuals who will operate or interact with your innovation. Ensure it is user-friendly and accessible to a wide range of users.
12. **Deployment Strategy:** Plan the deployment of your innovation. Will it be a web-based platform, a mobile app, or integrated into existing systems? Develop a rollout strategy that minimizes disruptions.
13. **Monitoring and Maintenance:** Establish a system for ongoing monitoring and maintenance. Regularly update and maintain your innovation to ensure it remains effective and secure.
14. **Data Analysis and Reporting:** Develop comprehensive data analysis tools and reporting features within your innovation to help users interpret the results of COVID vaccine analysis.
15. **User Feedback and Improvement:** Continuously gather feedback from users to identify areas of improvement. Use this feedback to enhance the performance and usability of your innovation.
16. **Documentation and Knowledge Sharing:** Create detailed documentation that outlines the design, development, and implementation of your innovation. This knowledge can be crucial for future updates or troubleshooting.
17. **Communication and Public Awareness:** Promote your innovation and its benefits through appropriate channels, including scientific publications, conferences, and collaborations with relevant organizations.
18. **Sustainability and Future Developments:** Plan for the sustainability of your innovation, considering future developments and advancements in COVID vaccine analysis.

Loading and Pre-processing of data:

```
from google.colab import drive

drive.mount('/content/drive/')
```

Loading data:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn import metrics
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.tree import DecisionTreeRegressor
import xgboost as xgb
from sklearn.cluster import KMeans
from sklearn.model_selection import cross_val_score, KFold

cov19=pd.read_csv('/content/drive/MyDrive/dataset/country_vaccinations.csv')
cov19
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	

```
cov19.describe()
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinat
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313065e+05	80.188543	
std	2.246004e+08	7.078731e+07	5.713920e+07	1.212427e+06	7.682388e+05	67.913577	
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	
25%	5.264100e+05	3.494642e+05	2.439622e+05	4.668000e+03	9.000000e+02	16.050000	
50%	3.590096e+06	2.187310e+06	1.722140e+06	2.530900e+04	7.343000e+03	67.520000	
75%	1.701230e+07	9.152520e+06	7.559870e+06	1.234925e+05	4.409800e+04	132.735000	
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	

This command is used to view the brief summary of the dataset. We can see the mathematical parameters such as percentiles, standard deviation , mean, minimum and maximum values and count of each column.

cov19.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
```

Info command is used check the datatype of every column and the count of each column. The difference between the describe() and info() is that describe command will give the mathematical parameters but info command will not give the mathematical parameters such as mean and standard deviation

Data Preprocessing:

cov19.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
```

cov19_fillna = cov19

cov19_fillna

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per
0	0	1	2021 02 22	0.000000e+00	0.000000e+00	1.413830e+07	270599.578248	131306.486075	
1	0	1	2021 02 23	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	6
2	0	1	2021 02 24	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	6
3	0	1	2021 02 25	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	6
4	0	1	2021 02 26	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	6
...	...	...	...	...	...	...	...	...	
86507	222	222	2022 03 25	8.691642e+06	4.814582e+06	3.473523e+06	139213.000000	69579.000000	5
86508	222	222	2022 03	8.791728e+06	4.886242e+06	3.487962e+06	100086.000000	83429.000000	5

0s completed at 11:11 PM

cov19_fillna.fillna(cov19_fillna.mean(), inplace=True)

count the number of NaN values in each column

print(cov19_fillna.isnull().sum())

cov19_fillna

```
country                                 0
iso_code                                0
date                                    0
total_vaccinations                      0
people_vaccinated                       0
people_fully_vaccinated                 0
daily_vaccinations_raw                  0
daily_vaccinations                      0
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                             0
source_website                          0
dtype: int64
<ipython-input-9-9e428849e60a>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to Fal
  cov19_fillna.fillna(cov19_fillna.mean(), inplace=True)
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccination
0	Afghanistan	AFG	2021-02-22	0.000000e+00	0.000000e+00	1.413830e+07	270599.578248	131305.486075	
1	Afghanistan	AFG	2021-02-23	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	

✓ 0s completed at 11:11 PM

le=LabelEncoder()

cov19['country']=le.fit_transform(cov19['country'])

cov19

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_p
0	0	AFG	2021-02-22	0.000000e+00	0.000000e+00	1.413830e+07	270599.578248	131305.486075	
1	0	AFG	2021-02-23	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
2	0	AFG	2021-02-24	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
3	0	AFG	2021-02-25	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
4	0	AFG	2021-02-26	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
...	...	...	...	...	...	...	...	...	
86507	222	ZWE	2022-03-25	8.691642e+06	4.814582e+06	3.473623e+06	139213.000000	69579.000000	

✓ 0s completed at 11:11 PM

le=LabelEncoder()

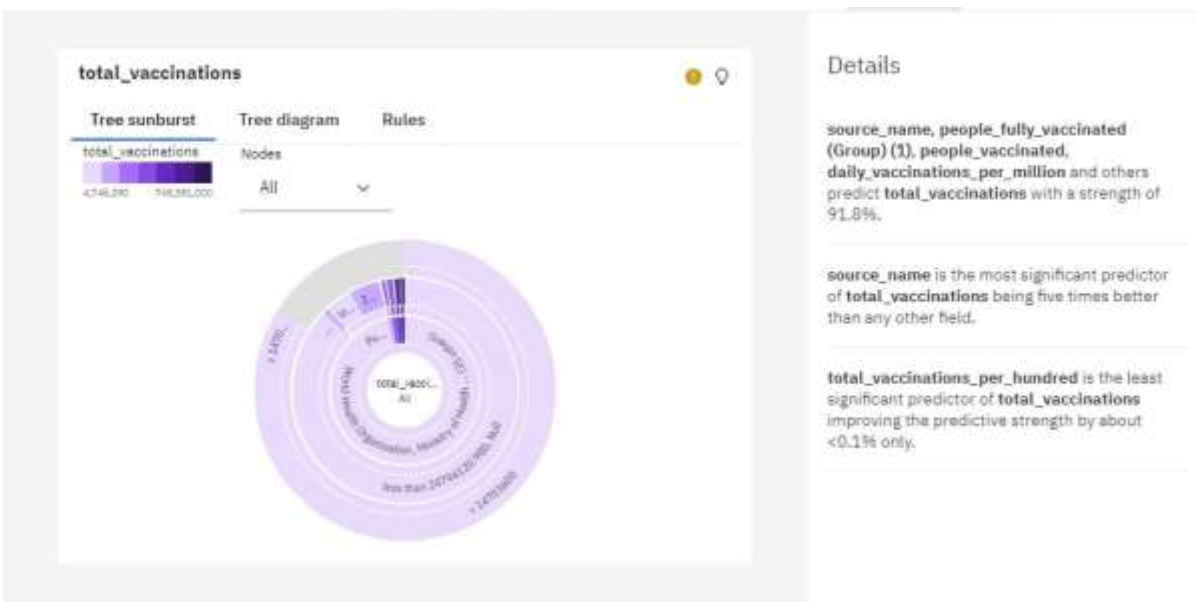
cov19['iso_code']=le.fit_transform(cov19['iso_code'])

cov19

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per
0	0	1	2021-02-22	0.000000e+00	0.000000e+00	1.413830e+07	270599.578248	131305.486075	
1	0	1	2021-02-23	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
2	0	1	2021-02-24	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
3	0	1	2021-02-25	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
4	0	1	2021-02-26	4.592964e+07	1.770508e+07	1.413830e+07	270599.578248	1367.000000	
...	...	...	...	...	...	...	...	...	
86507	222	222	2022-03-25	8.691642e+06	4.814582e+06	3.473523e+06	139213.000000	69579.000000	

cov19.columns

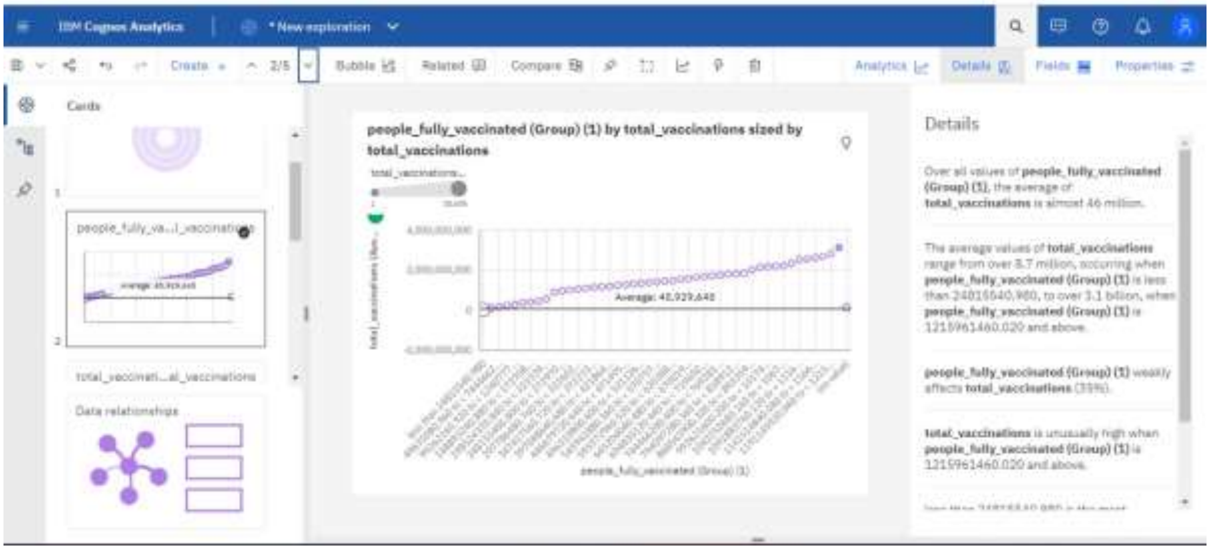
Index(['country', 'iso_code', 'date', 'total_vaccinations', 'people_vaccinated', 'people_fully_vaccinated', 'daily_vaccinations_raw', 'daily_vaccinations', 'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred', 'daily_vaccinations_per_million', 'vaccines', 'source_name', 'source_website'], dtype='object')



source_name, **people_fully_vaccinated (Group) (1)**, **people_vaccinated**, **daily_vaccinations_per_million** and others predict **total_vaccinations** with a strength of 91.8%.

source_name is the most significant predictor of **total_vaccinations** being five times better than any other field.

total_vaccinations_per_hundred is the least significant predictor of **total_vaccinations** improving the predictive strength by about <0.1% only.



total_vaccinations is unusually high when **total_vaccinations_per_hundred (Group) (1)** is 221.0368 to < 227.9442.

source_name strongly affects **total_vaccinations** (71%).

total_vaccinations is unusually high when **source_name** is National Health Commission.

Over all values of **total_vaccinations_per_hundred (Group) (1)** and **source_name**, the average of **total_vaccinations** is almost 46 million.

The average values of **total_vaccinations** range from 0 to over 3.2 billion.

total_vaccinations_per_hundred (Group) (1) and **source_name** strongly affect **total_vaccinations** (100%).

total_vaccinations is unusually high when the combinations of **total_vaccinations_per_hundred (Group) (1)** and **source_name** are 221.0368 to < 227.9442 and National Health Commission and 214.1294 to < 221.0368 and National Health Commission.

less than 6.9074 is the most frequently occurring category of **total_vaccinations_per_hundred (Group) (1)** with a count of 7505 items with **total_vaccinations** values (17.2 % of the total).

Ministry of Health is the most frequently occurring category of **source_name** with a count of 9981 items with **total_vaccinations** values (22.9 % of the total).

Chart A

date - Top 10 by daily_vaccinations_per_million

date, total_vaccinations and daily_vaccinations_per_million

5

date

total_vaccinations

daily_vaccinations_per_million

6/22/2021

2,699,790,526

965,713

6/23/2021

2,788,620,339

954,815

6/26/2021

2,877,147,766

954,034

6/28/2021

2,996,944,602

951,522

Chart B

daily_vaccinations and total_vaccinations by country colored by country

10,562,357

2 of 200 items

Summary	Select Chart A : total_vaccinationsChart A : daily_vaccinations_per_million	Select Chart B : daily_vaccinationsChart B : total_vaccinations	Combined
Chart percent of data set	1.72%	100%	-
Average	3,434,983,805.7	50,763,407.49	-
Chart total	34,349,838,057	11,320,239,871	-

people_fully_vaccinated (Group) (1) by total_vaccinations sized by total_vaccinations

less than 24815540.98049631080.960 to < 7444662...99262160.920 to < 1240777...148893240.880 to < 173708...198524320.840 to < 223339...248155400.800 to < 272970...297786480.760 to < 322602...347417560.720 to < 372233...397048640.680 to < 421864...446679720.640 to < 471495...496310800.600 to < 521126...545941880.560 to < 570757...595572960.520 to < 620388...645204040.480 to < 670019...694835120.440 to < 719650...744466200.400 to < 769281...794097280.360 to < 818912...868543900.300 to < 893359...992621600.200 to < 10174...1042252680.160 to < 1067...1091883760.120 to < 1116...1141514840.080 to < 1166...1191145920.040 to < 1215...(no value)people_fully_vaccinated (Group) (1)-2,000,000,00002,000,000,0004,000,000,000total_vaccinations (Ave…5Average: 45,929,645

total_vaccinations (Count)

133,635

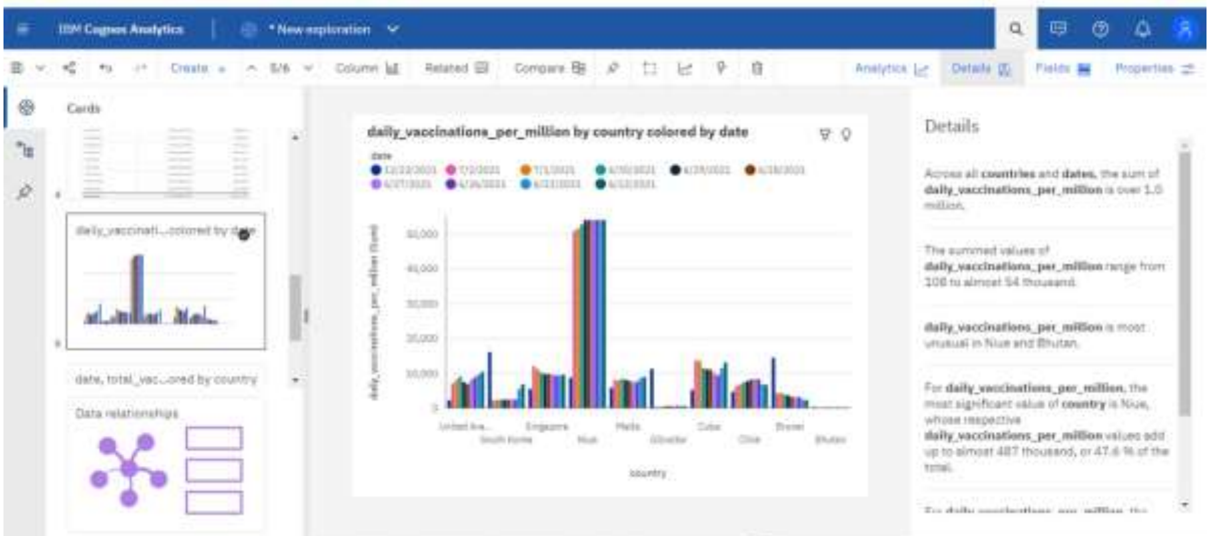
daily_vaccinations_per_million by country colored by date

United Arab...South KoreaSingaporeNiueMaltaGibraltarCubaChileBruneiBhutancountry010,00020,00030,00040,00050,000daily_vaccinations_per_million (Sum)

date

- 12/22/2021

- 7/2/2021
- 7/1/2021
- 6/30/2021
- 6/29/2021
- 6/28/2021
- 6/27/2021
- 6/26/2021
- 6/23/2021
- 6/22/2021



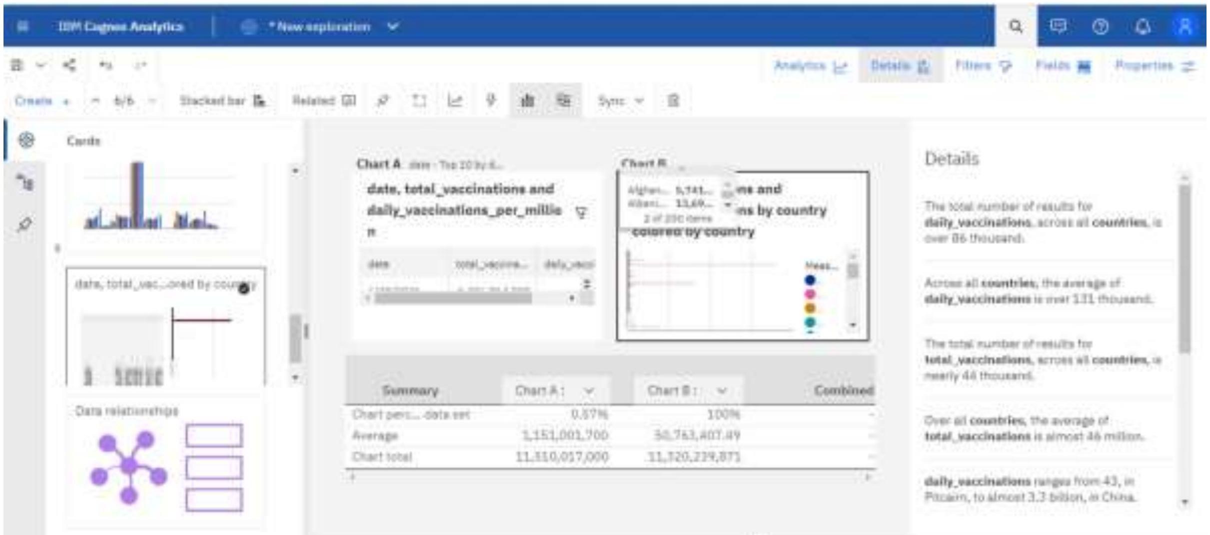
Across all **countries** and **dates**, the sum of **daily_vaccinations_per_million** is over 1.0 million.

The summed values of **daily_vaccinations_per_million** range from 108 to almost 54 thousand.

daily_vaccinations_per_million is most unusual in Niue and Bhutan.

For **daily_vaccinations_per_million**, the most significant value of **country** is Niue, whose respective **daily_vaccinations_per_million** values add up to almost 487 thousand, or 47.6 % of the total.

For **daily_vaccinations_per_million**, the most significant values of **date** are 2021-06-22, 2021-06-23, 2021-07-01, 2021-06-30, and 2021-07-02, whose respective **daily_vaccinations_per_million** values add up to over 535 thousand, or 52.3 % of the total.



The total number of results for **daily_vaccinations**, across all **countries**, is over 86 thousand.

Across all **countries**, the average of **daily_vaccinations** is over 131 thousand.

The total number of results for **total_vaccinations**, across all **countries**, is nearly 44 thousand.

Over all **countries**, the average of **total_vaccinations** is almost 46 million.

daily_vaccinations ranges from 43, in Pitcairn, to almost 3.3 billion, in China.

total_vaccinations ranges from 348, in Pitcairn, to approximately 709 billion, in China.

Norway (0.6 %), Latvia (0.6 %), and Denmark (0.6 %) are the most frequently occurring categories of **country** with a combined count of 1435 items with **daily_vaccinations** values (1.7 % of the total).

Norway is the most frequently occurring category of **country** with a count of 482 items with **total_vaccinations** values (1.1 % of the total).

date, total_vaccinations and daily_vaccinations_per_million 5

date	total_vaccinations	daily_vaccinations_per_million
6/22/2021	2,699,790,526	965,713
6/23/2021	2,788,620,339	954,815
6/26/2021	2,877,147,766	954,034
6/28/2021	2,996,944,602	951,522
6/30/2021	3,062,159,402	951,412
7/2/2021	3,072,014,637	951,132
7/1/2021	3,085,188,933	950,829
6/27/2021	2,942,024,392	944,228
12/22/2021	7,810,948,031	943,909
6/29/2021	3,014,999,429	943,898
Summary	34,349,838,057	9,511,492

VISUALIZATION:

```
from google.colab import drive
drive.mount('/content/drive/')
```

Mounted at /content/drive/

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df= pd.read_csv("/content/drive/MyDrive/country_vaccinations.csv")
```

```
df.head()
```

```
       country iso_code        date  total_vaccinations  people_vaccinated  \
0  Afghanistan      AFG  2021-02-22                 0.0                0.0
1  Afghanistan      AFG  2021-02-23                 NaN                NaN
2  Afghanistan      AFG  2021-02-24                 NaN                NaN
3  Afghanistan      AFG  2021-02-25                 NaN                NaN
4  Afghanistan      AFG  2021-02-26                 NaN                NaN
```

```
  people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
0                     NaN                     NaN                 NaN
1                     NaN                     NaN              1367.0
2                     NaN                     NaN              1367.0
3                     NaN                     NaN              1367.0
4                     NaN                     NaN              1367.0

  total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
0                            0.0                            0.0
1                            NaN                            NaN
2                            NaN                            NaN
3                            NaN                            NaN
4                            NaN                            NaN

  people_fully_vaccinated_per_hundred  daily_vaccinations_per_million  \
0                                 NaN                             NaN
1                                 NaN                            34.0
2                                 NaN                            34.0
3                                 NaN                            34.0
4                                 NaN                            34.0

                                            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...

                source_name           source_website
0  World Health Organization  https://covid19.who.int/
1  World Health Organization  https://covid19.who.int/
2  World Health Organization  https://covid19.who.int/
3  World Health Organization  https://covid19.who.int/
4  World Health Organization  https://covid19.who.int/
```

```
df.describe()
```

```
       total_vaccinations  people_vaccinated  people_fully_vaccinated  \
count        4.360700e+04       4.129400e+04             3.880200e+04
mean         4.592964e+07       1.770508e+07             1.413830e+07
std          2.246004e+08       7.078731e+07             5.713920e+07
min          0.000000e+00       0.000000e+00             1.000000e+00
25%          5.264100e+05       3.494642e+05             2.439622e+05
50%          3.590096e+06       2.187310e+06             1.722140e+06
75%          1.701230e+07       9.152520e+06             7.559870e+06
max          3.263129e+09       1.275541e+09             1.240777e+09

       daily_vaccinations_raw  daily_vaccinations  \
count            3.536200e+04        8.621300e+04
mean             2.705996e+05        1.313055e+05
```

```
std        1.212427e+06     7.682388e+05
min        0.000000e+00     0.000000e+00
25%        4.668000e+03     9.000000e+02
50%        2.530900e+04     7.343000e+03
75%        1.234925e+05     4.409800e+04
max        2.474100e+07     2.242429e+07

       total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
count                    43607.000000                   41294.000000
mean                        80.188543                      40.927317
std                         67.913577                      29.290759
min                          0.000000                       0.000000
25%                         16.050000                      11.370000
50%                         67.520000                      41.435000
75%                        132.735000                      67.910000
max                        345.370000                     124.760000

       people_fully_vaccinated_per_hundred  daily_vaccinations_per_million
count                         38802.000000                    86213.000000
mean                             35.523243                     3257.049157
std                              28.376252                     3934.312440
min                               0.000000                        0.000000
25%                               7.020000                      636.000000
50%                              31.750000                     2050.000000
75%                              62.080000                     4682.000000
max                             122.370000                   117497.000000
```

```
df.dtypes
```

```
country                                 object
iso_code                                object
date                                    object
total_vaccinations                     float64
people_vaccinated                      float64
people_fully_vaccinated                float64
daily_vaccinations_raw                 float64
daily_vaccinations                     float64
total_vaccinations_per_hundred         float64
people_vaccinated_per_hundred          float64
people_fully_vaccinated_per_hundred    float64
daily_vaccinations_per_million         float64
vaccines                                object
source_name                             object
source_website                          object
dtype: object
```

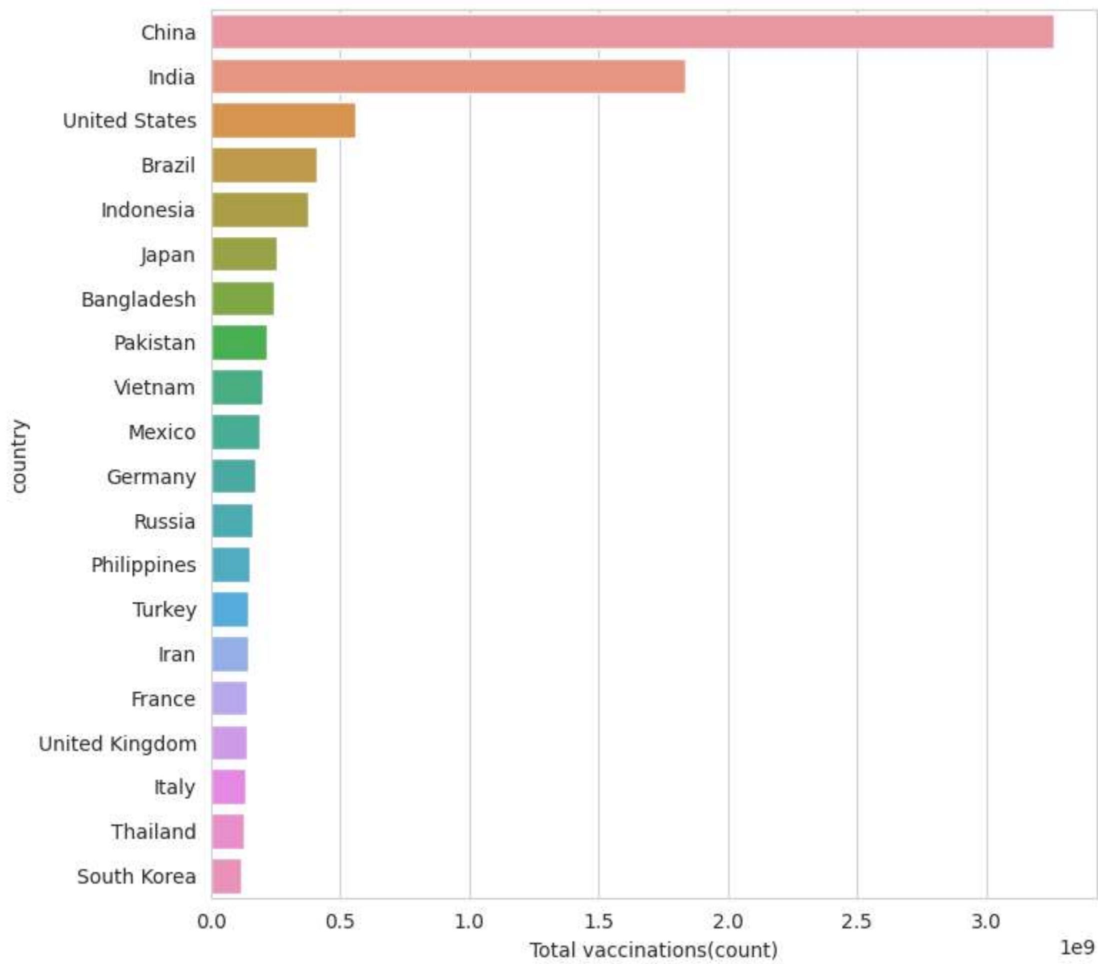
```
df["date"]= pd.to_datetime(df.date)
```

```
df["Total_vaccinations(count)"]= df.groupby("country").total_vaccinations.tail(1)
```

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

```
country
China             3.263129e+09
India             1.834501e+09
United States     5.601818e+08
Brazil            4.135596e+08
Indonesia         3.771089e+08
Japan             2.543456e+08
Bangladesh        2.436427e+08
Pakistan          2.193686e+08
Vietnam           2.031444e+08
Mexico            1.919079e+08
Germany           1.719400e+08
Russia            1.636012e+08
Philippines       1.487991e+08
Turkey            1.468819e+08
Iran              1.467926e+08
France            1.416662e+08
United Kingdom    1.409683e+08
Italy             1.358709e+08
Thailand          1.288824e+08
South Korea       1.206045e+08
Name: Total_vaccinations(count), dtype: float64
```

```
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=x.values, y=x.index)

ax.set_xlabel("Total vaccinations(count)")
plt.show()
```

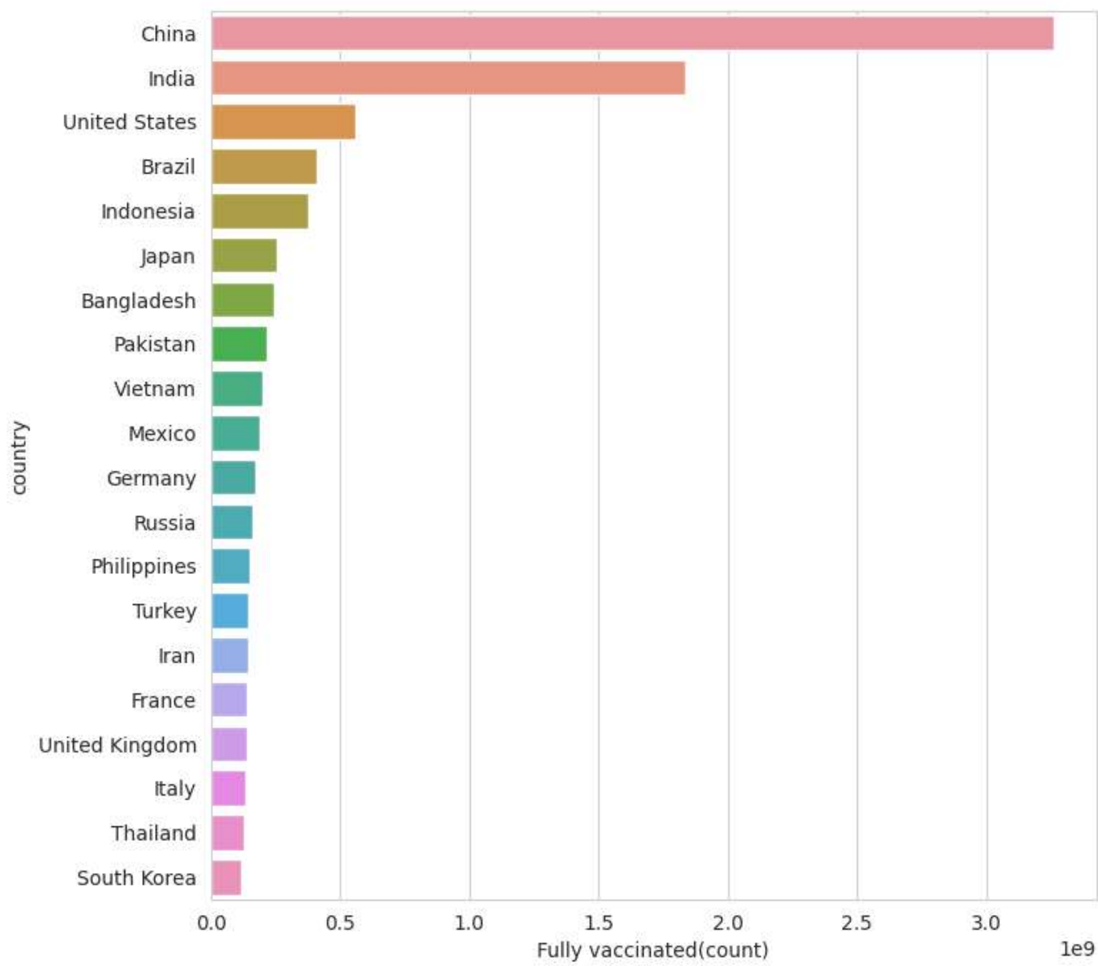



```
df["Full_vaccinations(count)"]= df.groupby("country").people_fully_vaccinated.tail(1)

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

```
country
India            828229455.0
United States    217498967.0
Brazil           160272858.0
Indonesia        158830466.0
Bangladesh       107712737.0
Pakistan         101881176.0
Japan            100633737.0
Mexico            79711762.0
Vietnam           77754108.0
Russia            72841232.0
Philippines       65804988.0
Germany           63142649.0
Iran              56810058.0
Turkey            52968985.0
France            52438706.0
Thailand          50159803.0
```

United Kingdom   49404026.0
Italy   47817555.0
South Korea   44482876.0
England   41501690.0
Name: Full_vaccinations(count), dtype: float64

```
#barplot visualization of top countries with most full vaccinations

sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax = sns.barplot(x=x.values, y=x.index)
ax.set_xlabel("Fully vaccinated(count)")
plt.show()
```




```
df["Full_vaccinations(count)"]= df.groupby("country").people_fully_vaccinated.tail(1)

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

country
India   828229455.0
United States   217498967.0

```
Brazil            160272858.0
Indonesia         158830466.0
Bangladesh        107712737.0
Pakistan          101881176.0
Japan             100633737.0
Mexico             79711762.0
Vietnam            77754108.0
Russia             72841232.0
Philippines        65804988.0
Germany            63142649.0
Iran               56810058.0
Turkey             52968985.0
France             52438706.0
Thailand           50159803.0
United Kingdom     49404026.0
Italy              47817555.0
South Korea        44482876.0
England            41501690.0
Name: Full_vaccinations(count), dtype: float64
```

```
#Vaccine types
x=df.vaccines.unique()
y= list(x)
for i in y: print(i)
```

```
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
Oxford/AstraZeneca
Oxford/AstraZeneca, Pfizer/BioNTech
Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
Pfizer/BioNTech
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light,
Sputnik V
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Moderna, Pfizer/BioNTech
Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
Johnson&Johnson, Oxford/AstraZeneca
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing
```

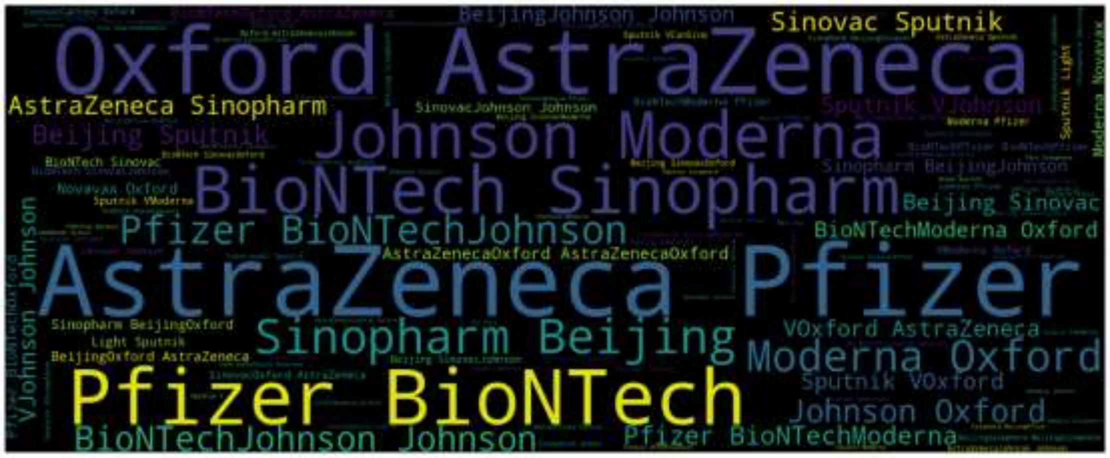
Sinopharm/Beijing
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac
Covaxin, Oxford/AstraZeneca
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V
Abdala, Soberana Plus, Soberana02
Johnson&Johnson, Moderna, Pfizer/BioNTech
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
Covaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac
Johnson&Johnson, Pfizer/BioNTech
Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Oxford/AstraZeneca, Sputnik V
Moderna
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
Oxford/AstraZeneca, Sinopharm/Beijing
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Moderna
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
Pfizer/BioNTech, Sinovac
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Covaxin, Oxford/AstraZeneca, Sputnik V
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik V
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
QazVac, Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V
Johnson&Johnson, Moderna, Novavax, Pfizer/BioNTech
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
Pfizer/BioNTech, Sinopharm/Beijing
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
CanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
Abdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik V
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
CanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V
Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
EpiVacCorona, Sputnik V
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V
Pfizer/BioNTech, Sputnik V
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V

```
Moderna, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V
Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
Pfizer/BioNTech, Sinovac, Turkovac
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001
Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik V
Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
Johnson&Johnson, Oxford/AstraZeneca, Sinovac
```

```
df.vaccines.value_counts()
```

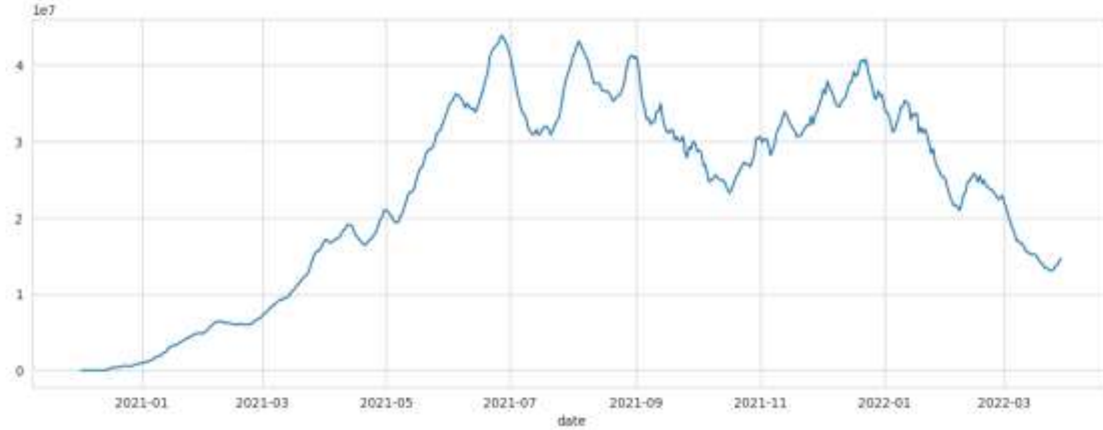
```
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech                   7608
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech                                    6263
Oxford/AstraZeneca                                                              6022
Oxford/AstraZeneca, Pfizer/BioNTech                                             4629
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech          3564
                                                                                ...
Johnson&Johnson, Oxford/AstraZeneca, Sinovac                                     312
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V                 311
Johnson&Johnson, Moderna                                                         251
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing                              228
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001   190
Name: vaccines, Length: 84, dtype: int64
```

```
from wordcloud import WordCloud, STOPWORDS
plt.figure(figsize= (20,20))
words= "".join(df["vaccines"])
final = WordCloud(width = 2000, height = 800, background_color ="black",min_font_size =
10).generate(words)
plt.imshow(final)
plt.axis("off")
plt.show()
```

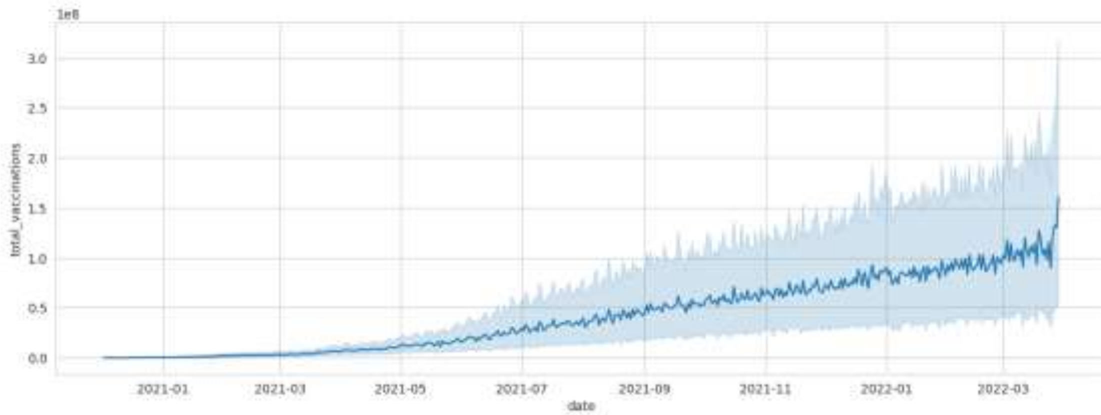



```
plt.figure(figsize=(15,15))
sns.countplot(y= "vaccines",data= df)
plt.show()
```

```
#daily vaccinations
x= df.groupby("date").daily_vaccinations.sum()
plt.figure(figsize= (15,5))
sns.lineplot(x=x.index, y=x.values)
plt.show()
```




```
#total vaccinations
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "total_vaccinations",data= df)
plt.show()
```

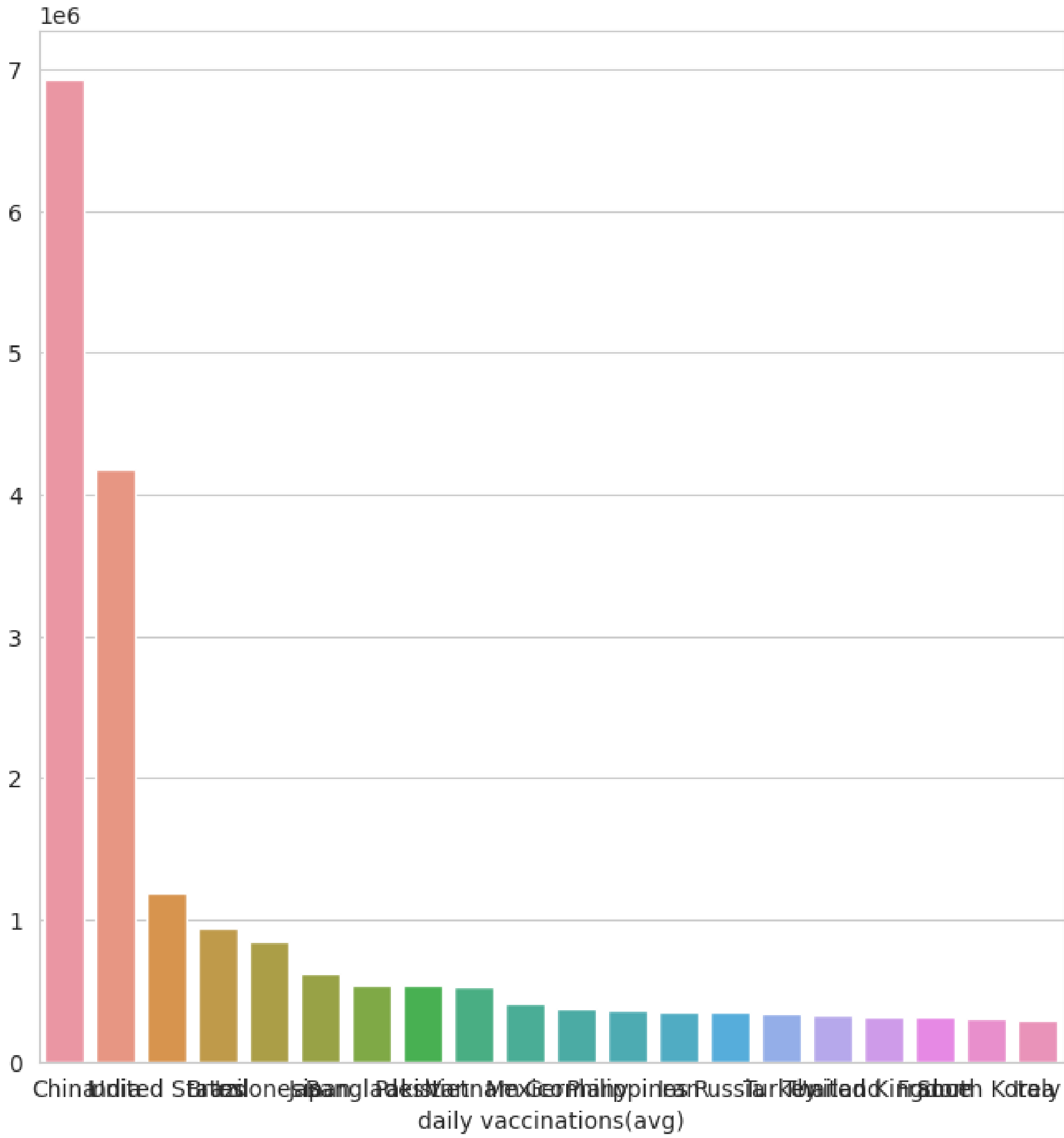



```
x= df.groupby("country").daily_vaccinations.mean().sort_values(ascending= False).head(20)
x
```

```
country
China            6.930368e+06
India            4.175994e+06
United States    1.191727e+06
Brazil           9.435287e+05
Indonesia        8.462893e+05
Japan            6.215795e+05
Bangladesh       5.453055e+05
Pakistan         5.430051e+05
Vietnam          5.310949e+05
Mexico           4.134253e+05
Germany          3.761575e+05
Philippines      3.665658e+05
Iran             3.535194e+05
Russia           3.480843e+05
Turkey           3.351917e+05
```

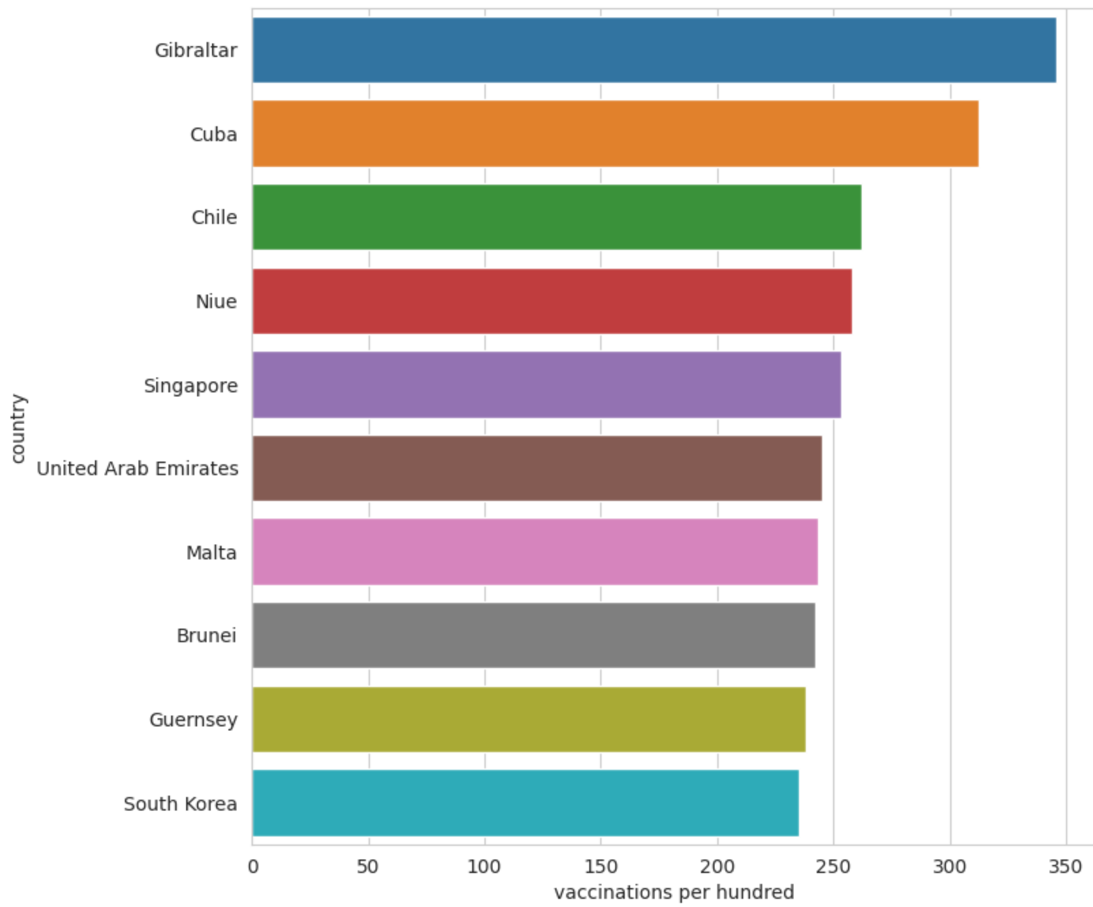
```
Thailand          3.251471e+05
United Kingdom    3.140841e+05
France            3.104963e+05
South Korea       3.042512e+05
Italy             2.970580e+05
Name: daily_vaccinations, dtype: float64
```

```
#daily vaccinations barplot
plt.figure(figsize= (8,8))
ax = sns.barplot(x=x.index, y=x.values)
ax.set_xlabel("daily vaccinations(avg)")
plt.show()
```




```
df["Total_vaccinations_per_hundred"]= df.groupby("country").total_vaccinations_per_hundred.tail(1)
```

```
x= df.groupby("country")["Total_vaccinations_per_hundred"].mean().sort_values(ascending=
False).head(10)
plt.figure(figsize= (8,8))
ax= sns.barplot(x=x.values,y=x.index)
ax.set_xlabel("vaccinations per hundred")
plt.show()
```



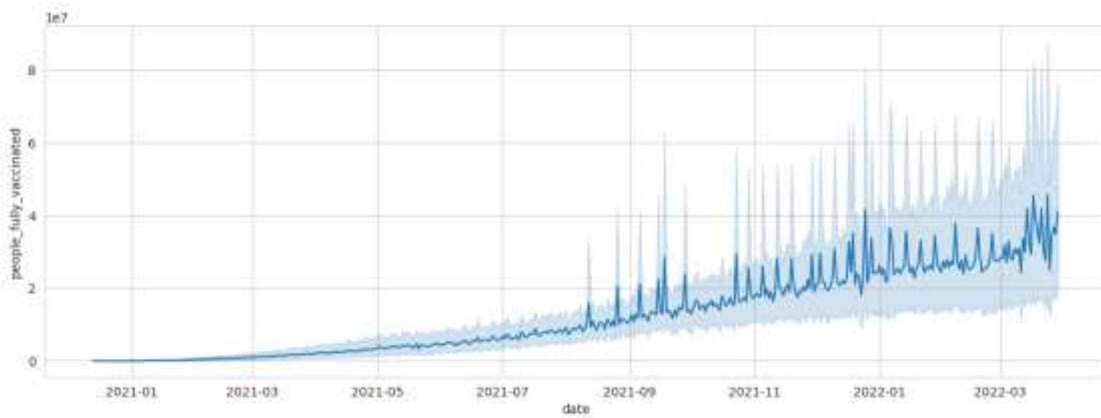

```
df.groupby("country")["daily_vaccinations_per_million"].mean().sort_values(ascending= False).head(20)
```

```
country
Falkland Islands                21185.393939
Saint Helena                    13915.164835
Tokelau                         12718.106195
Pitcairn                        10891.797619
Niue                            10109.509434
Cuba                             9955.943333
Gibraltar                        8000.463470
Bonaire Sint Eustatius and Saba  7412.000000
Bhutan                           7241.676880
Brunei                           6906.782857
Turkmenistan                     6618.888889
South Korea                      5930.227273
Uruguay                          5829.491139
```

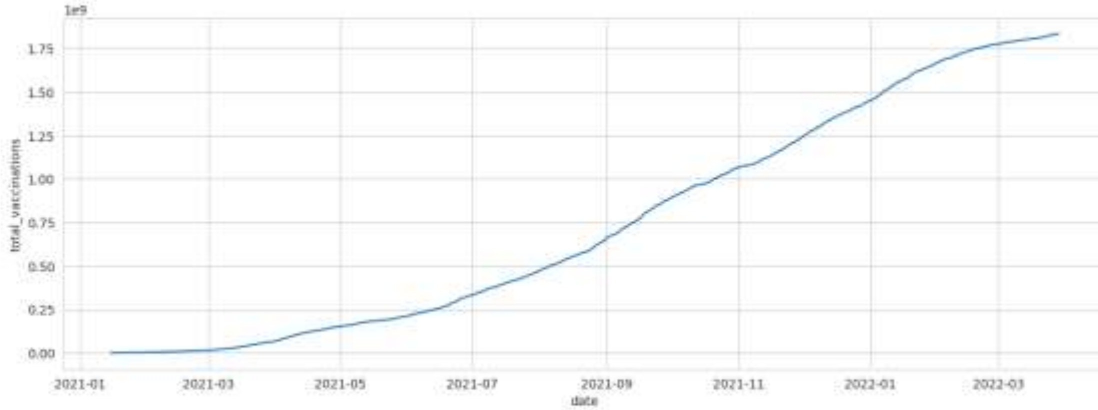
```
Chile                        5764.154525
Singapore                    5585.536424
Malta                        5553.986207
Taiwan                       5545.517426
Guernsey                     5437.624113
Australia                    5422.241895
Vietnam                      5410.000000
Name: daily_vaccinations_per_million, dtype: float64
```

```
#daily vaccination per million
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "daily_vaccinations_per_million",data= df)
plt.show()
```

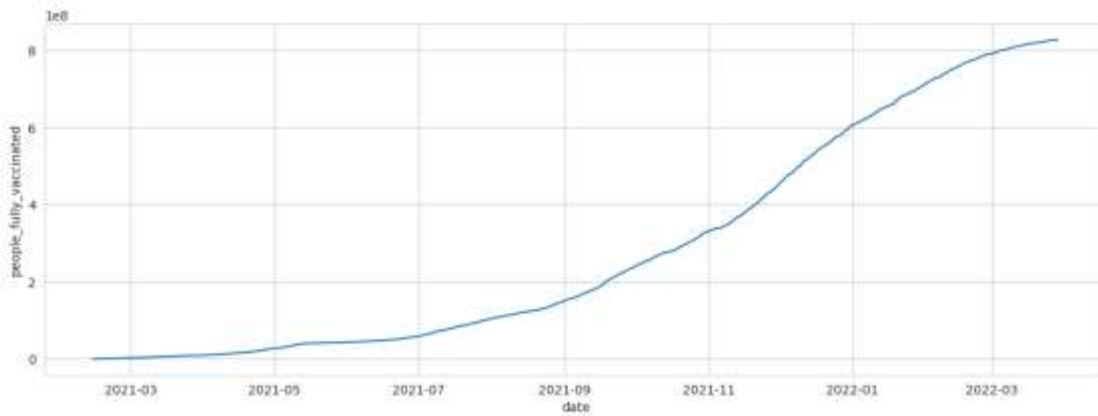

```
#people fully vaccinated
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "people_fully_vaccinated",data= df)
plt.show()
```



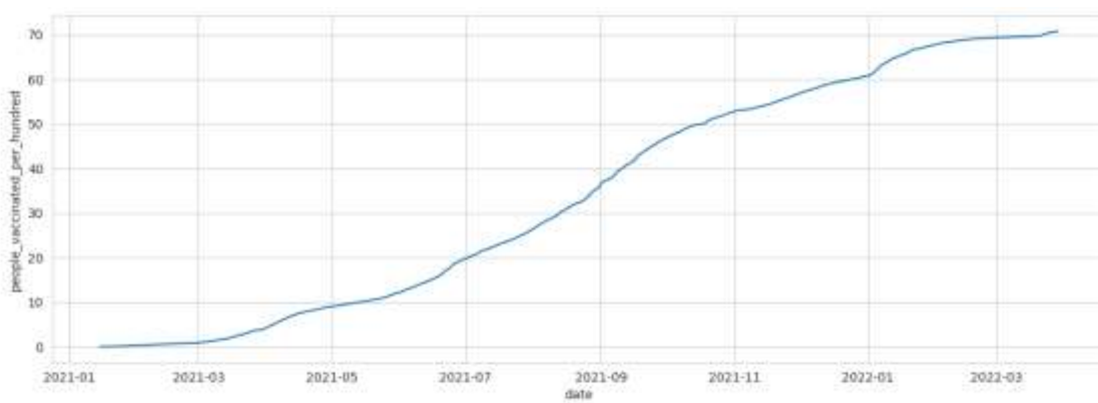

```
#Total vaccinations in India
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "total_vaccinations",data= df[df["country"]=="India"])
plt.show()
```




```
#full vaccinations in India
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "people_fully_vaccinated",data= df[df["country"]=="India"])
plt.show()
```




```
#people_vaccinated per hundred in India
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "people_vaccinated_per_hundred",data= df[df["country"]=="India"])
plt.show()
```




```
x= df[df["country"]=="India"]
z= x.vaccines.value_counts()
```

```
c= list(z.index)
c
```

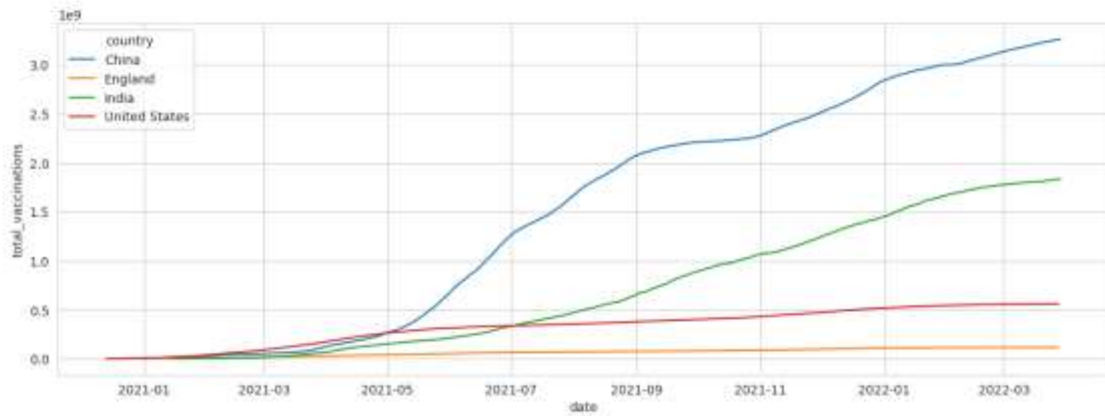
['Covaxin, Oxford/AstraZeneca, Sputnik V']

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

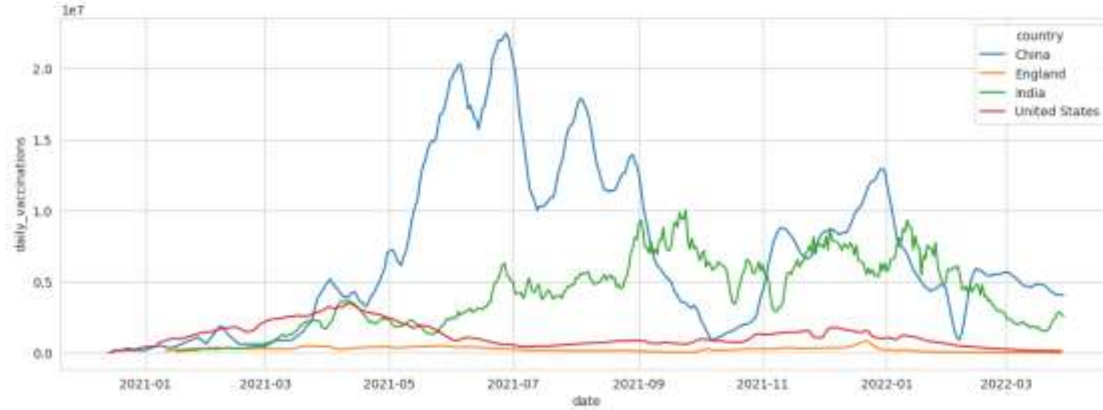
```
country
China           3.263129e+09
India           1.834501e+09
United States    5.601818e+08
Brazil          4.135596e+08
Indonesia       3.771089e+08
Name: Total_vaccinations(count), dtype: float64
```

```
#creating dataframe for top 5 vaccinated countries
x= df.loc[(df.country== "United States") | (df.country== "China")| (df.country== "India")| (df.country==
"Unted Kingdom")|(df.country== "England")]
```

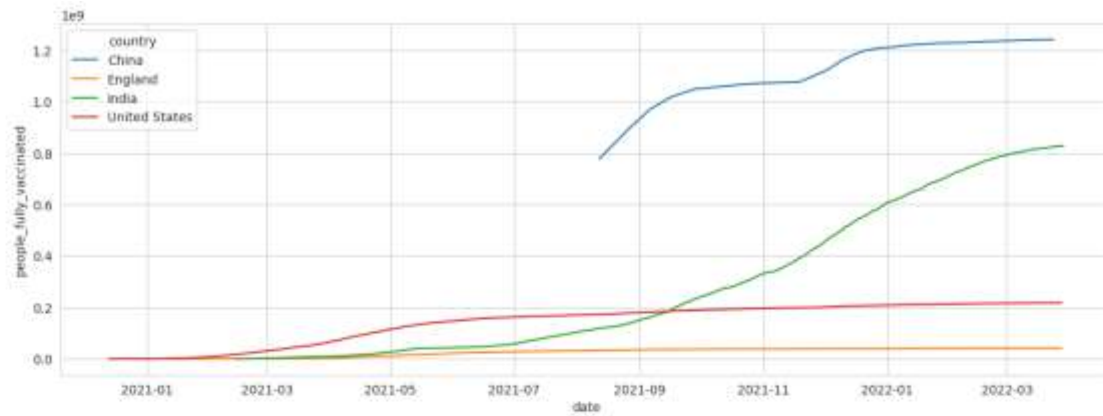
```
#total vaccination comparison
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "total_vaccinations" ,data= x,hue= "country")
plt.show()
```



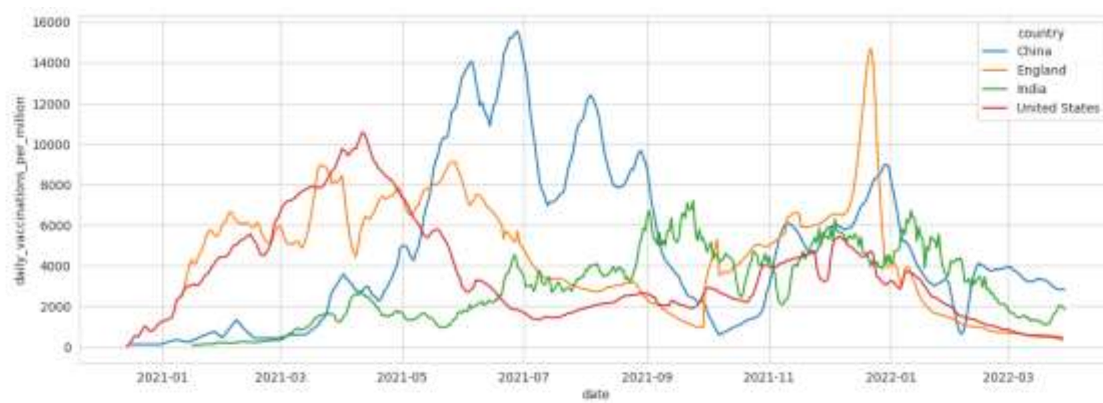

```
#daily vaccination comparison
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "daily_vaccinations" ,data= x,hue= "country")
plt.show()
```




```
#full vaccinations comparison
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "people_fully_vaccinated" ,data= x,hue= "country")
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()
```



```python
import pandas as pd

# Assuming your DataFrame is named df
summary_stats = df['daily_vaccinations'].describe()

print(summary_stats)
```

```
count    8.621300e+04
mean     1.313055e+05
std      7.682388e+05
min      0.000000e+00
25%      9.000000e+02
50%      7.343000e+03
75%      4.409800e+04
max      2.242429e+07
Name: daily_vaccinations, dtype: float64
```