Exploratory Data Analysis of

COVID-19 World Vaccination Progress

Welcome!

Introduction to the dataset:

This dataset contains the information on COVID-19 vaccination details that were collected across different countries around the world. This dataset is an open source data and is taken from the most popular website, kaggle. Since it is an open source data, anyone like me can access it for analysis purposes. Some of the information that can be seen in the dataset are:

1. **Country- this is the country for which the vaccination information is provided;**
2. **Country ISO Code - ISO code for the country;**
3. **Date - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;**
4. **Total number of vaccinations - this is the absolute number of total immunizations in the country;**
5. **Total number of people vaccinated - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;**
6. **Total number of people fully vaccinated - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;**
7. **Daily vaccinations (raw) - for a certain data entry, the number of vaccination for that date/country;**

8. **Daily vaccinations - for a certain data entry, the number of vaccination for that date/country;**
9. **Total vaccinations per hundred - ratio (in percent) between vaccination number and total population up to the date in the country;**
10. **Total number of people vaccinated per hundred - ratio (in percent) between population immunized and total population up to the date in the country;**
11. **Total number of people fully vaccinated per hundred - ratio (in percent) between population fully immunized and total population up to the date in the country;**
12. **Number of vaccinations per day - number of daily vaccination for that day and country;**
13. **Daily vaccinations per million - ratio (in ppm) between vaccination number and total population for the current date in the country;**
14. **Vaccines used in the country - total number of vaccines used in the country (up to date);**
    The Source of Data: The data was collected from `Kaggle` website. Link: https://www.kaggle.com/gpreda/covid-world-vaccination-progress/

How to run the code

This is an executable *Jupyter notebook* hosted on Jovian.ml, a platform for sharing data science projects. You can run and experiment with the code in a couple of ways: *using free online resources* (recommended) or *on your own computer*.

Option 1: Running using free online resources (1-click, recommended)

The easiest way to start executing this notebook is to click the "Run" button at the top of this page, and select "Run on Binder". This will run the notebook on mybinder.org, a free online service for running Jupyter notebooks. You can also select "Run on Colab" or "Run on Kaggle".

Option 2: Running on your computer locally

1. **Install Conda by following these instructions. Add Conda binaries to your system `PATH`, so you can use the `conda` command on your terminal.**

2. **Create a Conda environment and install the required libraries by running these commands on the terminal:**

```
conda create -n zerotopandas -y python=3.8
conda activate zerotopandas
pip install jovian jupyter numpy pandas matplotlib seaborn opendatasets --upgrade
```

1. Press the "Clone" button above to copy the command for downloading the notebook, and run it on the terminal. This will create a new directory and download the notebook. The command will look something like this:

   ```
   jovian clone notebook-owner/notebook-id
   ```

1. Enter the newly created directory using `cd directory-name` and start the Jupyter notebook.

   ```
   jupyter notebook
   ```

You can now access Jupyter's web interface by clicking the link that shows up on the terminal or by visiting http://localhost:8888 on your browser. Click on the notebook file (it has a `.ipynb` extension) to open it.

Step: 1

Downloading and Extracting the Dataset

We start the analysis process by first downloading the dataset from Kaggle. I have saved the dataset to my working dicterory mannually and had extracted the dataset to the jupyter notebook by using pandas library function

In [1]:
```
!pip install jovian opendatasets --upgrade --quiet
```

In [1]:
```
import os
import pandas as pd
os.getcwd()
pd.read_csv('country_vaccinations.csv')
```

Out[1]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_
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	
...	...	...	...	...	...	...	...	...	
17602	Zimbabwe	ZWE	2021-05-08	657838.0	509274.0	148564.0	17076.0	19648.0	
17603	Zimbabwe	ZWE	2021-05-09	684243.0	526066.0	158177.0	26405.0	22863.0	
17604	Zimbabwe	ZWE	2021-05-10	690653.0	529360.0	161293.0	6410.0	21877.0	
17605	Zimbabwe	ZWE	2021-05-11	709772.0	539526.0	170246.0	19119.0	21428.0	
17606	Zimbabwe	ZWE	2021-05-12	730365.0	549797.0	180568.0	20593.0	22019.0	

17607 rows × 15 columns

The dataset has been downloaded and extracted.

Let us save and upload our work to Jovian before continuing.

In [6]:
```
project_name = "COVID-19 World Vacination Progress Analysis"
```

In [7]:
```
!pip install jovian --upgrade -q
```

In [8]:
```
import jovian
```

In [9]:
```
jovian.commit(project=project_name)
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
```

Out[9]:
```
'https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis'
```

Step: 2

Data Preparation and Cleaning

** **

Here we do the process of dealing with missing data and filling it with appropriate values. Also understanding the dataset and its contents present in rows and columns

In [2]:
```
dataset_df = pd.read_csv('country_vaccinations.csv')
dataset_df
```

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total
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	

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_
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	
...	...	...	...	...	...	...	...	...	
17602	Zimbabwe	ZWE	2021-05-08	657838.0	509274.0	148564.0	17076.0	19648.0	
17603	Zimbabwe	ZWE	2021-05-09	684243.0	526066.0	158177.0	26405.0	22863.0	
17604	Zimbabwe	ZWE	2021-05-10	690653.0	529360.0	161293.0	6410.0	21877.0	
17605	Zimbabwe	ZWE	2021-05-11	709772.0	539526.0	170246.0	19119.0	21428.0	
17606	Zimbabwe	ZWE	2021-05-12	730365.0	549797.0	180568.0	20593.0	22019.0	

17607 rows × 15 columns

The Overal information contained in the dataset

In [11]:
```
dataset_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17607 entries, 0 to 17606
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              17607 non-null  object
 1   iso_code                             17607 non-null  object
```

```
 2   date                                 17607 non-null  object
 3   total_vaccinations                   10251 non-null  float64
 4   people_vaccinated                    9526 non-null   float64
 5   people_fully_vaccinated              7185 non-null   float64
 6   daily_vaccinations_raw               8568 non-null   float64
 7   daily_vaccinations                   17391 non-null  float64
 8   total_vaccinations_per_hundred       10251 non-null  float64
 9   people_vaccinated_per_hundred        9526 non-null   float64
 10  people_fully_vaccinated_per_hundred  7185 non-null   float64
 11  daily_vaccinations_per_million       17391 non-null  float64
 12  vaccines                             17607 non-null  object
 13  source_name                          17607 non-null  object
 14  source_website                       17607 non-null  object
dtypes: float64(9), object(6)
memory usage: 2.0+ MB
```

Checking No.of.Rows and columns

In [37]:
```
dataset_df.shape
```

Out[37]:
```
(17607, 15)
```

Summing the missing values

In [134…
```
dataset_df.isna().sum()
```

Out[134…
```
country                                  0
iso_code                                 0
date                                     0
total_vaccinations                    7356
people_vaccinated                     8081
people_fully_vaccinated              10422
daily_vaccinations_raw                9039
daily_vaccinations                     216
total_vaccinations_per_hundred        7356
people_vaccinated_per_hundred         8081
people_fully_vaccinated_per_hundred  10422
daily_vaccinations_per_million         216
vaccines                                 0
source_name                              0
source_website                           0
dtype: int64
```

Checking the data types contained in the dataset

In [39]:
```
dataset_df.dtypes
```

Out[39]:
```
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
```

Converting 'Date' Column to Date format

In [3]:
```
dataset_df["date"] = pd.to_datetime(dataset_df.date)
```

In [4]:
```
dataset_df.dtypes
```

Out[4]:
```
country                                object
iso_code                               object
date                           datetime64[ns]
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
```

Finally, Discriptive Statistics about the dataset

In [42]:
```
dataset_df.describe()
```

Out[42]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	peo
count	1.025100e+04	9.526000e+03	7.185000e+03	8.568000e+03	1.739100e+04	10251.000000	
mean	5.716299e+06	3.500000e+06	1.817309e+06	1.445607e+05	7.941384e+04	17.199317	
std	2.399767e+07	1.313534e+07	7.875693e+06	5.906624e+05	3.837504e+05	24.719775	
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	
25%	6.450950e+04	5.595875e+04	2.520600e+04	3.176000e+03	8.520000e+02	1.500000	
50%	4.701090e+05	3.617215e+05	1.920300e+05	1.674100e+04	5.974000e+03	7.170000	
75%	2.154874e+06	1.491252e+06	7.725270e+05	6.590250e+04	3.005350e+04	22.565000	
max	3.669100e+08	1.546242e+08	1.189873e+08	1.263800e+07	9.882286e+06	220.400000	

In [ ]:
```
import jovian
```

In [ ]:
```
jovian.commit()
```

Step: 3

Exploratory Analysis and Visualization

** **

Let's Analysis the Dataset!

Here we are exploring the data and trying to understand what the dataset's columns and rows contains and what analysis can be done through them. We use certain libraries like seaborn and matplot which are libraries of python to perfrom the analysis tasks and to visualize it using different charts and graphs

Let's begin by importing `matplotlib.pyplot` and `seaborn`.

In [5]:
```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

Number of countries present in the dataset

In [6]:
```
print("The Total number of Countries present in the dataset is {}".format(dataset_df.country.nunique()))
```

```
The Total number of Countries present in the dataset is 211
```

Top countries with maximum vaccinations

In [7]:
```
country_tot_vacc = dataset_df.groupby('country')[['total_vaccinations']].max()
top20_vacc = country_tot_vacc.sort_values('total_vaccinations',ascending=False).head(20)
top20_vacc
```

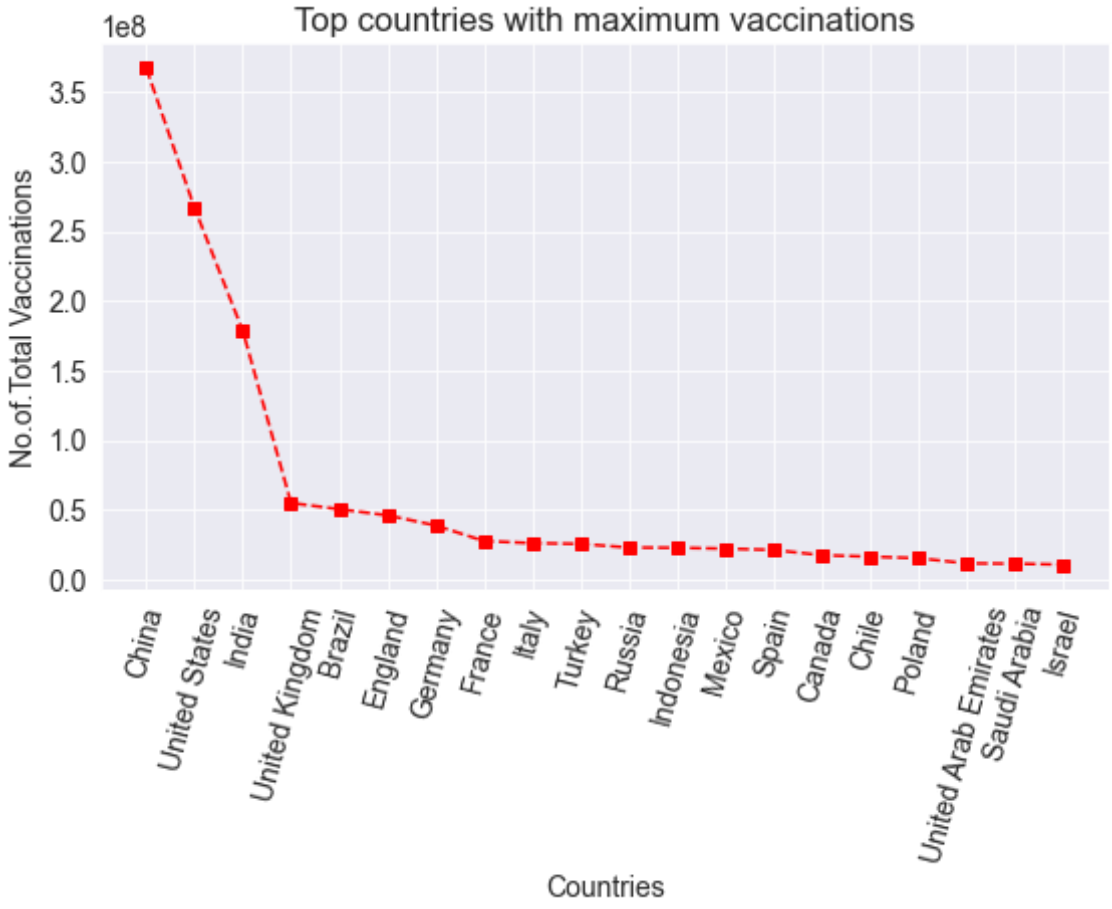
Out[7]:

country	total_vaccinations
China	366910000.0
United States	266596486.0
India	178361846.0
United Kingdom	54797640.0
Brazil	50308106.0

country	total_vaccinations
England	45908796.0
Germany	38646171.0
France	27455748.0
Italy	25948925.0
Turkey	25402277.0
Russia	22782931.0
Indonesia	22617205.0
Mexico	21986456.0
Spain	21071940.0
Canada	17297879.0
Chile	16246599.0
Poland	15144771.0
United Arab Emirates	11422565.0
Saudi Arabia	11195164.0
Israel	10525163.0

Visualization of top countries with maximum vaccinations

In [8]:
```
plt.plot(top20_vacc,'s--r');
plt.xticks(rotation=75);
plt.title("Top countries with maximum vaccinations");
plt.xlabel('Countries');
plt.ylabel('No.of.Total Vaccinations');
```



Number of various vaccines in the world till date

In [9]:
```
print("Number of various vaccanies given across the world are {}"
      .format(dataset_df.vaccines.nunique()))
```

```
Number of various vaccanies given across the world are 41
```

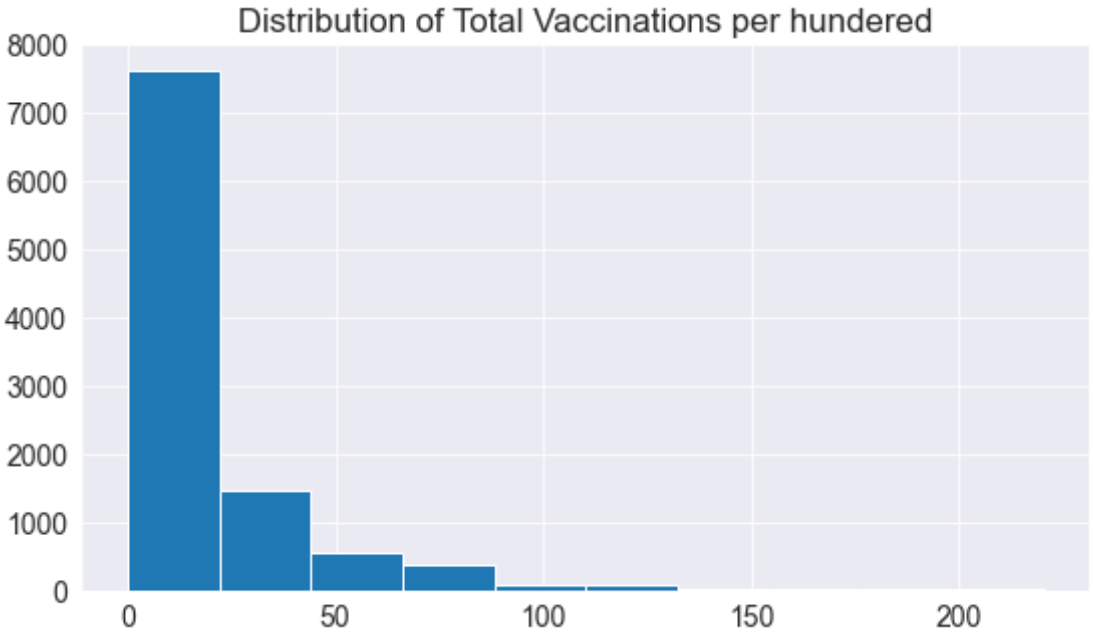
Average number of people vaccinated

In [10]:
```
print("Average number of people vaccinated across the world are {}" .format(dataset_df.people_vaccinated.mean()))
```

```
Average number of people vaccinated across the world are 3499999.948981734
```

Distribution of Total Vaccinations per hundered

In [11]:
```
plt.title('Distribution of Total Vaccinations per hundered')
plt.hist(dataset_df.total_vaccinations_per_hundred);
```



Count of various vaccinations in the dataset

In [12]:
```
dataset_df['vaccines'].value_counts()
```

Out[12]:

Oxford/AstraZeneca	3146
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	2265
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	1896
Oxford/AstraZeneca, Pfizer/BioNTech	1412
Pfizer/BioNTech	1184
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	892
Moderna, Pfizer/BioNTech	799
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V	775
Oxford/AstraZeneca, Sinopharm/Beijing	667
Sputnik V	499

```
Oxford/AstraZeneca, Sinovac                                                    316
Sinopharm/Beijing                                                              292
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing                         271
Pfizer/BioNTech, Sinopharm/Beijing                                             208
Pfizer/BioNTech, Sinovac                                                       202
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V                        199
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac                                 197
Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac                                    150
EpiVacCorona, Sputnik V                                                        150
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V                               148
Johnson&Johnson, Moderna, Pfizer/BioNTech                                      145
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V               140
Oxford/AstraZeneca, Sputnik V                                                  139
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V     137
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing                                 130
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V    129
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech                           127
Covaxin, Oxford/AstraZeneca                                                    120
Moderna, Oxford/AstraZeneca                                                    118
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V                      108
Sinopharm/Beijing, Sputnik V                                                   103
CanSino, Sinopharm/Beijing, Sinovac, Sputnik V                                 101
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V                       92
Johnson&Johnson                                                                 87
Pfizer/BioNTech, Sputnik V                                                      75
Oxford/AstraZeneca, Sinovac, Sputnik V                                          73
Pfizer/BioNTech, Sinovac, Sputnik V                                             60
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V                       29
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V        24
EpiVacCorona, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V                   1
Abdala                                                                           1
Name: vaccines, dtype: int64
```

In [13]:
```
print("Total number of vaccinations in the dataset are: {}"
      .format(dataset_df['vaccines'].value_counts().sum()))
```

```
Total number of vaccinations in the dataset are: 17607
```

Average number of vaccinations produced by manufactures

Here we extra the file called: `country_vaccinations_by_manufacturer.csv` from the working directory, which is originally downloaded from kaggle.

In [14]:
```
vacc_dataset = pd.read_csv('country_vaccinations_by_manufacturer.csv')
```

In [15]:
```
vacc_dataset.total_vaccinations.mean()
```

Out[15]:
```
5168580.8177521005
```

In [22]:
```
import jovian
```

In [23]:
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
```

Out[23]:
```
'https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis'
```

Step: 4

Asking and Answering Questions

** **

Here we try asking and solving various questions that can leads us to draw some conclusions and inferences about the COVID-19 Vaccination progress dataset

Q1: Which vaccine people had taken the most?

In [16]:
```
tot_vacc = dataset_df.groupby('vaccines')[['total_vaccinations']].max()
```

In [17]:
```
counts_vacc = tot_vacc.sort_values('total_vaccinations',ascending=False)
counts_vacc
```

Out[17]:

vaccines	total_vaccinations
Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac	366910000.0
Johnson&Johnson, Moderna, Pfizer/BioNTech	266596486.0
Covaxin, Oxford/AstraZeneca	178361846.0
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	54797640.0
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	50308106.0
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	38646171.0
Pfizer/BioNTech, Sinovac	25402277.0
EpiVacCorona, Sputnik V	22782931.0
Oxford/AstraZeneca, Sinovac	22617205.0
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	21986456.0
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V	11422565.0
Oxford/AstraZeneca, Pfizer/BioNTech	11195164.0
Moderna, Pfizer/BioNTech	10525163.0
Oxford/AstraZeneca, Sinopharm/Beijing	10436046.0
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V	9541511.0
Oxford/AstraZeneca	9316086.0
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V	7110455.0
Pfizer/BioNTech	5593436.0
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V	3915922.0
CanSino, Sinopharm/Beijing, Sinovac, Sputnik V	3836291.0
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac	3132686.0
Oxford/AstraZeneca, Sinovac, Sputnik V	2542066.0
Sputnik V	2455655.0
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing	2261096.0

vaccines	total_vaccinations
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V	1767570.0
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V	1371976.0
Pfizer/BioNTech, Sinopharm/Beijing	1091048.0
Oxford/AstraZeneca, Sputnik V	930460.0
Sinopharm/Beijing	730365.0
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	632676.0
Sinopharm/Beijing, Sputnik V	563466.0
Pfizer/BioNTech, Sinovac, Sputnik V	537380.0
Johnson&Johnson	455169.0
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	400004.0
Moderna, Oxford/AstraZeneca	245772.0
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing	220646.0
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V	106559.0
Abdala	70000.0
EpiVacCorona, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V	41993.0
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V	38151.0
Pfizer/BioNTech, Sputnik V	36735.0

From the observation above, People had taken `Sinopharm` Vaccine the most across the world, when compared to other vaccines present

Q2:What are the Countries who have Sinopharm vaccines?

In [18]:
```
vacc_country = dataset_df.groupby('country')[['vaccines']].max()
```

In [19]:
```
vacc_country[vacc_country.vaccines=='Sinopharm/Beijing']
```

Out[19]:

country	vaccines
Cameroon	Sinopharm/Beijing
Equatorial Guinea	Sinopharm/Beijing
Gabon	Sinopharm/Beijing
Mauritania	Sinopharm/Beijing
Senegal	Sinopharm/Beijing
Zimbabwe	Sinopharm/Beijing

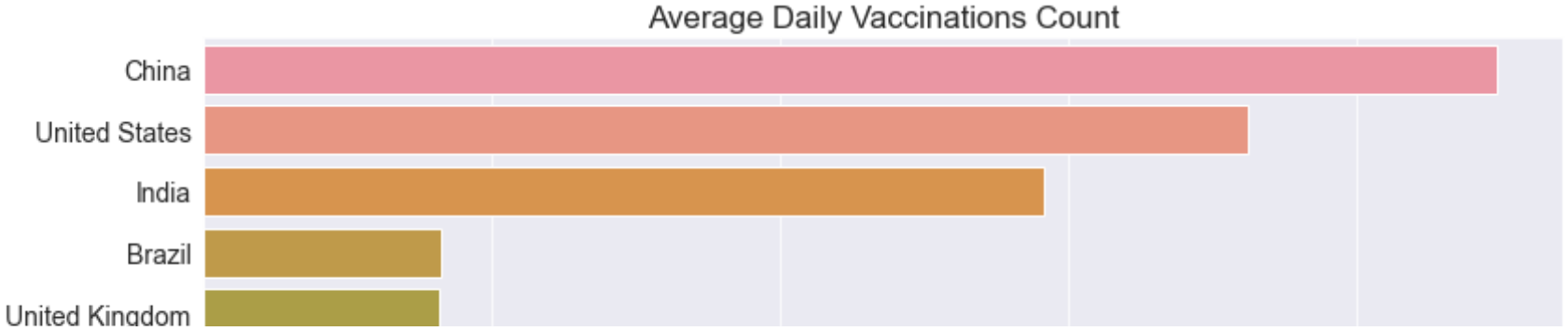
These are six countries with Sinopharm/Beijing vaccines

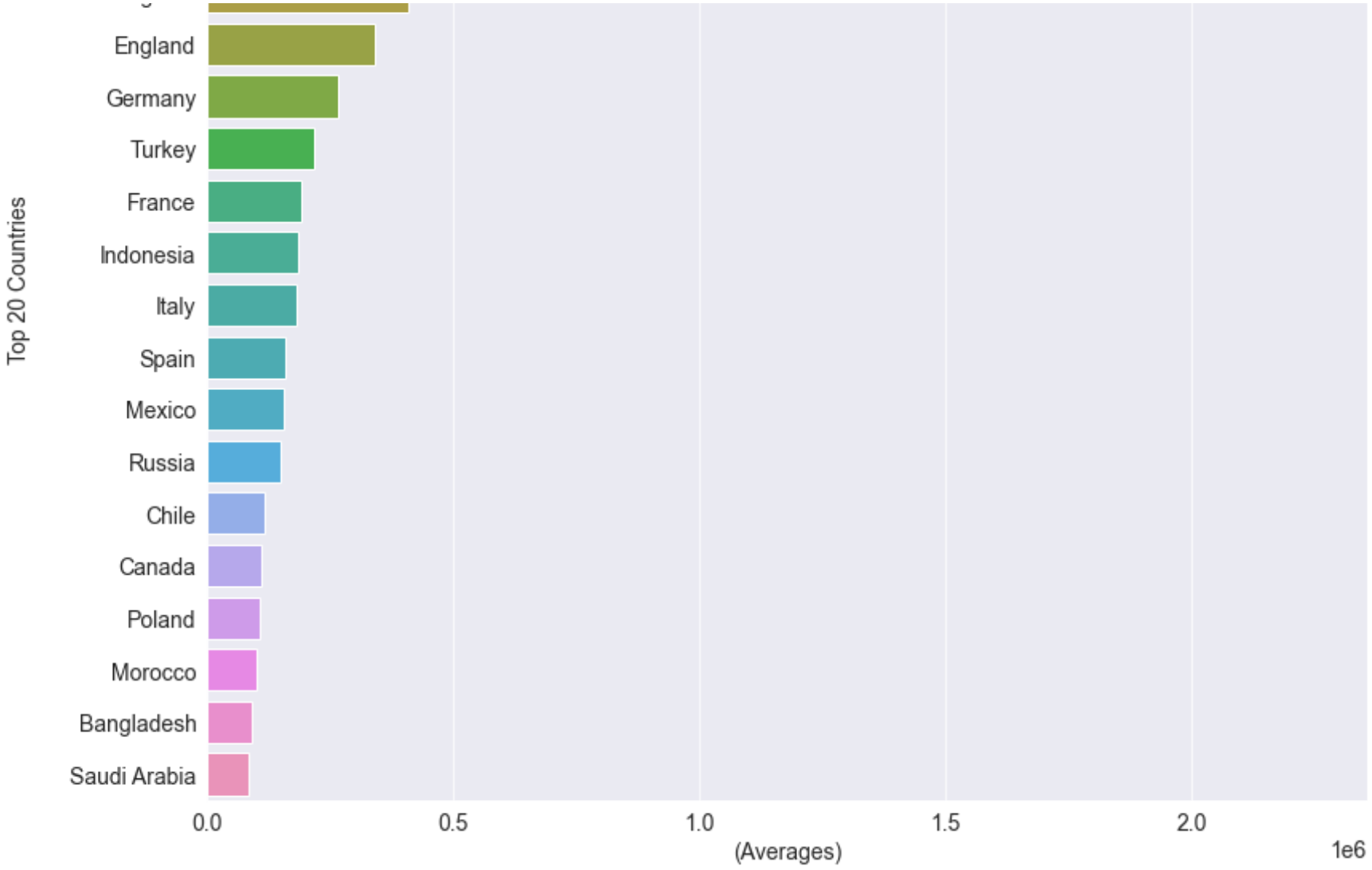
Q3: What are the Top 20 Countries with best daily average vacinations?

In [20]:
```
a= dataset_df.groupby("country").daily_vaccinations.mean().sort_values(ascending= False).head(20)
```

In [21]:
```
plt.figure(figsize= (13,12));
sns.barplot(a.values,a.index);
plt.title("Average Daily Vaccinations Count");
plt.xlabel("(Averages)");
plt.ylabel("Top 20 Countries");
```

```
C:\Users\Kiran\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(
```





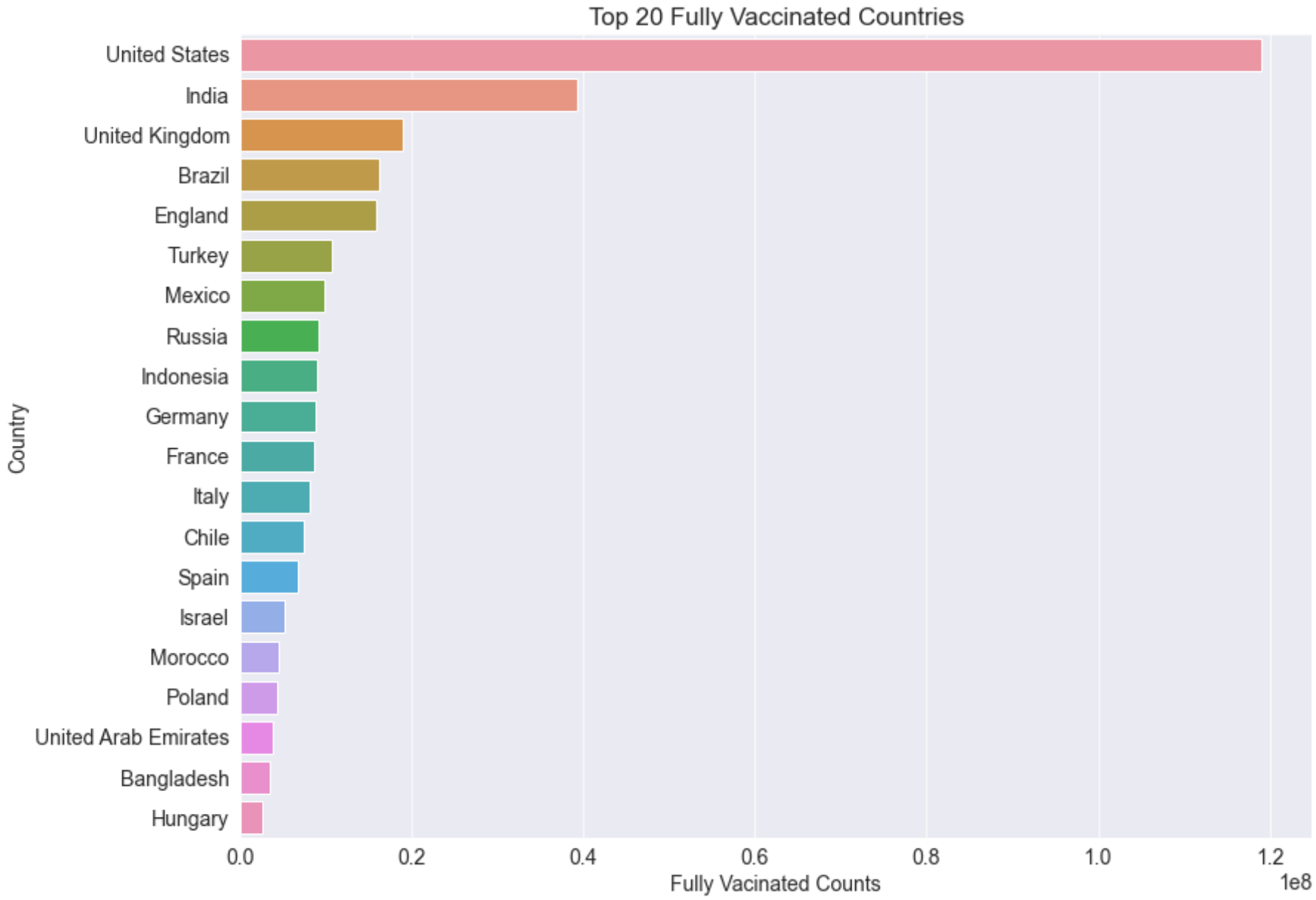
Q4: What are the top Countries with fully vaccinated people?

In [22]:
```
top_fully_vacc = dataset_df.groupby("country").people_fully_vaccinated.max().sort_values(ascending= False).head(20)
```

In [23]:
```
plt.figure(figsize=(13,10))
sns.barplot(top_fully_vacc.values,top_fully_vacc.index);
```

```
plt.title('Top 20 Fully Vaccinated Countries');
plt.xlabel('Fully Vacinated Counts');
plt.ylabel('Country');
```

```
C:\Users\Kiran\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(
```



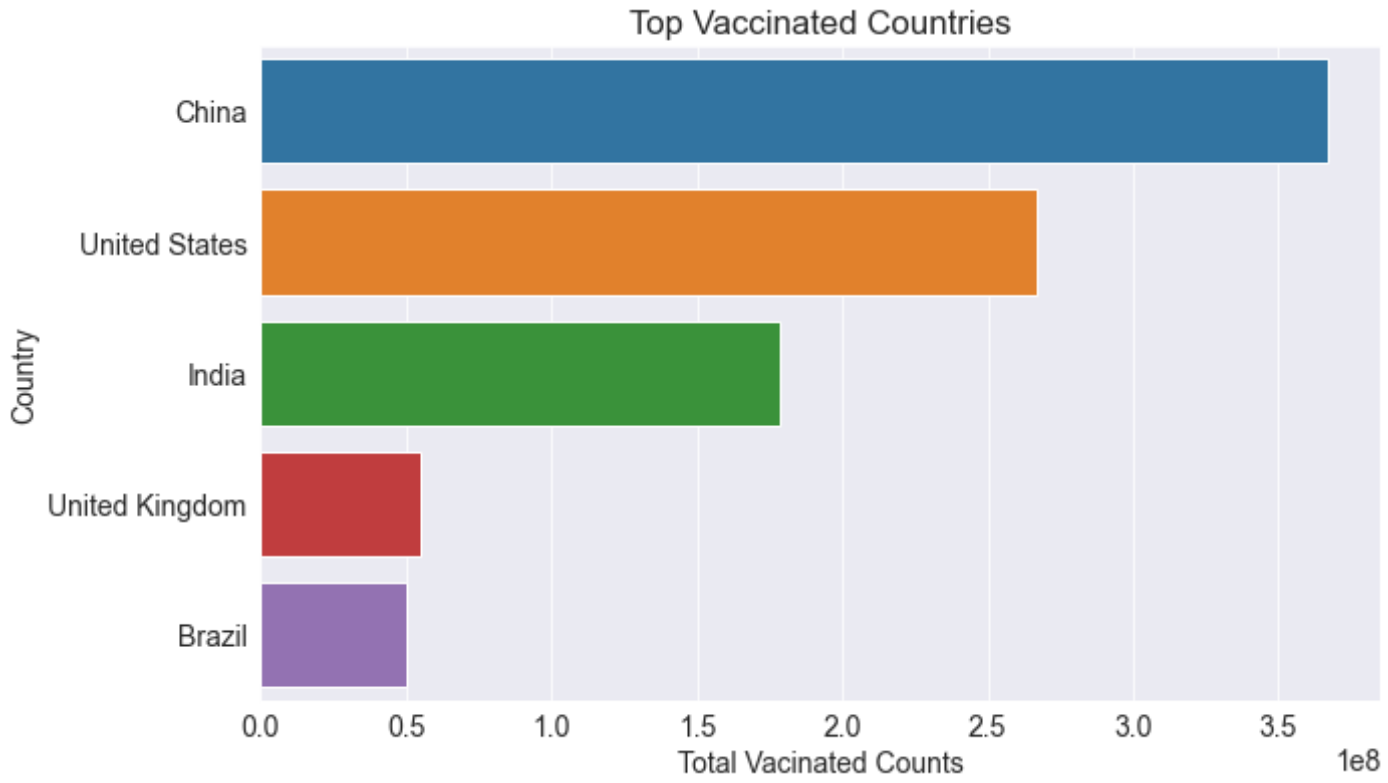
Q5: Top countries those got vaccinated the most?

In [24]:
```
top_vacci = dataset_df.groupby("country")['total_vaccinations'].max().sort_values(ascending= False).head(5)
top_vacci
```

Out[24]:
```
country
China             366910000.0
United States     266596486.0
India             178361846.0
United Kingdom     54797640.0
Brazil             50308106.0
Name: total_vaccinations, dtype: float64
```

In [25]:
```
plt.figure(figsize=(10,6))
sns.barplot(top_vacci.values,top_vacci.index);
plt.title('Top Vaccinated Countries');
plt.xlabel('Total Vacinated Counts');
plt.ylabel('Country');
```

```
C:\Users\Kiran\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(
```



Let us save and upload our work to Jovian before continuing.

In [164…
```
import jovian
```

In [165…
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
```

Out[165…
```
'https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis'
```

Inferences and Conclusion

** **

We reached the end!

We can conclude from the above questions with various points;

1. **Our world has total `41 various vaccines` that are consumed across different countries around the world**
2. **The average count of number of vaccinations taken by people around the world is `39,99,999` till date**
3. **The most consumed vaccine across the world is `"Sinopharm"` vaccine**
4. **Out of 211 countries that are getting vaccinations, only six countries contribute to taking Sinopharm vaccine, the most consumed vaccine**
5. **The top countries providing daily best average vaccinations are:**
   - **China**
   - **USA**
   - **India**
   - **Brazil**
   - **United Kingdom**
   - **England**
   - **Germany**
6. **Top vaccinated countries are:**
   - **China**
   - **USA**
   - **India**
   - **United Kingdom**
   - **Brazil**
7. **China is best in daily average vaccines provider and has also topped the list in most vaccinated countries, but surprisingly they do not have any data for those who have got fully vaccinated i.e two dosage of vaccine, in China.**

In [166…

```
import jovian
```

In [167…
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
```

Out[167…
```
'https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis'
```

References and Future Work

** **

Ideas for Future Projects using this dataset

Some of the other intersting facts people can work on this dataset includes;

1. **analysisng the trend of number of vaccinations to occurr in next 7 days or a month**
2. **Can perfrom Regression analysis to predict a particular countries vaccination progress**

Try exploring the dataset: https://www.kaggle.com/gpreda/covid-world-vaccination-progress

In [168…
```
import jovian
```

In [169…
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
```

Out[169…
```
'https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis'
```

In [171…
```
jovian.submit(assignment="zero-to-pandas-project")
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kiranprasanth01/covid-19-world-vacination-progress-analysis" on https://jovian.ai/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ai/kiranprasanth01/covid-19-world-vacination-progress-analysis
[jovian] Submitting assignment..
[jovian] Verify your submission at https://jovian.ai/learn/data-analysis-with-python-zero-to-pandas/assignment/course
-project
```