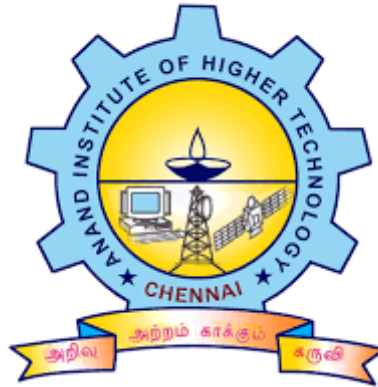# ANAND INSTITUTE OF HIGHER TECHNOLOGY
OLD MAHABALIPURAM ROAD,
KALASALINGAM NAGAR,
KAZHIPATTUR- 603103

IBM - NAAN MUDHALVAN
DATA ANALYTICS WITH COGNOS
COVID VACCINES ANALYSIS

PHASE – 5

NAME   **:**   **SHALINI S**

REG No.   **:**   **310121104095**

BRANCH   **:**   **COMPUTER SCIENCE & ENGINEERING**

YEAR/SEM   **:**   **III / V**

COVID VACCINES ANALYSIS

INTRODUCTION

- ❖ An introduction to the analysis of covid-19 vaccines provides an overview of critical processes and considerations involved in evaluating these crucial tools in the fight against the pandemic.
- ❖ Covid -19 vaccines have been at the forefront of global healthcare efforts, offering hope for controlling the spread of the virus and mitigating its impact on public health. The analysis of these vaccines encompasses a multifaceted approach, combining rigorous clinical trials, real-world effectiveness studies, safety monitoring and logistical assessments to ensure their successful deployment.
- ❖ The COVID-19 pandemic, caused by the novel coronavirus SARS-CoV-2, has taken the world by storm, leading to widespread illness, loss of life, and societal disruption.
- ❖ In response to this global health crisis, the development and deployment of COVID-19 Vaccines have played a pivotal role in controlling the spread of the virus.
- ❖ Analyzing COVID-19 Vaccines is a critical aspect of managing and mitigating the impact of the global pandemic. Such analysis encompasses various areas including: Vaccine Development, Efficacy and Effectiveness, Safety and Side Effects, Vaccine Distribution and Access, Variants and Vaccine Adaptations, Vaccine Hesitancy, Global Vaccination Strategies, Economic Impact and Future Preparedness.
- ❖ These analyses are crucial for optimizing vaccination strategies ensuring public health, and advancing our understanding of vaccine development and deployment in the face of global health crisis.
- ❖ The COVID-19 pandemic has spurred unprecedented efforts in vaccine development and distribution. As vaccines are administered to millions of people worldwide, it is crucial to monitor and optimize the distribution process while closely monitoring adverse effects. Advanced machine learning techniques can play a pivotal role in achieving these goals.
- ❖ The pace of the COVID-19 Vaccine development process is unprecedented and is challenging the traditional paradigm of vaccinology science.

- ❖ The main pressure comes from the pandemic situation, but what makes it possible is a complex set of factors and innovative environments built along the times, which this manuscript aims to study.
- ❖ The world has witnessed an unprecedented series of events triggered by the pandemic of COVID-19(coronavirus disease),a disease caused by SARS-CoV-2, a new virus belonging to the Coronavideae family, of great impact on individual and collective health worldwide, and high impact implications for the global economy.
- ❖ On the other hand, it is possible to identify positive aspects in facing the pandemic, ranging from humanitarian solidarity aid actions to accelerating strategies for the development of vaccines, which assumes the position of main hope in solving this problem of global scope.
- ❖ The COVID-19 pandemic has spurred unprecedented efforts in vaccine development and distribution. As vaccines are administered to millions of people worldwide, it is crucial to monitor and optimize the distribution process while closely monitoring adverse effects. Advanced machine learning techniques can play a pivotal role in achieving these goals.

Covid-19 vaccines are designed to stimulate the body's immune system to recognize and fight the virus. There are currently three types of vaccines available : mRNA, vector and inactivated vaccines. Each vaccine works differently but they all aim to provide immunity against covid-19.

WHY VACCINE IS IMPORTANT ?

Vaccination is important because it helps to prevent the spread of covid-19. It also reduces the severity of the disease if someone does get infected. This helps to protect individuals and communities.

TYPES OF COVID -19 VACCINES :

There are several type of covid-19 vaccines, including mRNA vaccines , viral vector vaccines, and protein subunit vaccine. They all work by teaching the immune system to recognize and fight the disease.

EFFICACY OF COVID-19 VACCINES :

Clinical traits have shown that covid-19 vaccines are highly effective in preventing infection and severe illness. The pfizer-BioNTech vaccine has an efficacy rate of 95% , while the Moderna vaccine has an efficacy rate of 94.1% , the Johnson & Johnson vaccine has an efficacy rate of 66.3% against moderate to severe diseases.

SAFETY OF COVID-19 VACCINES :

Covid-19 vaccines have undergone rigorous testing to ensure their safety. The most common side effects are mild and include pain at the injection site, fever and fatigue. Serious side effects are rare but have been reported. It is important to discuss any concerns with a health care provider.

SIDE EFFECTS OF COVID-19 VACCINES :

Covid-19 vaccines can cause side effects, but they are generally mild and go away quickly. The most common side effects include pain at the injection site , fever and fatigue.

VACCINE DISTRIBUTION STRATEGIES :

The distribution of covid-19 vaccines has been a complex and challenging task. Strategies have included prioritizing high risk populations , setting up mass vaccination sites, and partnering with community organizations to reach underserved communities.

COVID VACCINES ANALYSIS PROCESSES

- Start building & begin conducting the COVID VACCINES ANALYSIS by collecting and preprocessing the dataset and performing exploratory data analysis.
- Data preprocessing is a critical step in the analysis of COVID-19 Vaccine data, as it lays the foundation for extracting meaningful insights and patterns from the vast and diverse sources of information related to the pandemic.
- This process involves collecting, cleaning, transforming, reduction of null values, visualization, scalability, efficiency and structuring raw data to make it suitable for analysis.
- The goal of COVID-19 Vaccine Analysis is to prepare the raw data for analysis, modelling, and decision making.
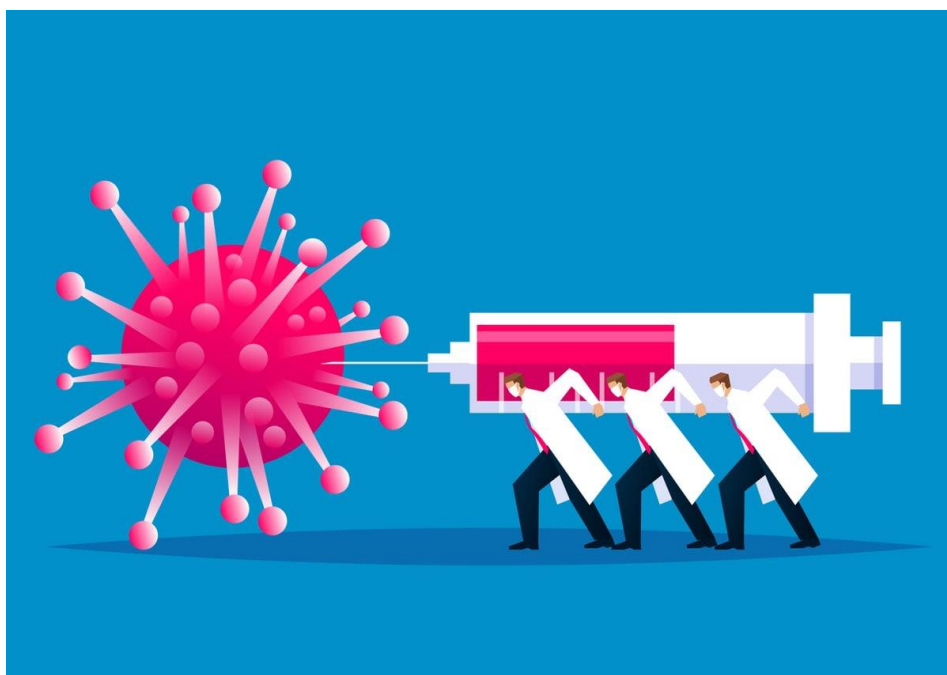
Importance of Covid 19 Vaccine Analysis

The importance of COVID-19 vaccines lies in their significant role in mitigating the impact of the pandemic on individuals, communities, and global health. Here is an analysis of the key reasons why COVID-19 vaccines are crucial:

Preventing Severe Illness and Death:

COVID-19 vaccines have proven to be highly effective in preventing severe illness, hospitalization, and death caused by the virus. By reducing the severity of the disease, vaccines have played a critical role in saving lives and alleviating the burden on healthcare systems.

Facilitating Economic Recovery:

The widespread administration of COVID-19 vaccines is instrumental in reopening economies. Vaccination campaigns contribute to a decrease in the number of cases, allowing businesses to operate more freely, reducing the need for lockdowns, and supporting economic recovery.

Global Public Health:

COVID-19 vaccines are a critical tool in global public health efforts. Controlling the spread of the virus internationally is essential to preventing new variants and ensuring that countries can collaborate in managing and overcoming the pandemic.

Supporting Education and Normalcy:

Vaccination has played a role in the safe reopening of schools and the return to more normal social and economic activities. This is particularly important for the well-being of children and young adults, ensuring the continuity of education and social development.

Advancements in Vaccine Technology:

The development and deployment of COVID-19 vaccines have accelerated advancements in vaccine technology. This experience may have broader implications for future vaccine development, including addressing other infectious diseases.

DATA COLLECTION :

COVID VACCINES ANALYSIS is done by using the Dataset of **"COVID-19 World Vaccination Progress"** provided by the dataset site

www.Kaggle.com

DATASET:

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

DATASET AND ITS DETAILS :

The dataset "COVID-19 World Vaccination Progress" on Kaggle is a collection of data related to the COVID-19 Vaccination efforts worldwide. It provides information about the progress of COVID-19 Vaccinations in various countries and regions. This dataset is designed to help researchers, data scientists, and analysts understand and analyze the progress of COVID-19 Vaccination campaigns across different countries. A second file, with manufacturers information is included. Below is a detailed overview of the dataset:

TITLE: COVID-19 World Vaccination Progress

DATASET ID: gpreda/covid-world-vaccination-progress

SOURCE: The dataset was created by a Kaggle user named Gabriel Preda, collected from various sources, including government health agencies, international organizations, and research institutions.

DESCRIPTION:
1. The dataset provides information about the COVID-19 Vaccination progress from various countries around the world.
2. It includes data on vaccine distribution, vaccination coverage, and other related statistics.
3. The dataset may include information about the types of vaccines used, vaccination rates over time, and population demographics.

COLUMNS/ATTRIBUTES:
1.The dataset typically contains columns such as country, iso_code, date, total_vaccinations, people_vaccinated, people_fully_vaccinated, daily_vaccinations_raw, daily_vaccinations, and more.

2.These columns provide information about the total number of vaccinations, daily vaccination rates, and other vaccination-related metrics for each country.

USAGE:
1.Analyzing vaccination progress over time for different countries.

2.Identifying countries with high vaccination rates or disparities.

3.Forecasting future vaccination trends.

4.Studying the impact of different vaccines on vaccination rates.

5.Correlating vaccination progress with COVID-19 infection and mortality rates.

DATA FORMAT:
The data is usually structured as a CSV (Comma-Separated Values) file, with rows representing different countries or regions and columns representing various attributes related to vaccination progress and population.

UPDATES:
The dataset may be updated regularly to reflect the latest vaccination data, making it useful for tracking changes and trends over time.
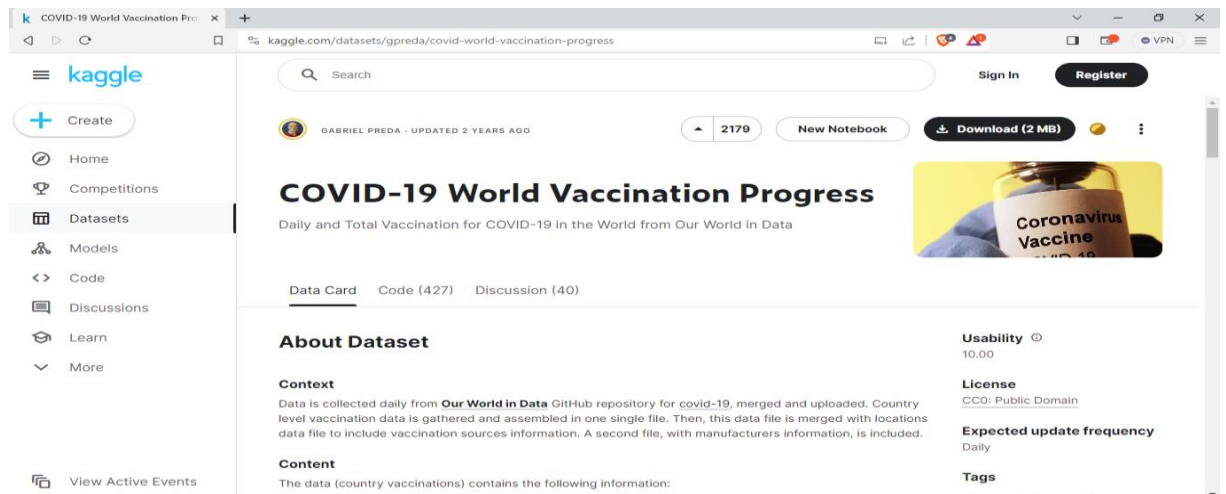
COLUMNS:
• Country- this is the country for which the vaccination information is provided.
• Country ISO Code - ISO code for the country.
• Date - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total.

- <u>Total number of vaccinations</u> - this is the absolute number of total immunizations in the country. 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 vaccinations might be larger than the number of people.
- <u>Total number of people fully vaccinated</u> - 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.
- <u>Daily vaccinations (raw)</u> - for a certain data entry, the number of vaccinations for that date/country.
- <u>Daily vaccinations</u> - for a certain data entry, the number of vaccinations for that date/country.
- <u>Total vaccinations per hundred</u> - ratio (in percent) between vaccination number and total population up to the date in the country.
- <u>Total number of people vaccinated per hour</u>- ratio (in percent) between population immunized and total population up to the date in the country.
- <u>Total number of people fully vaccinated per hundred</u> - ratio (in percent) between population fully immunized and total population up to the date in the country.
- <u>Number of vaccinations per day</u> - number of daily vaccinations for that day and country.
- <u>Daily vaccinations per million</u> - ratio (in ppm) between vaccination number and total population for the current date in the country.
- <u>Vaccines used in the country</u> - total number of vaccines used in the country (up to date).
- <u>Source name</u> - source of the information (national authority, international organization, local organization etc.).
- <u>Source website</u> - website of the source of information.

There is a second file added (country vaccinations by manufacturer), with the following columns:
- Location - country.
- Date - date.
- Vaccine - vaccine type.
- Total number of vaccinations - total number of vaccinations / current time and vaccine type.
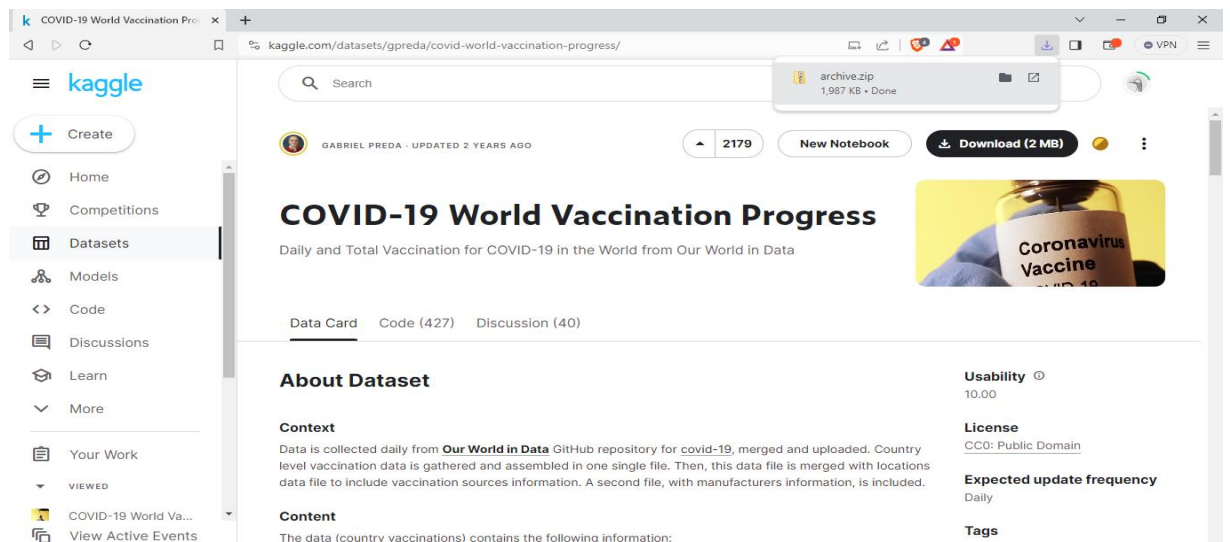
1. **VISIT THE KAGGLE WEBSITE FOR COVID-19 WORLD VACCINATION POGRESS DATASET :**
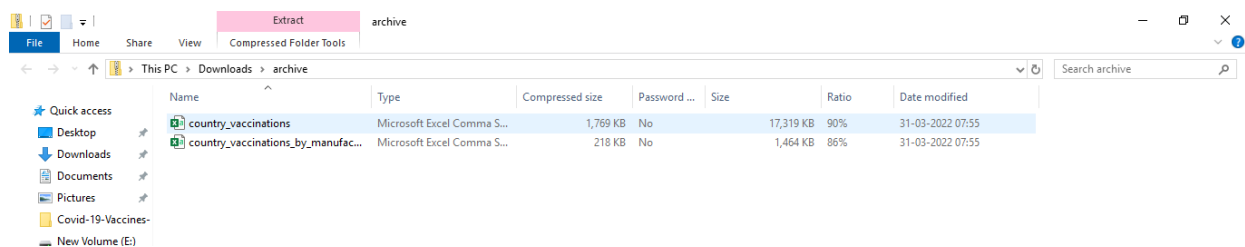
2. **DOWNLOAD THE DATASET FROM THE KAGGLE WEBSITE :**

3. **EXTRACT THE TWO CSV FILES FROM THE ARCHIVE ZIP FOLDER :**

4. DATASET : country_vaccinations.csv

5. DATASET: country_vaccinations_by_manufacturer.csv

6. **DATA LOADING :**

   Steps Involved in data loading on IBM cognos.

   - Login to your IBM cognos.

   - Click more menu from the left side.

   - Select new tab.
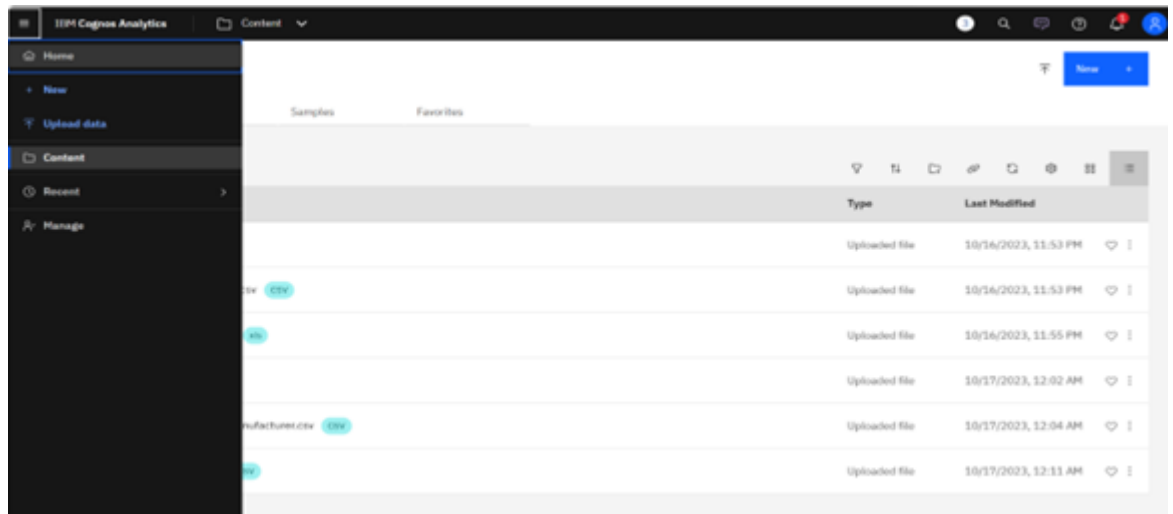
7. **CLICK DATA MODULE TAB :**
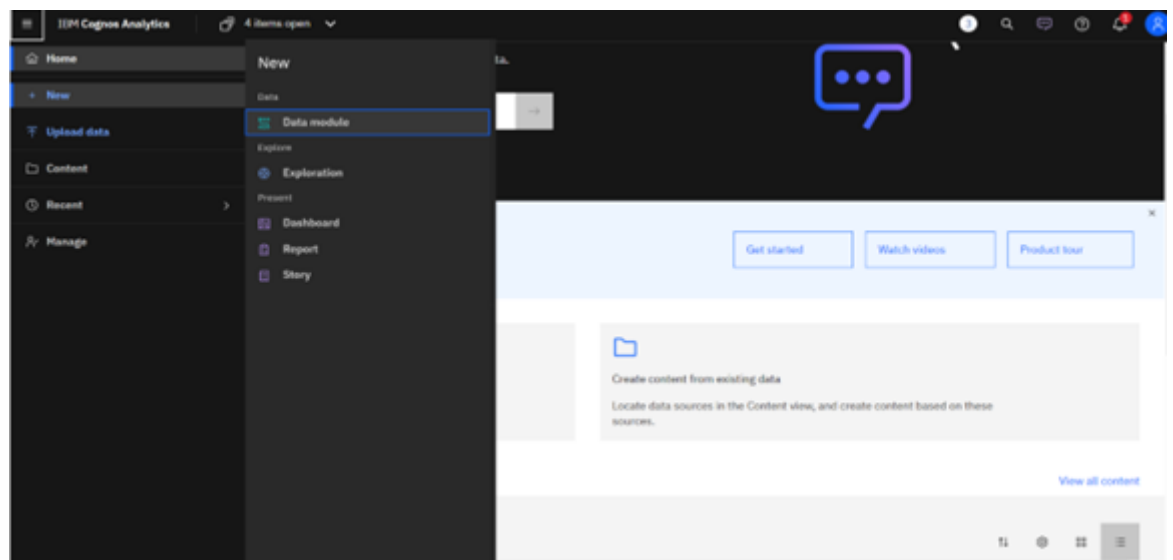
8. **UPLOAD THE DATASET FOR YOUR PROJECT AND SELECT THE CORRESPONDING FILE：**

9. PREVIEW THE DATA :

10. EXPLORE THE DATA :

11. SAVE THE DATA MODULE :

12. DATA PREPROCESSING AND CLEANING :

- Handling missing data.
- Data Transformation.
- Data Type Conversion.
- Removing Duplicates.
- Dealing with Outliers Once you saved the data module.
- Click the corresponding dataset on IBM cognos and Preview the module Right.
- Click the row where you want to clean the data . It provides the UI to Clean the data and makes the task easy one, Now Updating and Replacing the Null values are simple.
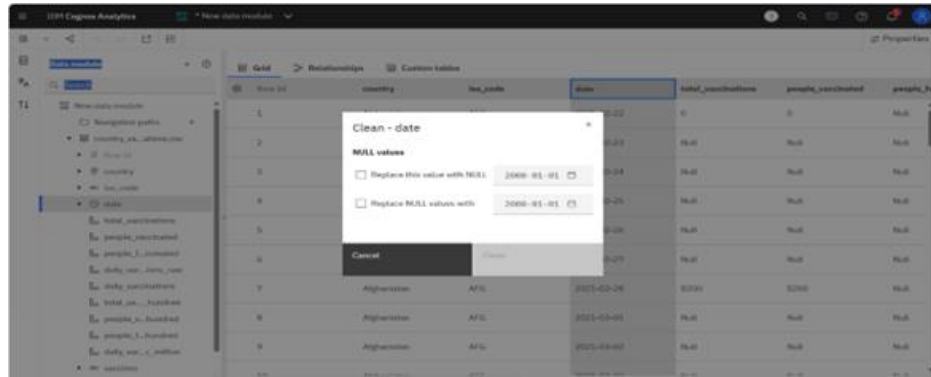
Data module will be updated by doing the above process
after the completion of process, start creating the dashboard for Visualization.

13. DASHBOARD CREATION :

Dashboard creation are helpful for visualizing the data
- Goto Home menu
- Select the new tab
- Click dashboard
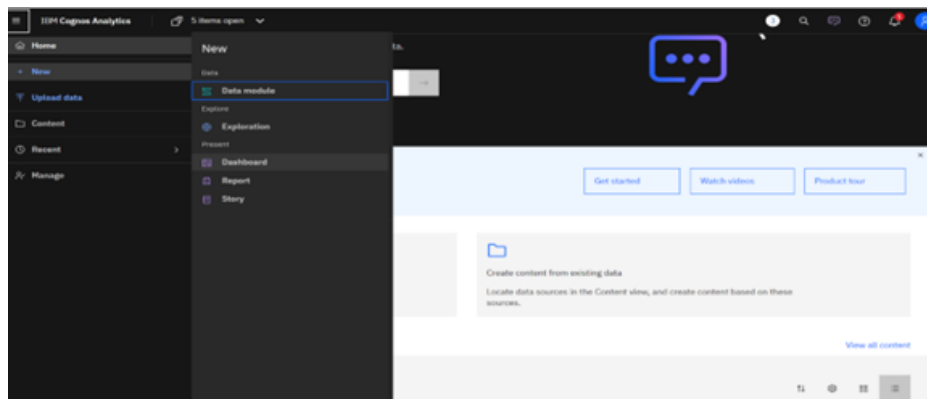
14. CHOOSE THE TEMPLATE FOR YOUR PROJECT :
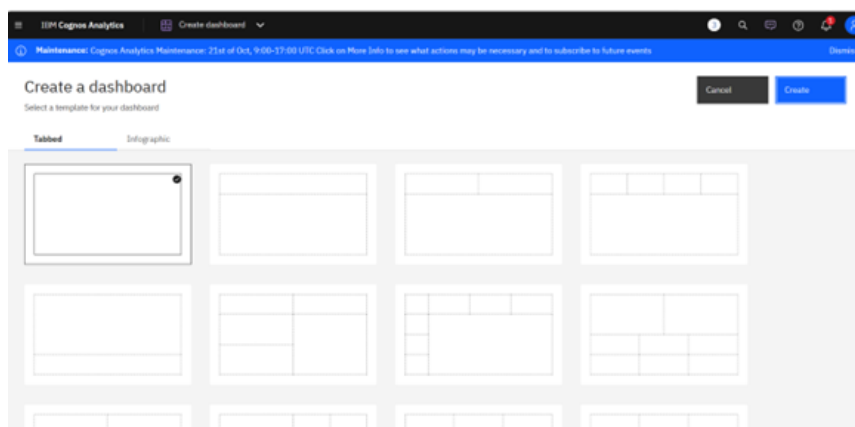
15. NOW THE DASHBOARD IS CREATED :
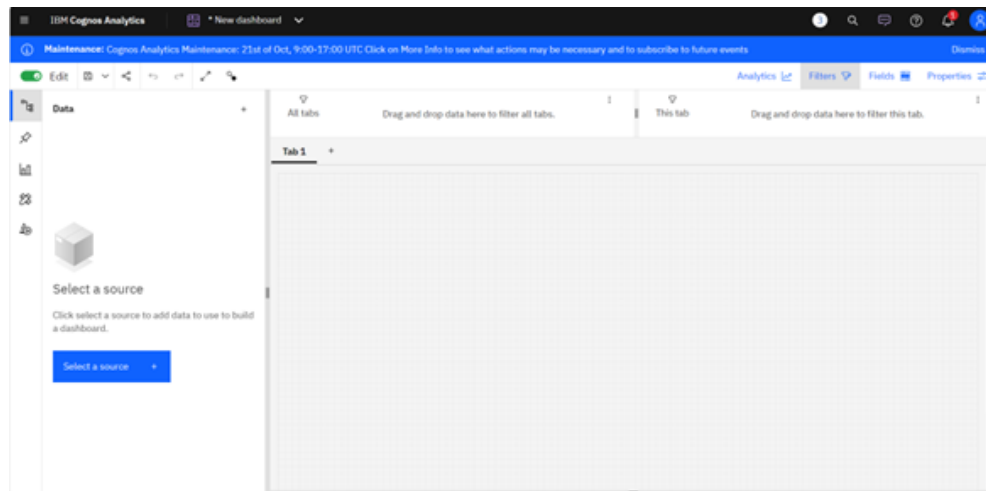
16. SELECT THE DATA SOURCE :
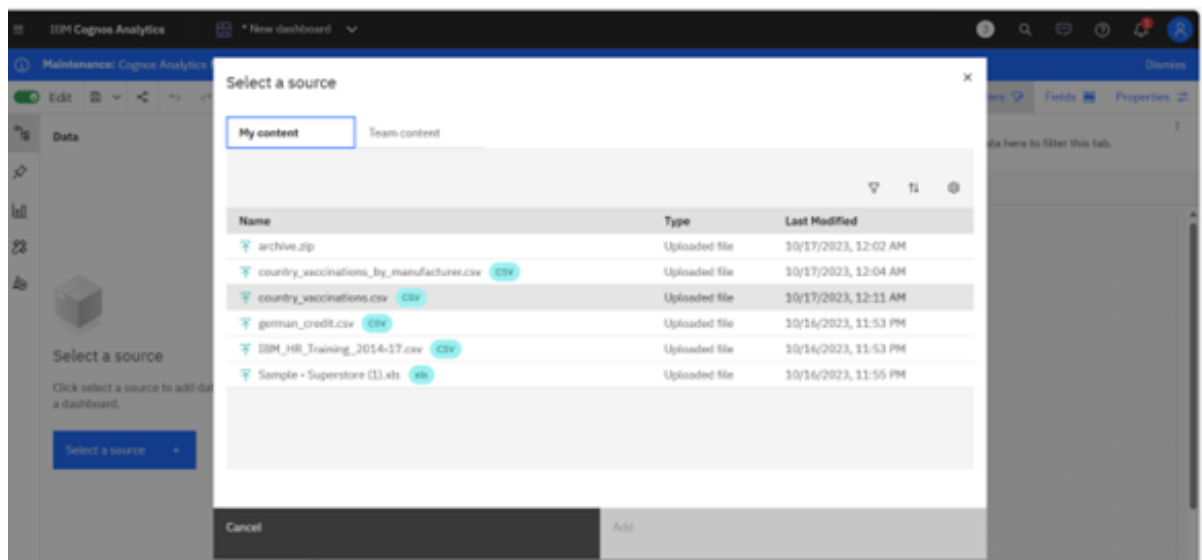
17. VISUALIZATION :
After creating the dashboard, the next step is to visualize the data in IBM Cognos
- Goes to the Corresponding Dashboard
- Select the visualizations tab in the left side of title bar
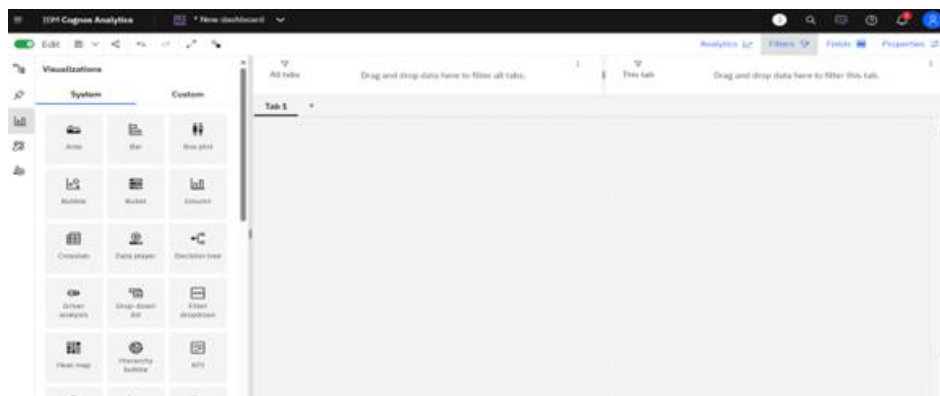
Choose the system as you want and put the data source for the required column

In the above screen shot displays the Line graph and model compares the "**people _vaccinated**" and "**people _fully _vaccinated**" from the time period of 2020 to 2022

    X-axis =**Dates**
    Y-axis = **people _vaccinated, people _fully _vaccinated**

After performing these activities a comprehensive document will be created to demonstrate the ability to Communicate and share finding.

DATA OBSERVATION :

- The country_vaccinations_by_manufacturer.csv and country_vaccinations.csv file contains metrics with rows representing different countries or regions and columns representing various attributes related to vaccination progress and population.
- It contains Covid Vaccines Analysis Parameters such as Location-country,
  Date,Vaccine- vaccine type,Total number of vaccinations, iso_code, total_vaccinations, people_vaccinated, people_fully_vaccinated, daily_vaccinations_raw, daily_vaccinations, and more.

IMPORTANCE OF LOADING AND PREPROCESSING DATASET :

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for advanced machine learning models, as vaccines datasets are often complex. By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

CHALLENGES INVOLVED IN LOADING AND PREPROCESSING A COVID VACCINES DATASET :

There are number of challenges involved in loading and preprocessing a Covid Vaccines dataset that we use for Covid Vaccines Analysis includes :

Handling Missing Values :

Covid Vaccines dataset often contain missing values, which can be due to a variety of factors, such as human error or incomplete data collection. Common methods for handling missing values include dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.

Scaling The Features :

It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

Splitting the Dataset into Training and Testing Sets:
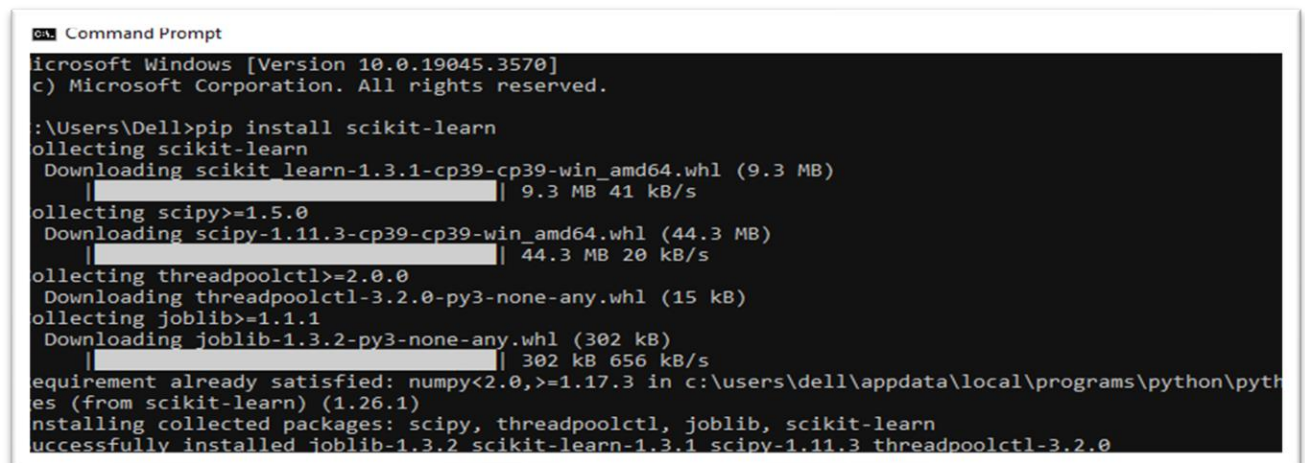
Once the data has been pre-processed, we need to split the dataset into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data. It is important to split the dataset in a way that is representative of the real world distribution of the data.

HOW TO OVERCOME THE CHALLENGES OF LOADING AND PREPROCESSING COVID VACCINES DATASET :

To overcome the challenges of loading and preprocessing a Covid Vaccines dataset, can include the following factors and features:

Use a Data Preprocessing Library:

There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

Carefully Consider the Specific needs of your Model:

The best way to preprocess the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that is compatible with the algorithm.

Validate the Preprocessed Data:

It is important to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods.

<u>LOADING THE DATASET:</u>

- Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.
- The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

<u>Identify the Dataset:</u>

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

- ♦ country_vaccinations.csv
- ♦ country_vaccinations_by_manufacturer.csv

<u>Load the Dataset:</u>

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

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

<u>Preprocess the Dataset:</u>

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Let's see, How the Covid Vaccines Dataset is Loaded and Accessed with the help of using the Python Jupyter Notebook.

IMPORT THE REQUIRED LIBRARIES:

To perform the data preprocessing, splitting, scaling, and other tasks as described, several libraries in Python are needed to be imported. Here are the required libraries for the code:

1. **For loading and preprocessing the dataset:**
   ```python
   import pandas as pd
   import numpy as np
   import matplotlib.pyplot as plt
   import seaborn as sns
   ```

2. **For handling missing data:**
   ```python
   from sklearn.impute import SimpleImputer
   ```

3. **For splitting the dataset into training and test sets:**
   ```python
   from sklearn.model_selection import train_test_split
   ```

4. **For feature scaling:**
   ```python
   from sklearn.preprocessing import StandardScaler
   ```

5. **To Install Python Libraries :**
   To Install the necessary libraries, run the given command in the command prompt.

Library		command
Pandas	-	pip install pandas
Numpy	-	pip install numpy
Matplotlib.pyplot	-	pip install matplotlib
Seaborn	-	pip install seaborn
Sklearn.model_selection	-	pip install scikit-learn
Sklearn.preprocessing	-	pip install scikit-learn

IMPORT AND LOAD THE DATASET :

Use Pandas to read the dataset file you downloaded and into a DataFrame:

Code:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('country_vaccinations.csv')
df
```

Output:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hund
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	5
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	5
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	5
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	5
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	5

86512 rows × 15 columns

PREPROCESSING THE DATASET :

- Data preprocessing is the process of
    1. Cleaning
    2. Transforming
    3. Integrating Data
  in order to make it ready for analysis.
- This may involve removing errors and inconsistencies, handling missing values, transforming the data into a consistent format, and scaling the data to a suitable range.

Some common data preprocessing tasks include:

Data Cleaning: This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missing values. Remove irrelevant data and handle missing values. Transform data into a consistent format.

Data Transformation: This involves converting the data into a format that is suitable for the analysis task. For example, this may involve converting categorical data to numerical data, or scaling the data to a suitable range.

Feature Engineering: This involves creating new features from the existing data. And derive meaningful insights, and enhance the predictive power of the data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.

Data Integration: This involves combining data from multiple sources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

Dimensionality Reduction: Reduce the number of variables to simplify analysis and improve model performance.

Normalization and Standardization: Scale data to a common range and format to ensure fairness in analysis.

Data preprocessing is an essential step in many data science projects. By carefully preprocessing the data, data scientists can improve the accuracy and reliability.

HANDLING THE MISSING DATA :

```
[3]:  #DATA PREPROCESSING:

      #1.Handling Missing Values:

      # Check for missing values
      df.isnull().sum()
```

```
[3]:  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
```

```
[4]:  # Fill missing values with appropriate values (e.g., mean, median, or a
      ↪specific value)
      df.fillna({'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}, inplace=True)

      df.isnull().sum()
```

```
[4]:  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
```

```
[5]: #2.Data Type Conversion:

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

```
df
```

```
[5]:          country iso_code       date  total_vaccinations  people_vaccinated  \
     0      Afghanistan      AFG 2021-02-22                 0.0                0.0
     1      Afghanistan      AFG 2021-02-23                 0.0                0.0
     2      Afghanistan      AFG 2021-02-24                 0.0                0.0
     3      Afghanistan      AFG 2021-02-25                 0.0                0.0
     4      Afghanistan      AFG 2021-02-26                 0.0                0.0
     ...            ...      ...        ...                 ...                ...
     86507     Zimbabwe      ZWE 2022-03-25           8691642.0          4814582.0
     86508     Zimbabwe      ZWE 2022-03-26           8791728.0          4886242.0
     86509     Zimbabwe      ZWE 2022-03-27           8845039.0          4918147.0
     86510     Zimbabwe      ZWE 2022-03-28           8934360.0          4975433.0
     86511     Zimbabwe      ZWE 2022-03-29           9039729.0          5053114.0

            people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
     0                          0.0                     0.0                 0.0
     1                          0.0                     0.0              1367.0
     2                          0.0                     0.0              1367.0
     3                          0.0                     0.0              1367.0
     4                          0.0                     0.0              1367.0
     ...                        ...                     ...                 ...
     86507                3473523.0                139213.0             69579.0
     86508                3487962.0                100086.0             83429.0
     86509                3493763.0                 53311.0             90629.0
     86510                3501493.0                 89321.0            100614.0
     86511                3510256.0                105369.0            103751.0

            total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
     0                                 0.00                           0.00
     1                                 0.00                           0.00
     2                                 0.00                           0.00
     3                                 0.00                           0.00
     4                                 0.00                           0.00
     ...                                ...                            ...
     86507                            57.59                          31.90
     86508                            58.25                          32.38
     86509                            58.61                          32.59
     86510                            59.20                          32.97
     86511                            59.90                          33.48

            people_fully_vaccinated_per_hundred  daily_vaccinations_per_million  \
     0                                      0.00                             0.0
     1                                      0.00                            34.0
     2                                      0.00                            34.0
     3                                      0.00                            34.0
     4                                      0.00                            34.0
     ...                                     ...                             ...
     86507                                 23.02                          4610.0
     86508                                 23.11                          5528.0
     86509                                 23.15                          6005.0
     86510                                 23.20                          6667.0
     86511                                 23.26                          6874.0

                                                     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...
     ...                                                   ...
     86507   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86508   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86509   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86510   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86511   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
     ...                           ...
     86507           Ministry of Health
     86508           Ministry of Health
     86509           Ministry of Health
     86510           Ministry of Health
     86511           Ministry of Health

                                               source_website
     0                               https://covid19.who.int/
     1                               https://covid19.who.int/
     2                               https://covid19.who.int/
     3                               https://covid19.who.int/
     4                               https://covid19.who.int/
     ...                                                  ...
     86507   https://www.arcgis.com/home/webmap/viewer.html...
     86508   https://www.arcgis.com/home/webmap/viewer.html...
     86509   https://www.arcgis.com/home/webmap/viewer.html...
     86510   https://www.arcgis.com/home/webmap/viewer.html...
     86511   https://www.arcgis.com/home/webmap/viewer.html...

     [86512 rows x 15 columns]
```

Handling Duplication :

```
[6]: #3.Handling Duplicates:

     df.drop_duplicates(inplace=True)

     df
```

```
[6]:            country iso_code       date  total_vaccinations  people_vaccinated  \
     0       Afghanistan      AFG 2021-02-22                 0.0                0.0
     1       Afghanistan      AFG 2021-02-23                 0.0                0.0
     2       Afghanistan      AFG 2021-02-24                 0.0                0.0
     3       Afghanistan      AFG 2021-02-25                 0.0                0.0
     4       Afghanistan      AFG 2021-02-26                 0.0                0.0
     ...             ...      ...        ...                 ...                ...
     86507      Zimbabwe      ZWE 2022-03-25           8691642.0          4814582.0
     86508      Zimbabwe      ZWE 2022-03-26           8791728.0          4886242.0
     86509      Zimbabwe      ZWE 2022-03-27           8845039.0          4918147.0
     86510      Zimbabwe      ZWE 2022-03-28           8934360.0          4975433.0
     86511      Zimbabwe      ZWE 2022-03-29           9039729.0          5053114.0

            people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
     0                          0.0                     0.0                 0.0
     1                          0.0                     0.0              1367.0
     2                          0.0                     0.0              1367.0
     3                          0.0                     0.0              1367.0
     4                          0.0                     0.0              1367.0
     ...                        ...                     ...                 ...
     86507                3473523.0                139213.0             69579.0
     86508                3487962.0                100086.0             83429.0
     86509                3493763.0                 53311.0             90629.0
     86510                3501493.0                 89321.0            100614.0
     86511                3510256.0                105369.0            103751.0

            total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
     0                                0.00                           0.00
     1                                0.00                           0.00
     2                                0.00                           0.00
     3                                0.00                           0.00
     4                                0.00                           0.00
     ...                               ...                            ...
     86507                           57.59                          31.90
     86508                           58.25                          32.38
     86509                           58.61                          32.59
     86510                           59.20                          32.97
     86511                           59.90                          33.48

            people_fully_vaccinated_per_hundred  daily_vaccinations_per_million  \
     0                                     0.00                             0.0
     1                                     0.00                            34.0
     2                                     0.00                            34.0
     3                                     0.00                            34.0
     4                                     0.00                            34.0
     ...                                    ...                             ...
     86507                                23.02                          4610.0
     86508                                23.11                          5528.0
     86509                                23.15                          6005.0
     86510                                23.20                          6667.0
     86511                                23.26                          6874.0

                                                     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...
     ...                                                    ...
     86507      Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86508      Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86509      Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86510      Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
     86511      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
     ...                          ...
     86507              Ministry of Health
     86508              Ministry of Health
     86509              Ministry of Health
     86510              Ministry of Health
     86511              Ministry of Health

                                    source_website
     0                        https://covid19.who.int/
     1                        https://covid19.who.int/
     2                        https://covid19.who.int/
     3                        https://covid19.who.int/
     4                        https://covid19.who.int/
     ...                                          ...
     86507      https://www.arcgis.com/home/webmap/viewer.html...
     86508      https://www.arcgis.com/home/webmap/viewer.html...
     86509      https://www.arcgis.com/home/webmap/viewer.html...

     86510      https://www.arcgis.com/home/webmap/viewer.html...
     86511      https://www.arcgis.com/home/webmap/viewer.html...

     [86512 rows x 15 columns]
```

Dropping Null Values :

```
In [7]: #drop the null values in the datsets using drop()
        df1=df.dropna()
        print(df1)
```

```
              country iso_code        date  total_vaccinations  \
94        Afghanistan      AFG  2021-05-27            593313.0
101       Afghanistan      AFG  2021-06-03            630305.0
339       Afghanistan      AFG  2022-01-27           5081064.0
433           Albania      ALB  2021-02-18              3049.0
515           Albania      ALB  2021-05-11            622507.0
...               ...      ...         ...                 ...
86507        Zimbabwe      ZWE  2022-03-25           8691642.0
86508        Zimbabwe      ZWE  2022-03-26           8791728.0
86509        Zimbabwe      ZWE  2022-03-27           8845039.0
86510        Zimbabwe      ZWE  2022-03-28           8934360.0
86511        Zimbabwe      ZWE  2022-03-29           9039729.0

        people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  \
94              479574.0                 113739.0                  2859.0
101             481800.0                 148505.0                  4015.0
339            4517380.0                3868832.0                  6868.0
433               2438.0                    611.0                  1348.0
515             440921.0                 181586.0                  9548.0
...                  ...                      ...                     ...
86507          4814582.0                3473523.0                139213.0
86508          4886242.0                3487962.0                100086.0
86509          4918147.0                3493763.0                 53311.0
86510          4975433.0                3501493.0                 89321.0
86511          5053114.0                3510256.0                105369.0

        daily_vaccinations  total_vaccinations_per_hundred  \
94                 6487.0                            1.49
101                5285.0                            1.58
339                9802.0                           12.76
433                 254.0                            0.11
515               12160.0                           21.67
...                   ...                             ...
86507             69579.0                           57.59
86508             83429.0                           58.25
86509             90629.0                           58.61
86510            100614.0                           59.20
86511            103751.0                           59.90

        people_vaccinated_per_hundred  people_fully_vaccinated_per_hundred  \
94                               1.20                                 0.29
101                              1.21                                 0.37
339                             11.34                                 9.71
433                              0.08                                 0.02
515                             15.35                                 6.32
...                               ...                                  ...
...                               ...                                  ...
86507                           31.90                                23.02
86508                           32.38                                23.11
86509                           32.59                                23.15
86510                           32.97                                23.20
86511                           33.48                                23.26

        daily_vaccinations_per_million  \
94                              163.0
101                             133.0
339                             246.0
433                              88.0
515                            4233.0
...                               ...
86507                          4610.0
```

```
86508                    5528.0
86509                    6005.0
86510                    6667.0
86511                    6874.0

                                            vaccines  \
94      Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
101     Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
339     Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
433     Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...
515     Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...
...                                               ...
86507   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
86508   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
86509   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
86510   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
86511   Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

                        source_name  \
94        World Health Organization
101       World Health Organization
339       World Health Organization
433                 Ministry of Health
515                 Ministry of Health
...                             ...
86507               Ministry of Health
86508               Ministry of Health
86509               Ministry of Health
86510               Ministry of Health
86511               Ministry of Health

                                        source_website
94                            https://covid19.who.int/
101                           https://covid19.who.int/
339                           https://covid19.who.int/
433     https://shendetesia.gov.al/vaksinimi-anticovid...
515     https://shendetesia.gov.al/vaksinimi-anticovid...
...                                               ...
86507   https://www.arcgis.com/home/webmap/viewer.html...
86508   https://www.arcgis.com/home/webmap/viewer.html...
86509   https://www.arcgis.com/home/webmap/viewer.html...
86510   https://www.arcgis.com/home/webmap/viewer.html...
86511   https://www.arcgis.com/home/webmap/viewer.html...

[30847 rows x 15 columns]
```

In [8]:
```python
#view the information of the dataset
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30847 entries, 94 to 86511
Data columns (total 15 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   country                             30847 non-null  object
 1   iso_code                            30847 non-null  object
 2   date                                30847 non-null  object
 3   total_vaccinations                  30847 non-null  float64
 4   people_vaccinated                   30847 non-null  float64
 5   people_fully_vaccinated             30847 non-null  float64
 6   daily_vaccinations_raw              30847 non-null  float64
 7   daily_vaccinations                  30847 non-null  float64
 8   total_vaccinations_per_hundred      30847 non-null  float64
 9   people_vaccinated_per_hundred       30847 non-null  float64
 10  people_fully_vaccinated_per_hundred 30847 non-null  float64
 11  daily_vaccinations_per_million      30847 non-null  float64
 12  vaccines                            30847 non-null  object
 13  source_name                         30847 non-null  object
 14  source_website                      30847 non-null  object
dtypes: float64(9), object(6)
memory usage: 3.8+ MB
```

```
In [9]: #view the statistical analysis the dataset
        df1.describe()
```

Out[9]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vacc
count	3.084700e+04	3.084700e+04	3.084700e+04	3.084700e+04	3.084
mean	3.980375e+07	2.177533e+07	1.579596e+07	2.021875e+05	1.975
std	1.451667e+08	8.053173e+07	5.898165e+07	7.041931e+05	6.400
min	3.000000e+00	3.000000e+00	1.000000e+00	0.000000e+00	0.000
25%	1.153332e+06	7.339795e+05	3.704450e+05	5.498000e+03	7.329
50%	6.335305e+06	3.688092e+06	2.211035e+06	2.908100e+04	3.247
75%	2.520629e+07	1.440668e+07	9.121526e+06	1.344580e+05	1.402
max	3.243599e+09	1.275541e+09	1.240777e+09	1.862727e+07	1.307

```
In [10]: #view the columns count
         df.isnull().sum()
```

```
Out[10]: 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
```

```
In [11]: #view the columns in the dataset
         df.columns
```

```
Out[11]: Index(['country', 'iso_code', 'date', 'total_vaccinations',
               'people_vaccinated', 'people_fully_vaccinated',
               'daily_vaccinations_raw', 'daily_vaccinations',
               'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
               'people_fully_vaccinated_per_hundred', 'daily_vaccinations_per_million',
               'vaccines', 'source_name', 'source_website'],
              dtype='object')
```

```
In [12]:  #convert the float column into integer column

          df1['people_vaccinated'] = df1['people_vaccinated'].astype(int)

          df1['people_fully_vaccinated'] = df1['people_fully_vaccinated'].astype(int)

          df1['daily_vaccinations_raw'] = df1['daily_vaccinations_raw'].astype(int)

          df1['total_vaccinations_per_hundred'] = df1['total_vaccinations_per_hundred'].astype(i

          df1['people_vaccinated_per_hundred'] = df1['people_vaccinated_per_hundred'].astype(int

          df1['people_fully_vaccinated_per_hundred'] = df1['people_fully_vaccinated_per_hundred'

          df1['daily_vaccinations_per_million'] = df1['daily_vaccinations_per_million'].astype(i

          df1.head()
```

Out[12]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
94	Afghanistan	AFG	2021-05-27	593313.0	479574	113739	
101	Afghanistan	AFG	2021-06-03	630305.0	481800	148505	
339	Afghanistan	AFG	2022-01-27	5081064.0	4517380	3868832	
433	Albania	ALB	2021-02-18	3049.0	2438	611	
515	Albania	ALB	2021-05-11	622507.0	440921	181586	

```
In [13]:  #again check the information of dataset
          df1.info()

          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 30847 entries, 94 to 86511
          Data columns (total 15 columns):
           #   Column                               Non-Null Count  Dtype
          ---  ------                               --------------  -----
           0   country                              30847 non-null  object
           1   iso_code                             30847 non-null  object
           2   date                                 30847 non-null  object
           3   total_vaccinations                   30847 non-null  float64
           4   people_vaccinated                    30847 non-null  int32
           5   people_fully_vaccinated              30847 non-null  int32
           6   daily_vaccinations_raw               30847 non-null  int32
           7   daily_vaccinations                   30847 non-null  float64
           8   total_vaccinations_per_hundred       30847 non-null  int32
           9   people_vaccinated_per_hundred        30847 non-null  int32
           10  people_fully_vaccinated_per_hundred  30847 non-null  int32
           11  daily_vaccinations_per_million       30847 non-null  int32
           12  vaccines                             30847 non-null  object
           13  source_name                          30847 non-null  object
           14  source_website                       30847 non-null  object
          dtypes: float64(2), int32(7), object(6)
          memory usage: 2.9+ MB
```

Dropping the Unwanted columns in the Dataset :

```
In [14]: #drop the unwanted column in dataset

         df1=df1.drop(['vaccines','source_name','source_website'],axis=1)

         df1
```

Out[14]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	da
94	Afghanistan	AFG	2021-05-27	593313.0	479574	113739	
101	Afghanistan	AFG	2021-06-03	630305.0	481800	148505	
339	Afghanistan	AFG	2022-01-27	5081064.0	4517380	3868832	
433	Albania	ALB	2021-02-18	3049.0	2438	611	
515	Albania	ALB	2021-05-11	622507.0	440921	181586	
...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582	3473523	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242	3487962	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147	3493763	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433	3501493	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114	3510256	

30847 rows × 12 columns

```
In [39]: #The date is in the 'object' format. Let us change it to Datetime format for easy hand
         df1['date'] =pd.to_datetime(df['date'], format='%Y-%m-%d')

In [15]: df1
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	da
94	Afghanistan	AFG	2021-05-27	593313.0	479574	113739	
101	Afghanistan	AFG	2021-06-03	630305.0	481800	148505	
339	Afghanistan	AFG	2022-01-27	5081064.0	4517380	3868832	
433	Albania	ALB	2021-02-18	3049.0	2438	611	
515	Albania	ALB	2021-05-11	622507.0	440921	181586	
...	...	...	...	...	...	...	...
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582	3473523	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242	3487962	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147	3493763	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433	3501493	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114	3510256	

30847 rows × 12 columns

In [46]:
```python
#Group by total vaccinations given by country and sort descending to identify the top
vacc_by_country = df.groupby('country').max().sort_values('total_vaccinations', ascend
vacc_by_country = vacc_by_country.iloc[:10]
vacc_by_country
```

country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vacc
Afghanistan	AFG	2022-03-22	nan	5082824.0	4420127.0	
Russia	RUS	2022-03-29	nan	79954746.0	72841232.0	
Nauru	NRU	2022-03-21	nan	9150.0	7674.0	
Nepal	NPL	2022-03-29	nan	21994736.0	19014212.0	
Netherlands	NLD	2022-03-19	nan	13455761.0	12366525.0	
New Caledonia	NCL	2022-03-28	nan	188003.0	179880.0	
New Zealand	NZL	2022-03-29	nan	4284293.0	4051832.0	
Nicaragua	NIC	2022-03-25	nan	5498389.0	4113547.0	
Niger	NER	2022-03-24	nan	2180972.0	1545630.0	
Nigeria	NGA	2022-03-27	nan	21049754.0	9565143.0	

ENCODING CATEGORICAL DATA :

To encode categorical data using one-hot encoding in Python, you can use the pd.get_dummies function in the Pandas library. One-hot encoding converts categorical variables into binary (0/1) format, making them suitable for machine learning algorithms.

Code:

```python
[14]: # Use get_dummies to perform one-hot encoding
dataset_encoded = pd.get_dummies(dataset, columns=['country'])
# Display the DataFrame with one-hot encoding
print(dataset_encoded.head())
```

Output:

```
    iso_code        date  total_vaccinations  people_vaccinated  \
0        AFG  2021-02-22                 0.0                0.0
1        AFG  2021-02-23                 NaN                NaN
2        AFG  2021-02-24                 NaN                NaN
3        AFG  2021-02-25                 NaN                NaN
4        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  ...  country_Uruguay  \
0                                  NaN  ...            False
1                                  NaN  ...            False
2                                  NaN  ...            False
3                                  NaN  ...            False
4                                  NaN  ...            False

   country_Uzbekistan country_Vanuatu country_Venezuela  country_Vietnam  \
0               False           False            False            False
1               False           False            False            False
2               False           False            False            False
3               False           False            False            False
4               False           False            False            False

   country_Wales  country_Wallis and Futuna  country_Yemen  country_Zambia  \
0          False                      False          False           False
1          False                      False          False           False
2          False                      False          False           False
3          False                      False          False           False
4          False                      False          False           False

   country_Zimbabwe
0             False
1             False
2             False
3             False
4             False

[5 rows x 237 columns]
```

 The get_dummies function will create binary (0/1) columns for each unique category in the ' country' column. This process effectively converts the categorical data into a numerical form at suitable for analysis or machine learning.

Categorical Variable Encoding Techniques

Find and replace — Replaces each matching occurrence of an old character in a string with a new character.

Label encoding — Assigns each label with a unique integer based on alphabetical ordering.

One-hot encoding — Converts each categorical variable into a column and assigns it a 1 or 0 value.
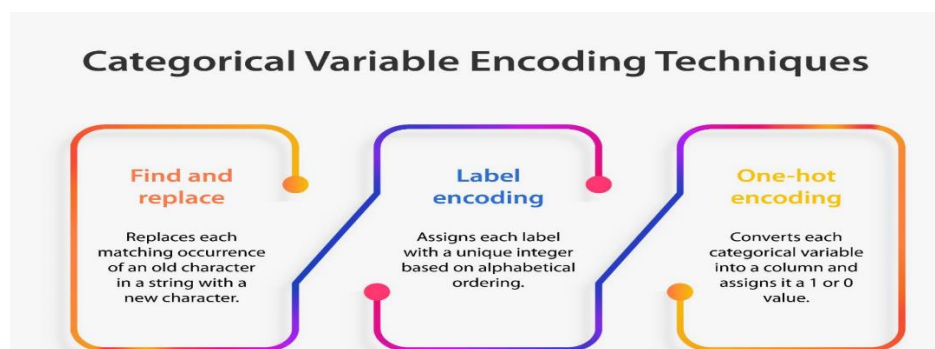
```
[8]: #5.Encoding Categorical Variables:

     df = pd.get_dummies(df, columns=['country', 'vaccines'], drop_first=True)


     df
```

```
[8]:         iso_code        date  total_vaccinations  people_vaccinated  \
     0            AFG  2021-02-22           -0.143704          -0.170046
     1            AFG  2021-02-23           -0.143704          -0.170046
     2            AFG  2021-02-24           -0.143704          -0.170046
     3            AFG  2021-02-25           -0.143704          -0.170046
     4            AFG  2021-02-26           -0.143704          -0.170046
     ...          ...         ...                 ...                ...
     86507        ZWE  2022-03-25           -0.089753          -0.073170
     86508        ZWE  2022-03-26           -0.089132          -0.071728
     86509        ZWE  2022-03-27           -0.088801          -0.071086
     86510        ZWE  2022-03-28           -0.088247          -0.069933
     86511        ZWE  2022-03-29           -0.087593          -0.068370

            people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
     0                          0.0                     0.0                 0.0
     1                          0.0                     0.0              1367.0
     2                          0.0                     0.0              1367.0
     3                          0.0                     0.0              1367.0
     4                          0.0                     0.0              1367.0
     ...                        ...                     ...                 ...
     86507                3473523.0                139213.0             69579.0
     86508                3487962.0                100086.0             83429.0
     86509                3493763.0                 53311.0             90629.0
     86510                3501493.0                 89321.0            100614.0
     86511                3510256.0                105369.0            103751.0

            total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
```

```
0                                      0.00                           0.00
1                                      0.00                           0.00
2                                      0.00                           0.00
3                                      0.00                           0.00
4                                      0.00                           0.00
...                                     ...                            ...
86507                                 57.59                          31.90
86508                                 58.25                          32.38
86509                                 58.61                          32.59
86510                                 59.20                          32.97
86511                                 59.90                          33.48

       people_fully_vaccinated_per_hundred  ... \
0                                     0.00  ...
1                                     0.00  ...
2                                     0.00  ...
3                                     0.00  ...
4                                     0.00  ...
...                                    ...  ...
86507                                23.02  ...
86508                                23.11  ...
86509                                23.15  ...
86510                                23.20  ...
86511                                23.26  ...

       vaccines_Oxford/AstraZeneca, Sputnik V  vaccines_Pfizer/BioNTech  \
0                                           0                         0
1                                           0                         0
2                                           0                         0
3                                           0                         0
4                                           0                         0
...                                       ...                       ...
86507                                       0                         0
86508                                       0                         0
86509                                       0                         0
86510                                       0                         0
86511                                       0                         0

       vaccines_Pfizer/BioNTech, Sinopharm/Beijing  \
0                                                0
1                                                0
2                                                0
3                                                0
4                                                0
...                                            ...
86507                                            0
86508                                            0

86509                                            0
86510                                            0
86511                                            0

       vaccines_Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V  \
0                                                           0
1                                                           0
2                                                           0
3                                                           0
4                                                           0
...                                                       ...
86507                                                       0
86508                                                       0
86509                                                       0
86510                                                       0
86511                                                       0
```

```
        vaccines_Pfizer/BioNTech, Sinovac  \
0                                        0
1                                        0
2                                        0
3                                        0
4                                        0
...                                    ...
86507                                    0
86508                                    0
86509                                    0
86510                                    0
86511                                    0

        vaccines_Pfizer/BioNTech, Sinovac, Turkovac  \
0                                                  0
1                                                  0
2                                                  0
3                                                  0
4                                                  0
...                                              ...
86507                                              0
86508                                              0
86509                                              0
86510                                              0
86511                                              0

        vaccines_Pfizer/BioNTech, Sputnik V  \
0                                          0
1                                          0
2                                          0
3                                          0
4                                          0
...                                      ...
86507                                      0
86508                                      0
86509                                      0
86510                                      0
86511                                      0

        vaccines_QazVac, Sinopharm/Beijing, Sputnik V  \
0                                                    0
1                                                    0
2                                                    0
3                                                    0
4                                                    0
...                                                ...
86507                                                0
86508                                                0
86509                                                0
86510                                                0
86511                                                0

        vaccines_Sinopharm/Beijing  vaccines_Sinopharm/Beijing, Sputnik V
0                                0                                      0
1                                0                                      0
2                                0                                      0
3                                0                                      0
4                                0                                      0
...                            ...                                    ...
86507                            0                                      0
86508                            0                                      0
86509                            0                                      0
86510                            0                                      0
86511                            0                                      0

[86512 rows x 318 columns]
```

SPLITTING THE DATASET INTO TEST SET AND TRAINING SET :

To split dataset into training and test sets using the train_test_split function from scikitlearn,input features (X) and target variable (Y) needed to be specified first.

Code:

Features Splitting :

```
x=cov19[['country', 'iso_code','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']]
```

Target Splitting :

```
y=cov19[['total_vaccinations']]
```

Train_Test_Splitting :

```
x_train, x_test, y_train, y_test = train_test_split(x,y, random_state=42)
```

In this code, we first separate the features (X) and the target variable (Y) from the dataset. Then, we use train_test_split to split the data into training and test sets. The test_size parameter determines the proportion of the data that will be allocated to the test set, and random_state is set to a specific value (e.g., 42) to ensure reproducibility.

Machine Learning Algorithms

Machine learning algorithms play a crucial role in data analysis by enabling automated data modeling, pattern recognition, and predictive analytics.

Machine learning algorithms enhance data analysis by automating complex tasks, uncovering hidden patterns, and providing data-driven insights.

Linear Regression :

Linear regression is a statistical method used for modeling the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship, where changes in the independent variables result in a proportional change in the dependent variable. The goal is to find the best-fitting line (linear equation) that minimizes the sum of squared differences between observed and predicted values, allowing for predictions and understanding the impact of independent variables on the dependent variable. Linear regression is widely used for tasks such as prediction, trend analysis, and understanding correlations.

```
LR=LinearRegression()
LR.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
coeffcients = pd.DataFrame([x_train.columns,LR.coef_]).T
coeffcients = coeffcients.rename(columns={0: 'Attribute',1: 'Coefficients'})
coeffcients
```

	Attribute	Coefficients
0	country	[-99529.21653173528, 38852.46701139158, -0.403...
1	iso_code	None
2	people_vaccinated	None
3	people_fully_vaccinated	None
4	daily_vaccinations_raw	None
5	daily_vaccinations	None
6	total_vaccinations_per_hundred	None
7	people_vaccinated_per_hundred	None
8	people_fully_vaccinated_per_hundred	None
9	daily_vaccinations_per_million	None
10	vaccines	None
11	source_name	None
12	source_website	None

```
# Model Evaluation
print('R^2:',metrics.r2_score(y_test, y_pred_LR))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred_LR))
print('MSE:',metrics.mean_squared_error(y_test, y_pred_LR))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred_LR)))
```

```
R^2: 0.6117337462998536
MAE: 26021756.076752324
MSE: 1.0588822188481566e+16
RMSE: 102902002.83999124
```

R^2 : It is a measure of the linear relationship between X and Y. It is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable.

Adjusted R^2 :The adjusted R-squared compares the explanatory power of regression models that contain different numbers of predictors.

MAE : It is the mean of the absolute value of the errors. It measures the difference between two continuous variables, here actual and predicted values of y.

MSE: The mean square error (MSE) is just like the MAE, but squares the difference before summing them all instead of using the absolute value.

RMSE: The mean square error (MSE) is just like the MAE, but squares the difference before summing them all instead of using the absolute value.

```
plt.scatter(y_test, y_pred_LR)
plt.xlabel("total vaccines")
plt.ylabel("Predicted total vaccines")
plt.title("TOTAL VACCINES vs Predicted TOTAL VACCINES with LR")
plt.show()
```


TOTAL VACCINES vs Predicted TOTAL VACCINES with LR
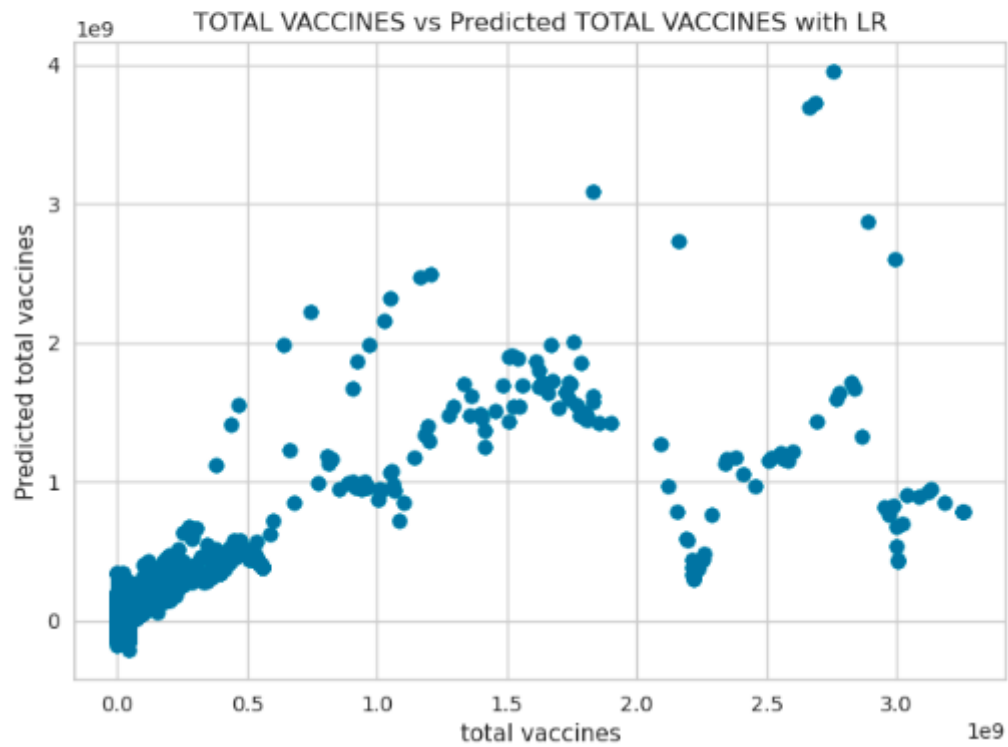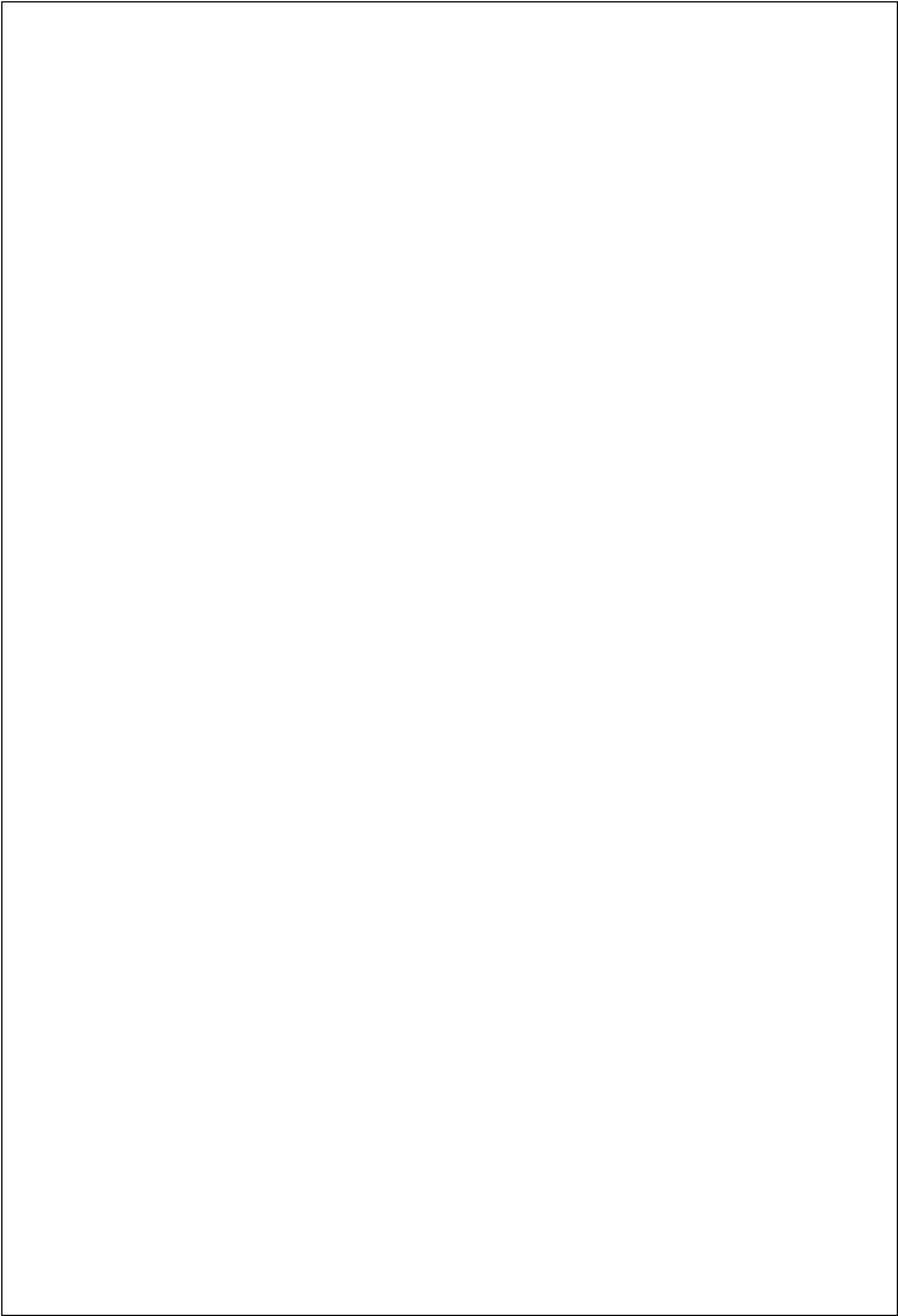
Importance of Visualizing the Dataset :

The importance of visualizing datasets cannot be overstated in the realm of data analysis. Visualization serves as a powerful bridge between raw data and human comprehension. It enhances our ability to understand, interpret, and extract meaningful insights from complex datasets.

By transforming numbers and statistics into charts, graphs, and interactive displays, visualization offers several key advantages. Firstly, it enables us to detect patterns, trends, and outliers that might remain hidden in tabular data, facilitating more accurate and timely decision-making. Moreover, it supports data exploration by allowing users to interact with the data, making it easier to uncover specific details and refine analysis.

This feature is particularly valuable in the age of big data, where sifting through vast datasets can be a formidable challenge. It is a time-saving tool that provides a rapid overview of data, streamlining the analysis process.
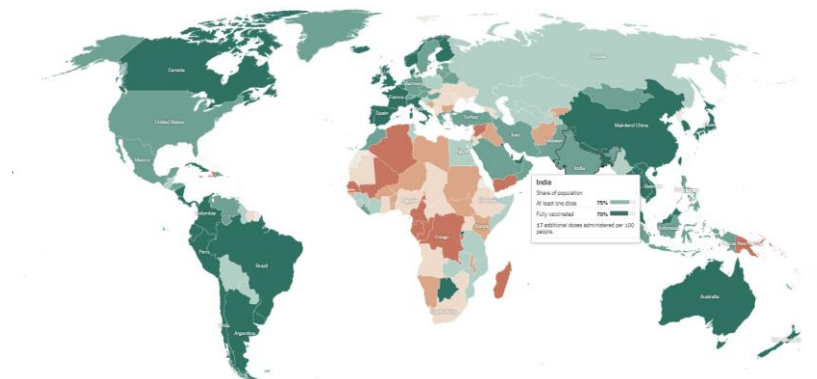
Visualizing the dataset :

- Visualizing a dataset in Python is a fundamental aspect of data analysis and interpretation. It involves creating graphical representations of data to uncover patterns, relationships, and insights, making it an essential tool in data-driven decision-making and storytelling.

- Visualizing a dataset in Python is the process of using data visualization libraries like Matplotlib, Seaborn, or Plotly to create graphical representations of data. The goal is to gain insights, identify patterns, and present data in a visual format that is easy to understand.

Identify The Dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

Load The Dataset:

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing our own code.

Preprocess The Dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start visualizing the dataset. Because, the raw data may contain numerous Null values and anomaly values. So it could be preprocessd also for training and evaluating the model. This may involve cleaning the data, transforming the data into a suitable format, handling the missing values.

Let's see, How the covid vaccines dataset is loaded and visualized using Python Jupyter Notebook and IBM cognos.

Exploratory Data Analysis

Correlation analysis : explore the relationships between different variables to uncover patterns and potential factors.

Data visualization : create informative charts, graphs and plots to visually present the covid vaccines data.

Hypothesis testing : formulate hypotheses and perform statistical test to validate or refute assumptions.

BASIC INFO ABOUT DATASET :

```
print('Data point starts from:',df.date.min(),'\n')
print('Data point ends at:',df.date.max(),'\n')
print('Total no of Countries in the data set:',len(df.country.unique()),'\n')
print('Total no of unique Vaccine Schemes in the data set:',len(df.vaccines.unique()),'\n')
```

```
Data point starts from: 2020-12-02

Data point ends at: 2022-03-29

Total no of Countries in the data set: 219

Total no of unique Vaccine Schemes in the data set: 84
```

DATAFRAME DESCRIBE :

```
In [7]: df.describe()
```

Out[7]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccina
count	8.651200e+04	8.651200e+04	8.651200e+04	8.651200e+04	8.651200e+04	86512.000000	
mean	2.315117e+07	8.451007e+06	6.341251e+06	1.106083e+05	1.308517e+05	40.419616	
std	1.611037e+08	4.969867e+07	3.890729e+07	7.864756e+05	7.669487e+05	62.707869	
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	
25%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	8.770000e+02	0.000000	
50%	1.008000e+03	0.000000e+00	0.000000e+00	0.000000e+00	7.245000e+03	0.010000	
75%	3.697554e+06	1.843103e+06	1.137869e+06	1.280625e+04	4.370450e+04	68.750000	
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	

VACCINIZATIONS WORLD-WIDE COMPARISION :

```
fig = px.bar(country_data[:40], x = 'Country', y = 'Total Vaccinations', color = 'Total Vaccinations')

fig.update_layout(title = dict(text = 'Vaccinizations World-Wide Comparision', x=0.5, y=0.95))
fig.update_xaxes(title = 'Countries', title_font = dict(size=18, family='Courier', color='crimson'), tickangle=-90)
fig.update_yaxes(title = 'Total Vaccinations', title_font = dict(size=18, family='Courier', color='crimson'))

fig.show()
```
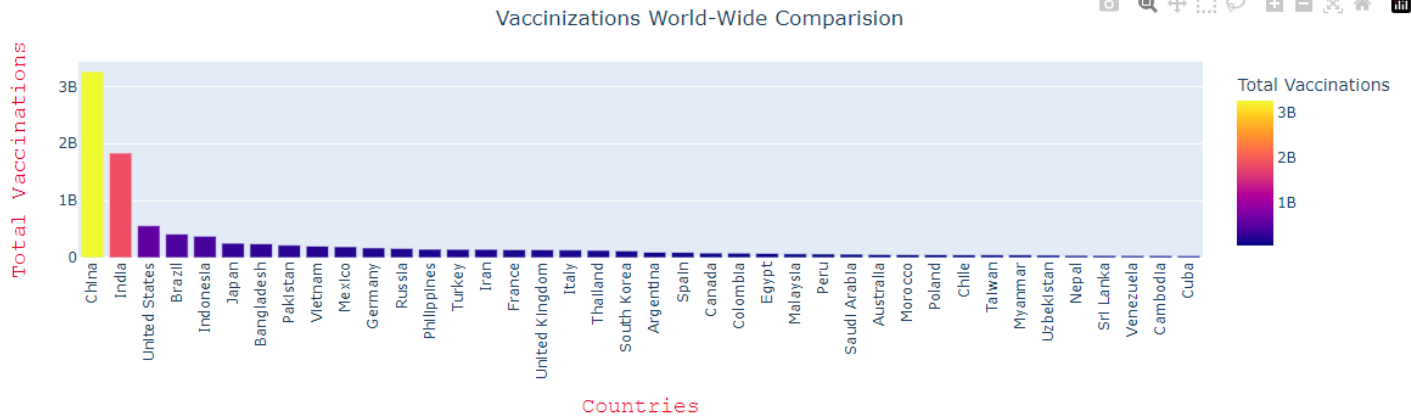

From the plot, some interesting facts stand out:

- The **United States**, despite having the highest number of people affected by Covid-19, has the highest number of vaccinated people.
- **China**, from where the virus started spreading, is at second.
- **India**, who has been supplying vaccines to the world is at 3th position.
- **UK**, where we have found a new variant strain of the virus, is right next.
- Following that, we have **Israel**, **UAE**, **Brazil**, **Germany** and others

COUNTRY WISE DAILY VACCINATION :

```
top_countries = ['USA','CHN','GBR','IND','ISR','ARE','BRA','DEU','TUR','ITA','FRA']
fig = px.line(df[df.iso_code.isin(top_countries)], x='date', y='daily_vaccinations', color='country')

fig.update_layout(title = dict(text = 'World-Wide Daily Vaccination Timeline', x=0.5, y=0.95),
                legend = dict(title = 'Country', traceorder = 'reversed'))
fig.update_xaxes(title = 'Timeline', title_font = dict(size=18, family='Courier', color='crimson'))
fig.update_yaxes(title = 'Daily Vaccinations', title_font = dict(size=18, family='Courier', color='crimson'))

fig.show()
#Country wise daily vaccination
```
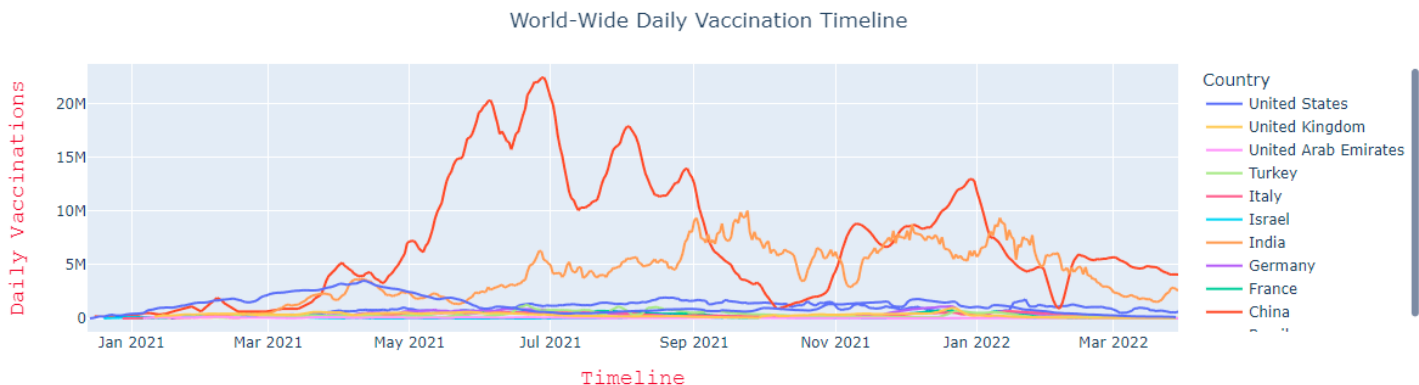
From the plot, we can deduce:

- The Line plot for China is composed entirely of straight lines. This can be attributed to the CCP which tries to restrict flow of information in and out of China. Thus, information from China usually comes in intervals and can be taken with a grain of salt.
- Comparatively, the plot of vaccinations in the USA is better plotted. We can also see that while the USA was heavily affected by the virus, its vaccination drive is highly effective.
- Others like the UK have a steady increase in Daily Vaccinations and India, while supplying to many countries, maintains a respectable 3th position.

TREEMAP OF TOTAL VACCINATIONS PER COUNTRY, GROUPED BY VACCINE SCHEME :

```
fig = px.treemap(country_data, path = ['Vaccines', 'Country'], values = 'Total Vaccinations', height = 650,
            custom_data = ['Country', 'Vaccines', 'Total Vaccinations'])

fig.update_layout(title = dict(text = 'Total vaccinations per country, grouped by Vaccine Scheme', x=0.5, y=0.95))
fig.update_traces(hovertemplate = 'Country: %{customdata[0]}<br>Vaccine: %{customdata[1]}<br>Total Vaccinations: %{customdata[2]}')
fig.show()
```


Total vaccinations per country, grouped by Vaccine Scheme

- From the above Treemap we can realise that a Bar and Pie Plot may often only show a part of the information that can be observed, whereas a Treemap can accurately show the share of a particular vaccine world-wide, the countries that are using the said vaccine and can even show comparisons between all the countries.
- As the Treemap shows so much information at a time, it can help one understand the data much more accurately.

TOTAL VACCINATIONS IN EVERY COUNTRY :

```python
fig = px.choropleth(country_data, locations = 'Country', color = 'Total Vaccinations',
                    locationmode = 'country names', color_continuous_scale = 'rainbow',
                    hover_name = 'Country', projection = 'natural earth')

fig.update_layout(title = dict(text = 'Total Vaccinations in every Country', x=0.5, y=0.95),
                  geo = dict(showocean = True, oceancolor = "#7af8ff", showland = True,
                             landcolor = "white",showlakes = False, showframe = False))

fig.show()
```

Total Vaccinations in every Country
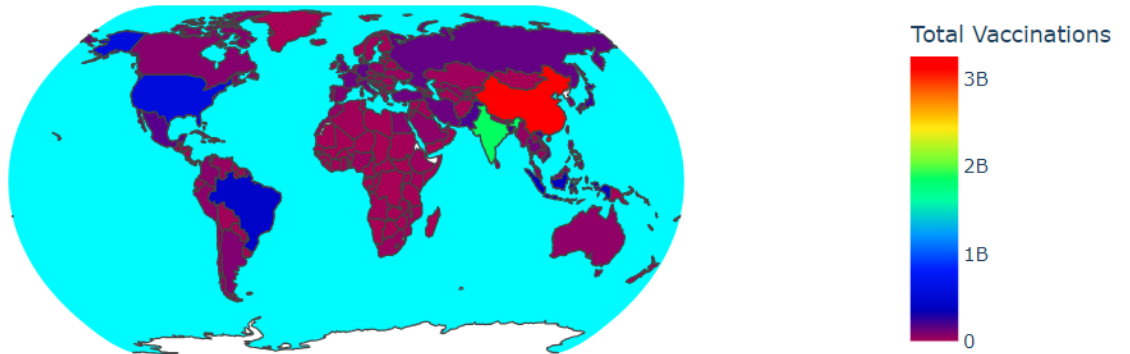
- In the above visualisation, we can see the countries and the total vaccinations they have completed.

Statistical Analysis

Descriptive statistics : summarize the covid vaccines data using measures like mean, median and standard deviation .

Inferential statistics : draw conclusions about the populations based on sample data and calculate confidence intervals.

Regression analysis : examine the relationship between variables and predict future trends using regression models.

Which countries started vaccinations first?

```
In [5]:   # Find out which countries started vaccinations earliest
          vacc['date'] = pd.to_datetime(vacc['date'], utc=True)
          vacc_start = vacc.loc[vacc[vacc.total_vaccinations > 0].groupby('country')['date'].idxmin()].sort_va
          lues('date')
          vacc_start.head(5)
```

Out[5]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
3487	Wales	NaN	2020-12-13 00:00:00+00:00	8212.0	8212.0	NaN	NaN
3357	United Kingdom	GBR	2020-12-13 00:00:00+00:00	86265.0	86265.0	NaN	NaN
999	England	NaN	2020-12-13 00:00:00+00:00	55437.0	55437.0	NaN	NaN
2807	Scotland	NaN	2020-12-13 00:00:00+00:00	18993.0	18993.0	NaN	NaN
2271	Northern Ireland	NaN	2020-12-13 00:00:00+00:00	3623.0	3623.0	NaN	NaN

How have the cumulative number of countries adopting covid-19 vaccinations evolved over time?

In [7]:

```python
# Cumulative distribution of vaccination start dates
events = pd.Series(vacc_start.date.value_counts())
events.index = pd.to_datetime(events.index)
events.sort_index(inplace=True)

plt.plot(events.cumsum())
plt.xticks(rotation=90)
plt.title('Cumulative Frequency of Countries that have Started Vaccinations')
plt.xlabel('Date')
plt.ylabel('Number of Countries')
plt.show()
```


Cumulative Frequency of Countries that have Started Vaccinations
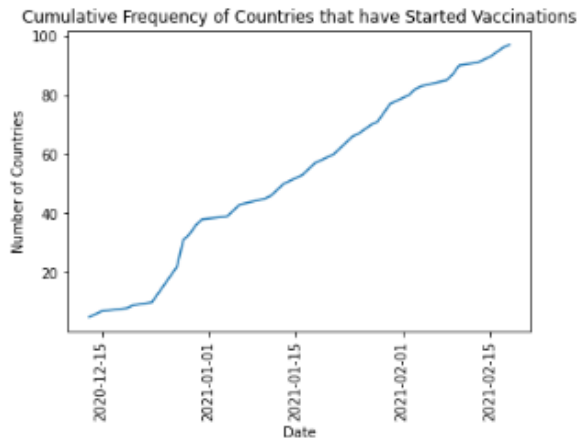
What are the top 20 countries in terms of total number of vaccines administered?

In [11]:

```python
# Plot out which countries have performed most vaccinations in descending order
plt.figure(figsize=(10, 4))
plt.bar(vacc_total.country[0:20], vacc_total.total_vaccinations[0:20])
plt.xticks(rotation=90)
plt.title('Top 20 Countries by Total Vaccinations')
plt.xlabel('Country')
plt.ylabel('Total Vaccinations (x 10Million)')
plt.show()
```


Top 20 Countries by Total Vaccinations
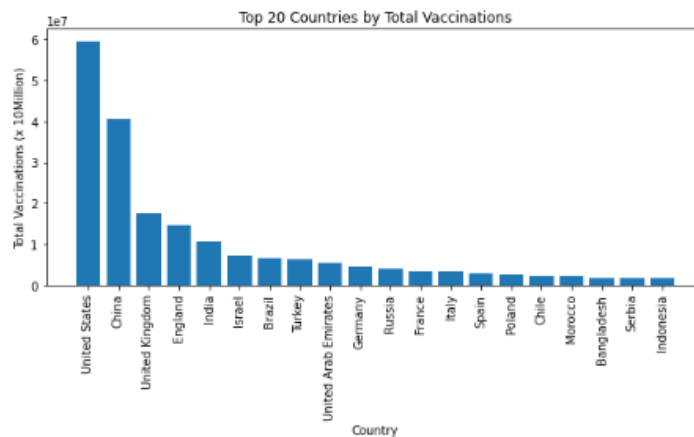
What are the top 20 countries in terms of number of vaccines administered per 100 population?

```
In [14]:
# Find out which countries have performed most vaccinations with respect to their populations

plt.figure(figsize=(10, 4))
plt.bar(vacc_total.country[0:20], vacc_total.total_vaccinations_per_hundred[0:20])
plt.xticks(rotation=90)
plt.title('Top 20 Countries by Total Vaccinations per 100 Population')
plt.xlabel('Country')
plt.ylabel('Total Vaccinations per 100 Population')
plt.show()
```
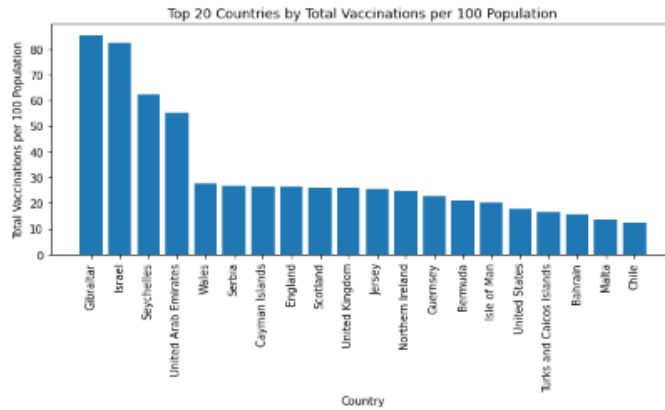

Which countries are using more than 1 type of vaccine, and how many are they using?

```
# How many vaccines is each country using

vacc_total['vacc_brands'] = vacc_total.iloc[:,-5:].apply(lambda x: (5 - x.isnull().sum()), axis='columns')
vacc_total = vacc_total.sort_values('vacc_brands',ascending=False)
plt.figure(figsize=(20, 4))
plt.bar(vacc_total.country, vacc_total.vacc_brands)
plt.xticks(rotation=90)
plt.title('Total Number of Vaccine Brands Used')
plt.xlabel('Country')
plt.ylabel('Number of Brands')
plt.show()
```
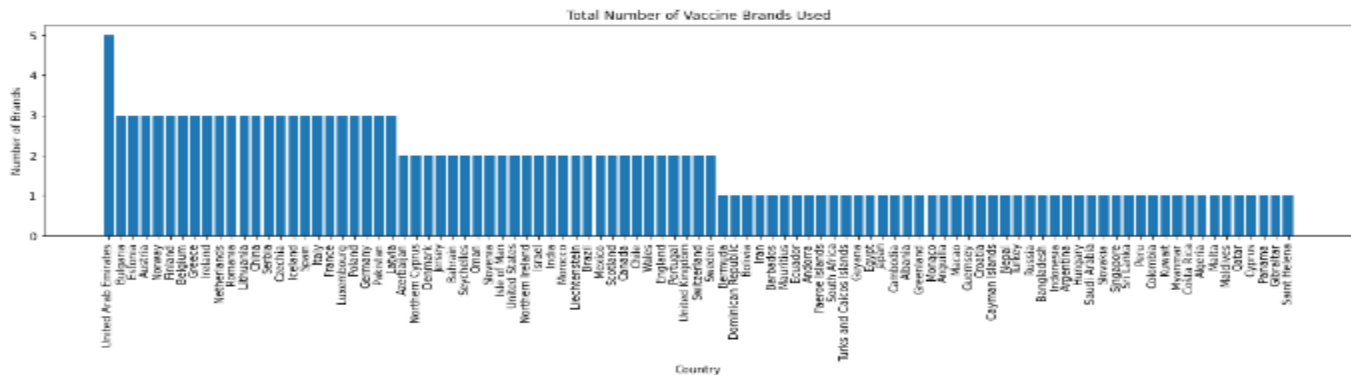
Visualization of the dataset Using Cognos

Visualization using pie plot :
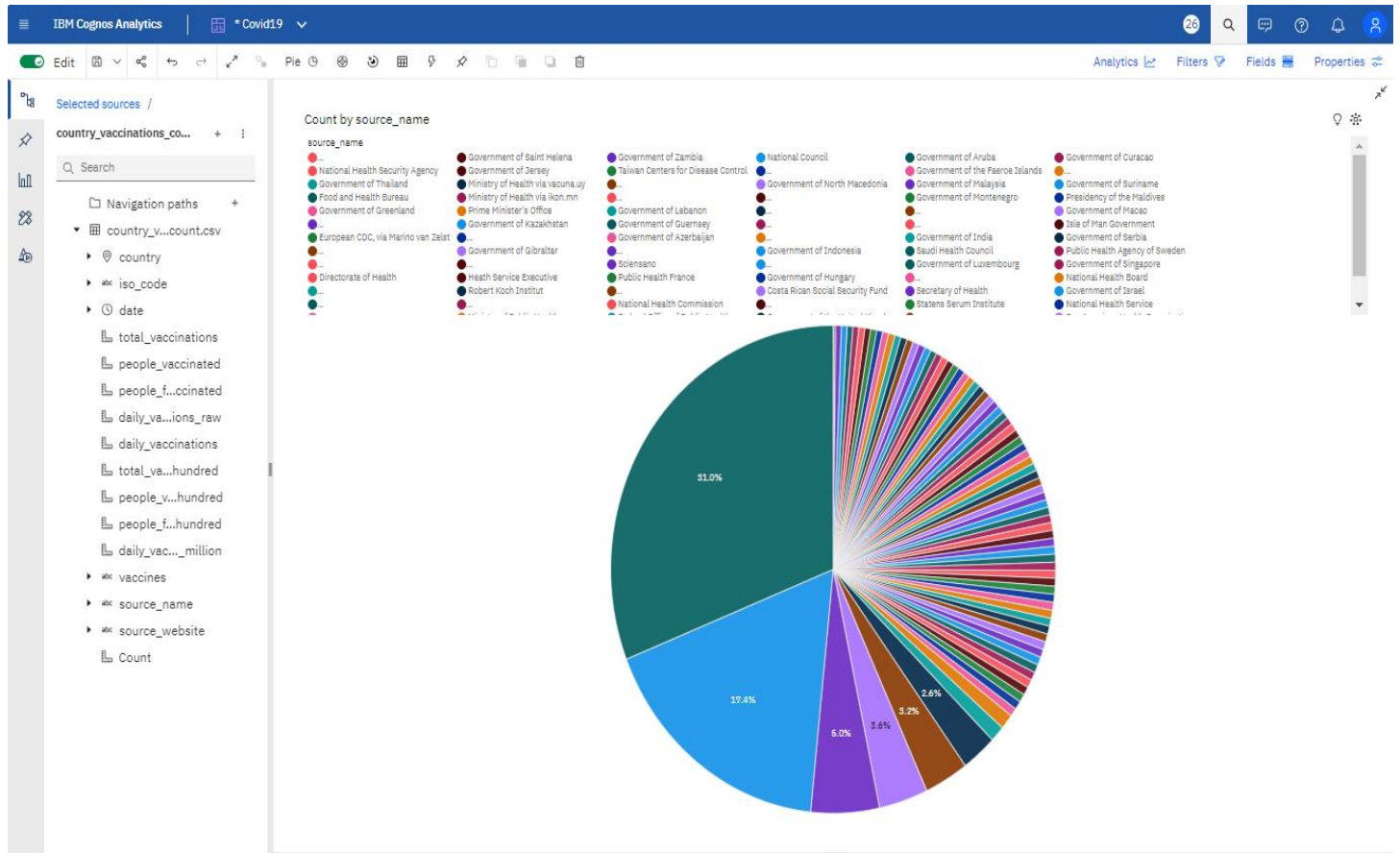
Visualizing the percentage of Source Names for Vaccines using the Pie Chart provided in the Cognos.
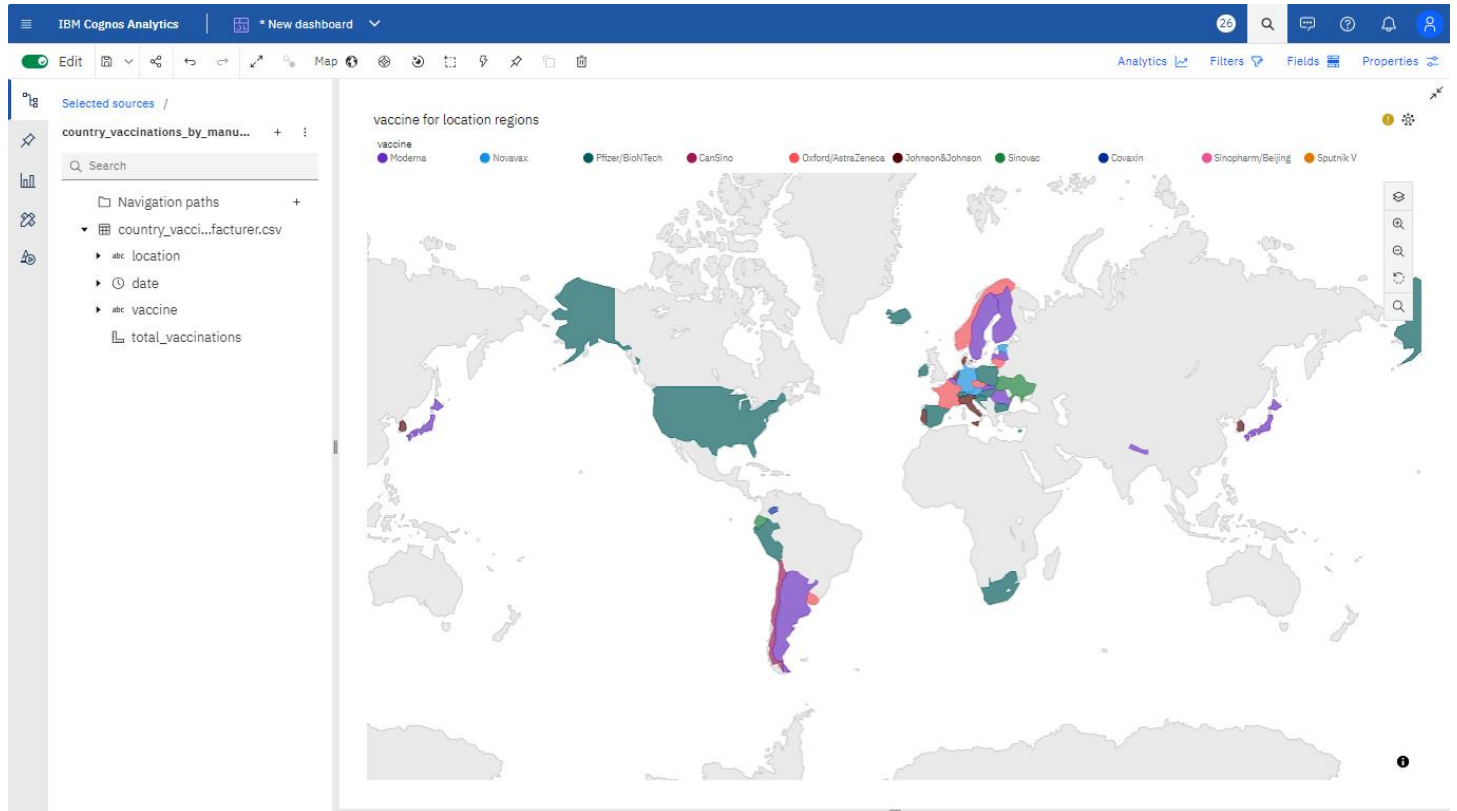
Visualizing using Line Chart :
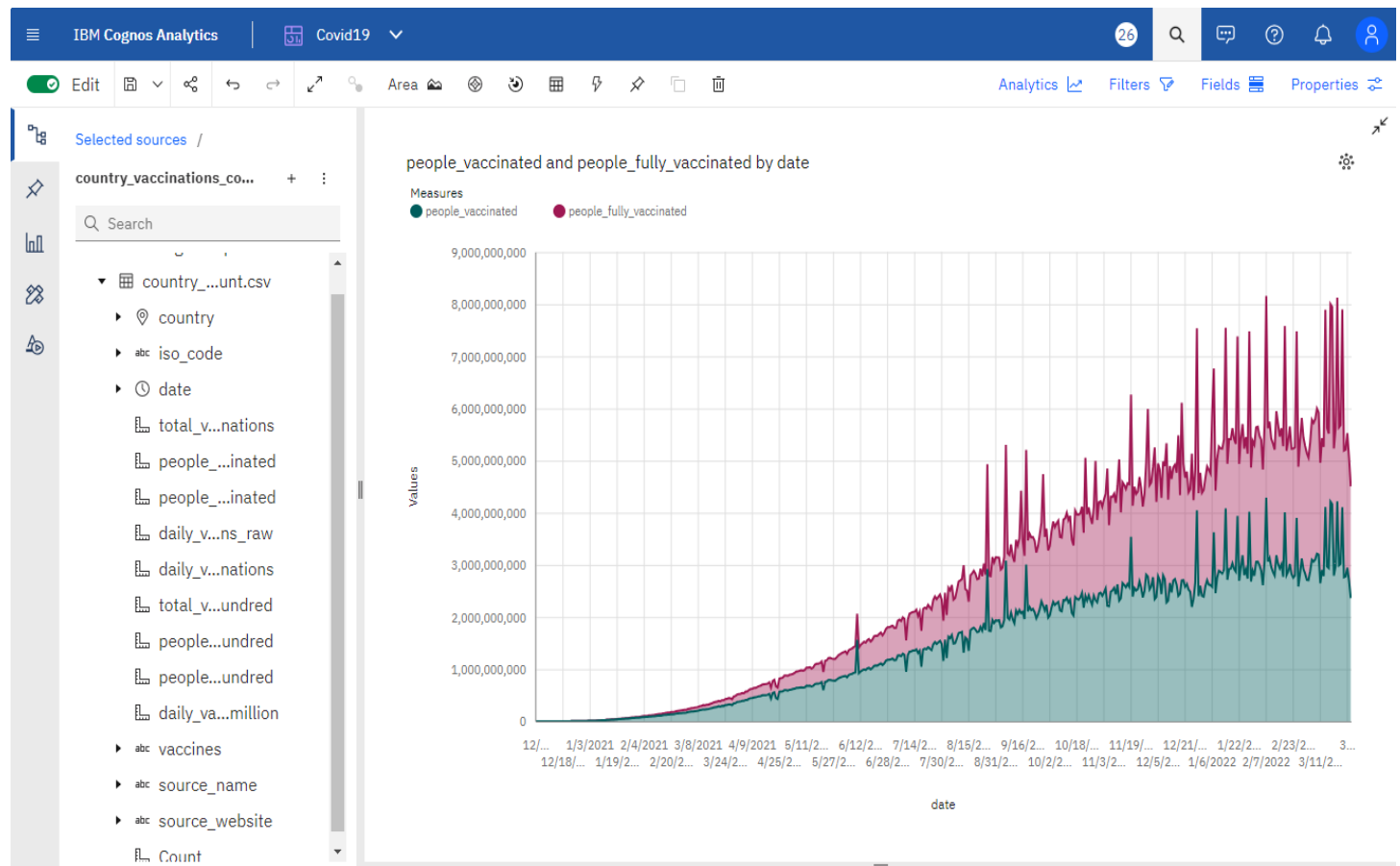
Visualizing the people_vaccinated and people_fully_vaccinated by the date using the Line Chart provided in the Cognos.
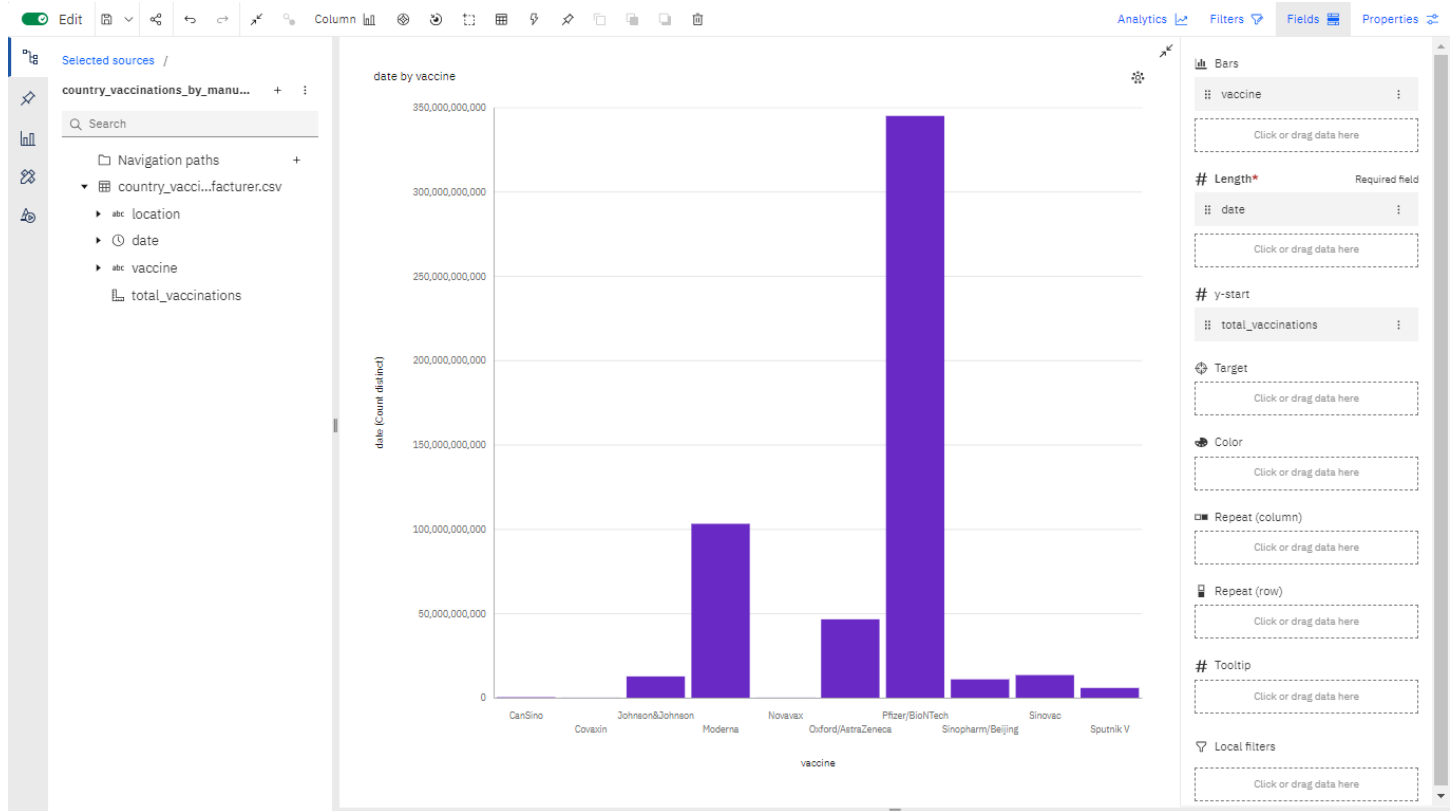
Visualization using Area :
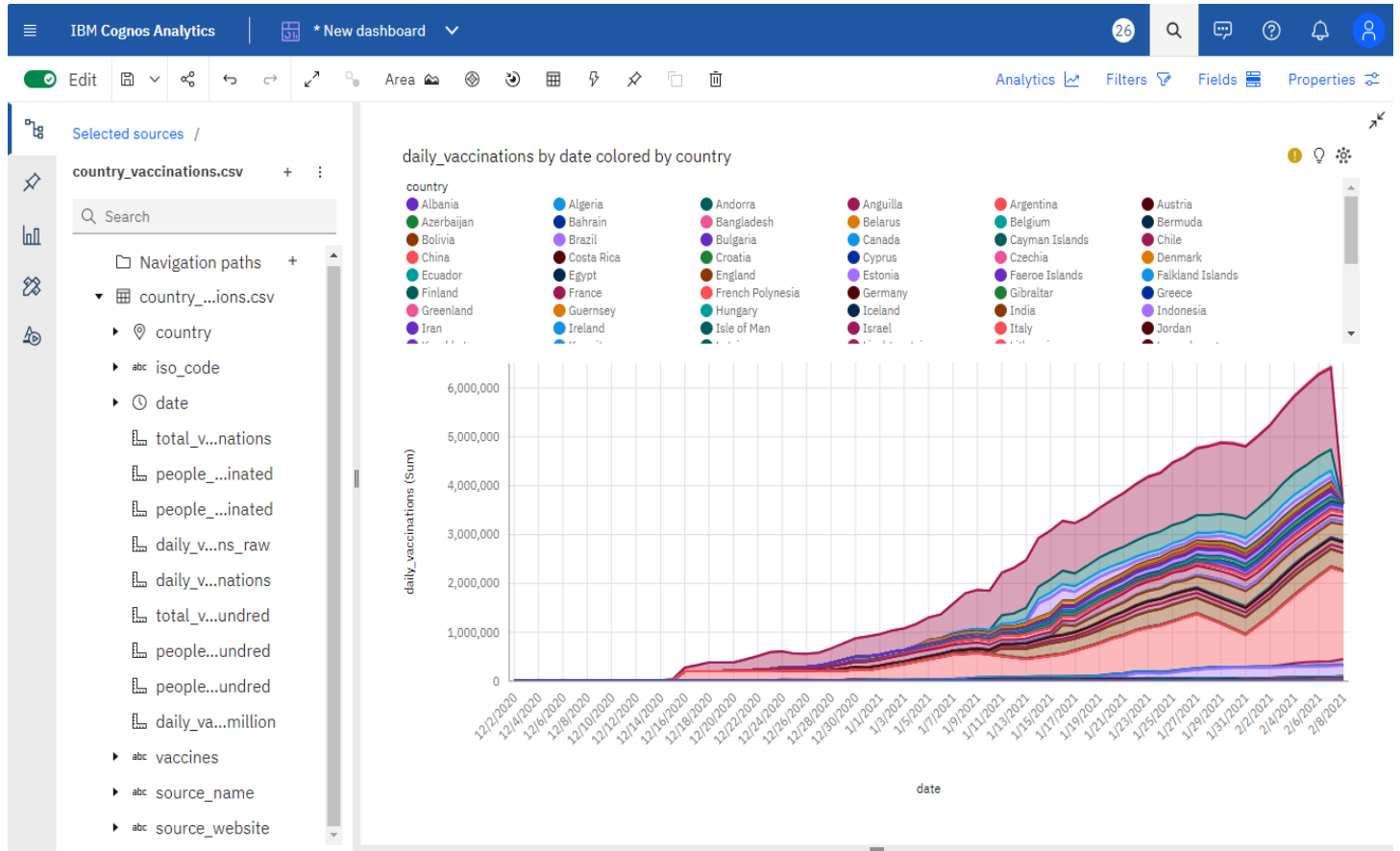
   Visualizing the daily_vaccinations by the date in all the country using the Area Chart provided in the Cognos.

Visualization using Map :

Visualizing the most vaccines used in the locations using the Map Chart provided in the Cognos.

Visualization using Bar Chart :

   Visualizing the amount of vaccines used in the date using the Bar Chart provided in the Cognos.

ADVANTAGES

The advantages of COVID-19 vaccines are multifaceted and have played a crucial role in the global response to the pandemic. Here's an analysis of the key advantages:

1.Disease Prevention and Control:
Analysis: COVID-19 vaccines have demonstrated high efficacy in preventing infection, reducing the severity of illness, and minimizing the risk of hospitalization and death.
Advantage: By effectively controlling the spread of the virus, vaccines contribute to breaking the chain of transmission and preventing widespread outbreaks.

2.Saving Lives:
Analysis: Vaccination significantly reduces the likelihood of severe outcomes and fatalities associated with COVID-19.
Advantage: The primary and most immediate benefit of COVID-19 vaccines is their role in saving lives, particularly among vulnerable populations.

3.Reducing Healthcare Burden:
Analysis: Vaccination decreases the number of severe cases, easing the burden on healthcare systems and preventing overwhelmed hospitals.
Advantage: By preventing a surge in hospitalizations, vaccines ensure that healthcare resources can be efficiently allocated to those in need, enhancing overall healthcare system resilience.

4.Scientific Advancements:
Analysis: The development and deployment of COVID-19 vaccines have driven advancements in vaccine technology and distribution strategies.
Advantage: These advancements have broader implications for future vaccine development, potentially improving our ability to respond to other infectious diseases.

5. Increased Confidence in Public Health Measures:
Analysis: Successful vaccination campaigns contribute to increased public confidence in public health measures.
Advantage: Higher confidence levels promote adherence to preventive measures, vaccination programs, and public health recommendations, enhancing the effectiveness of the overall response.

DISADVANTAGES

While COVID-19 vaccines have been instrumental in controlling the spread of the virus and preventing severe illness, there are some challenges and concerns associated with their deployment. Here's an analysis of the disadvantages of COVID-19 vaccines:

Vaccine Hesitancy:
Analysis: Some individuals are hesitant or unwilling to receive COVID-19 vaccines due to concerns about safety, efficacy, or mistrust in the healthcare system.
Disadvantage: Vaccine hesitancy can impede achieving high levels of population immunity, leaving communities vulnerable to outbreaks and hindering the overall effectiveness of vaccination campaigns.

Unequal Global Distribution:
Analysis: There are significant disparities in global vaccine distribution, with some countries having limited access to vaccines while others secure a surplus.
Disadvantage: Unequal distribution exacerbates global health inequalities, prolongs the duration of the pandemic, and increases the risk of the virus spreading and evolving in areas with low vaccination coverage.

Emergence of Variants:
Analysis: Despite the effectiveness of vaccines, the virus can mutate, leading to the emergence of new variants that may partially evade immunity conferred by existing vaccines.
Disadvantage: Variants may pose challenges to vaccine efficacy and necessitate ongoing efforts to update and adapt vaccines to address evolving strains of the virus.

Limited Long-Term Data:
Analysis: The long-term safety and efficacy of COVID-19 vaccines are still being monitored, as these vaccines were developed and deployed relatively quickly.
Disadvantage: Some individuals may be cautious due to the limited long-term data, even though extensive short-term data and regulatory scrutiny support the safety and efficacy of authorized vaccines.

CONCLUSION

COVID-19 Vaccines are a crucial tool in the fight against the pandemic. They have shown high efficacy rates and have undergone rigorous testing to ensure their safety.

However, vaccine distribution and access must be equitable to ensure that all populations have access to these life-saving vaccines.

In the quest to build a Covid Vaccines Analysis , we have embarked on a critical journey that begins with loading and preprocessing the Covid Vaccines dataset. We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.

Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) is essential for informed decision-making.

Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that it aligns with the requirements of machine learning algorithms.

With these foundational steps completed, our dataset is now primed for the subsequent stages of building and training a Covid Vaccines Analysis .

Covid Vaccines Analysis, as exemplified by this dataset, is a vital component responsible for Vaccination management in the face of managing and mitigating the impact of the global pandemic thereby optimizing vaccination strategies. we have built upon the foundation established in the earlier phases.

This defines project by loading and preprocessing the dataset and performed different analysis and visualization using IBM Cognos.

The need to rapidly develop a vaccine against COVID-19 occurs at a time of great excitement in basic scientific understanding, as well as strategies learned in the past by industry and optimization of regulatory pathways. It is expected that these factors, arising from the global emergency, may redirect the R&D processes for new drugs, especially in times of pandemic.

we have built our project by conducting the Covid-19 vaccines analysis by performing : Data collection, loading, preprocessing, handling missing values, feature scaling, encoding categorical variables, Exploratory data analysis, Statistical analysis and Visualization.

IBM DATA ANALYTICS WITH COGNOS

TEAM NAME : Proj_229800_Team_1

PROJECT : 3101-COVID Vaccines Analysis

TEAM MEMBERS :

- **PAVITHRA V**
- **SHARU DHARSHINI S**
- **SUBATHRA E**
- **SHALINI S**

TEAM LEADER : PAVITHRA V