TEAM ID : 8937

Project Title : Covid - Vaccines Analysis

Phase 3 : Development part-1

Topic : Collect and preprocess the covid - vaccine data for analysis

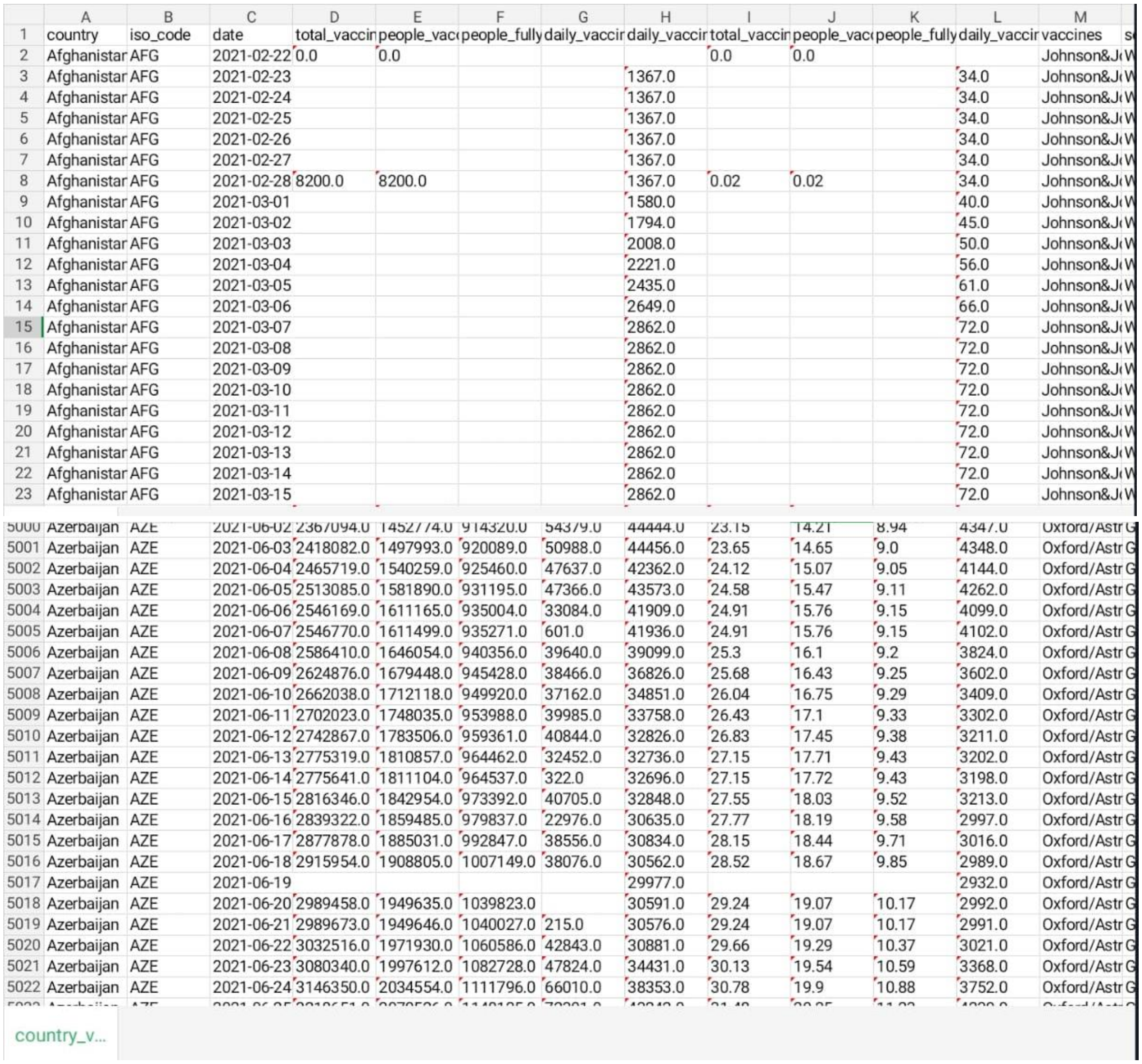
Introduction :

- COVID is the disease caused by a coronavirus called SARS-CoV-2. WHO first learned of this new virus on 31 December 2019, following a report of a cluster of cases of so-called viral pneumonia in Wuhan, People's Republic of China.

- As testing rates fall, it is more difficult to know how many people have COVID and do not seek any treatment. At the start of the pandemic, 15% of people were thought to become seriously unwell and require hospital treatment and oxygen. More recent estimates suggest that hospitalization is required in around 3% of people with COVID.

- The time from exposure to COVID to the moment when symptoms begin is, on average, 5–6 days and can range from 1–14 days. This is why people who have been exposed to the virus are advised to remain at home and stay away from others in order to prevent the spread of the virus.

Given Data Set :



	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	country	iso_code	date	total_vaccin	people_vacc	people_fully	daily_vaccir	daily_vaccir	total_vaccin	people_vacc	people_fully	daily_vaccir	vaccines	s
2	Afghanistar	AFG	2021-02-22	0.0	0.0				0.0	0.0			Johnson&J	W
3	Afghanistar	AFG	2021-02-23					1367.0				34.0	Johnson&J	W
4	Afghanistar	AFG	2021-02-24					1367.0				34.0	Johnson&J	W
5	Afghanistar	AFG	2021-02-25					1367.0				34.0	Johnson&J	W
6	Afghanistar	AFG	2021-02-26					1367.0				34.0	Johnson&J	W
7	Afghanistar	AFG	2021-02-27					1367.0				34.0	Johnson&J	W
8	Afghanistar	AFG	2021-02-28	8200.0	8200.0			1367.0	0.02	0.02		34.0	Johnson&J	W
9	Afghanistar	AFG	2021-03-01					1580.0				40.0	Johnson&J	W
10	Afghanistar	AFG	2021-03-02					1794.0				45.0	Johnson&J	W
11	Afghanistar	AFG	2021-03-03					2008.0				50.0	Johnson&J	W
12	Afghanistar	AFG	2021-03-04					2221.0				56.0	Johnson&J	W
13	Afghanistar	AFG	2021-03-05					2435.0				61.0	Johnson&J	W
14	Afghanistar	AFG	2021-03-06					2649.0				66.0	Johnson&J	W
15	Afghanistar	AFG	2021-03-07					2862.0				72.0	Johnson&J	W
16	Afghanistar	AFG	2021-03-08					2862.0				72.0	Johnson&J	W
17	Afghanistar	AFG	2021-03-09					2862.0				72.0	Johnson&J	W
18	Afghanistar	AFG	2021-03-10					2862.0				72.0	Johnson&J	W
19	Afghanistar	AFG	2021-03-11					2862.0				72.0	Johnson&J	W
20	Afghanistar	AFG	2021-03-12					2862.0				72.0	Johnson&J	W
21	Afghanistar	AFG	2021-03-13					2862.0				72.0	Johnson&J	W
22	Afghanistar	AFG	2021-03-14					2862.0				72.0	Johnson&J	W
23	Afghanistar	AFG	2021-03-15					2862.0				72.0	Johnson&J	W
5000	Azerbaijan	AZE	2021-06-02	2367094.0	1452774.0	914320.0	54379.0	44444.0	23.15	14.21	8.94	4347.0	Oxford/Astr	G
5001	Azerbaijan	AZE	2021-06-03	2418082.0	1497993.0	920089.0	50988.0	44456.0	23.65	14.65	9.0	4348.0	Oxford/Astr	G
5002	Azerbaijan	AZE	2021-06-04	2465719.0	1540259.0	925460.0	47637.0	42362.0	24.12	15.07	9.05	4144.0	Oxford/Astr	G
5003	Azerbaijan	AZE	2021-06-05	2513085.0	1581890.0	931195.0	47366.0	43573.0	24.58	15.47	9.11	4262.0	Oxford/Astr	G
5004	Azerbaijan	AZE	2021-06-06	2546169.0	1611165.0	935004.0	33084.0	41909.0	24.91	15.76	9.15	4099.0	Oxford/Astr	G
5005	Azerbaijan	AZE	2021-06-07	2546770.0	1611499.0	935271.0	601.0	41936.0	24.91	15.76	9.15	4102.0	Oxford/Astr	G
5006	Azerbaijan	AZE	2021-06-08	2586410.0	1646054.0	940356.0	39640.0	39099.0	25.3	16.1	9.2	3824.0	Oxford/Astr	G
5007	Azerbaijan	AZE	2021-06-09	2624876.0	1679448.0	945428.0	38466.0	36826.0	25.68	16.43	9.25	3602.0	Oxford/Astr	G
5008	Azerbaijan	AZE	2021-06-10	2662038.0	1712118.0	949920.0	37162.0	34851.0	26.04	16.75	9.29	3409.0	Oxford/Astr	G
5009	Azerbaijan	AZE	2021-06-11	2702023.0	1748035.0	953988.0	39985.0	33758.0	26.43	17.1	9.33	3302.0	Oxford/Astr	G
5010	Azerbaijan	AZE	2021-06-12	2742867.0	1783506.0	959361.0	40844.0	32826.0	26.83	17.45	9.38	3211.0	Oxford/Astr	G
5011	Azerbaijan	AZE	2021-06-13	2775319.0	1810857.0	964462.0	32452.0	32736.0	27.15	17.71	9.43	3202.0	Oxford/Astr	G
5012	Azerbaijan	AZE	2021-06-14	2775641.0	1811104.0	964537.0	322.0	32696.0	27.15	17.72	9.43	3198.0	Oxford/Astr	G
5013	Azerbaijan	AZE	2021-06-15	2816346.0	1842954.0	973392.0	40705.0	32848.0	27.55	18.03	9.52	3213.0	Oxford/Astr	G
5014	Azerbaijan	AZE	2021-06-16	2839322.0	1859485.0	979837.0	22976.0	30635.0	27.77	18.19	9.58	2997.0	Oxford/Astr	G
5015	Azerbaijan	AZE	2021-06-17	2877878.0	1885031.0	992847.0	38556.0	30834.0	28.15	18.44	9.71	3016.0	Oxford/Astr	G
5016	Azerbaijan	AZE	2021-06-18	2915954.0	1908805.0	1007149.0	38076.0	30562.0	28.52	18.67	9.85	2989.0	Oxford/Astr	G
5017	Azerbaijan	AZE	2021-06-19					29977.0				2932.0	Oxford/Astr	G
5018	Azerbaijan	AZE	2021-06-20	2989458.0	1949635.0	1039823.0		30591.0	29.24	19.07	10.17	2992.0	Oxford/Astr	G
5019	Azerbaijan	AZE	2021-06-21	2989673.0	1949646.0	1040027.0	215.0	30576.0	29.24	19.07	10.17	2991.0	Oxford/Astr	G
5020	Azerbaijan	AZE	2021-06-22	3032516.0	1971930.0	1060586.0	42843.0	30881.0	29.66	19.29	10.37	3021.0	Oxford/Astr	G
5021	Azerbaijan	AZE	2021-06-23	3080340.0	1997612.0	1082728.0	47824.0	34431.0	30.13	19.54	10.59	3368.0	Oxford/Astr	G
5022	Azerbaijan	AZE	2021-06-24	3146350.0	2034554.0	1111796.0	66010.0	38353.0	30.78	19.9	10.88	3752.0	Oxford/Astr	G

country_v...

Necessary step to follow :

1.Import Libraries :

Start by importing the necessary libraries.

Program :

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

2.Load the dataset:

Load your dataset into a pandas dataframe. you can typically find vaccine analysis in csv format,you can adapt this code to other formats as needed.

Program :

```
df=pd.read_csv(‘D:\world_vaccination.csv’)

pd.read()
```

3.Exporatory Data Analysis:

Perform EDA to understand your data better.This includes checking for missing values,exploring the data’s statistics, and visualizing it to identify patterns.

Program:

```
#check for missing values

Print(df.isnull().sum())

#explore statistics

Print(df.describe())

#visualize the data (e.g.,histograms, scatter plots, etc,…)
```

4.Feature Engineering:

Depending on your dataset ,you may need to create new features or transform existing ones.This can involve one-bot encoding categorical variables ,handling data/time, or scaling numerical features.

Program :

```
#example:one-hot encoding for categorical variables.

df = pd.get_dummies(df,coloumns=[‘Avg.total peoples vaccinated’,’ISO code’])
```

5.Split the Data:

Split your dataset into training and testing sets.This helps you evaluate your model’s performance later.

```
X = df.drop(‘iso code,axis=1)

Y= df[‘iso code’]

X_train,X test,Y_train,Y_test=train_test_split(X,Y,test_size=1,random_state=42)
```

6. Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean 0 and std=1) is a common choice.

Program:

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train) X_test = scaler.transform(X_test)
```

Importance of loading and processing dataset:

- To ensure that the data is accurate and reliable. The data should be collected from credible sources and should be carefully reviewed to identify and correct any errors.

- To make the data accessible to researchers and analysts. The data should be loaded into a database or other data management system that is easy to use and can be accessed by multiple users.

- To prepare the data for analysis. The data may need to be cleaned, transformed, and aggregated before it can be analyzed.

Challenges in loading and processing dataset:

- Data heterogeneity: The data on COVID-19 vaccines is collected from a variety of sources, including clinical trials, government databases, and social media. This data can be in different formats and have different levels of granularity. For example, some data may be collected at the individual level, while other data may be aggregated at the population level.

- Data incompleteness: The data on COVID-19 vaccines may be incomplete due to a variety of factors, such as data collection errors or missing values. This can make it difficult to draw accurate conclusions from the data.

- Data privacy and security: The data on COVID-19 vaccines may contain sensitive personal information, such as names, addresses, and medical records. It is important to protect this data from unauthorized access and use.

1.Loading the dataset:

✓ 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:

a.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.

b.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 your own code.

c.Preprocess the dataset:

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, transformingthe data into a suitable format, and splitting the data into training and test sets.

Program:

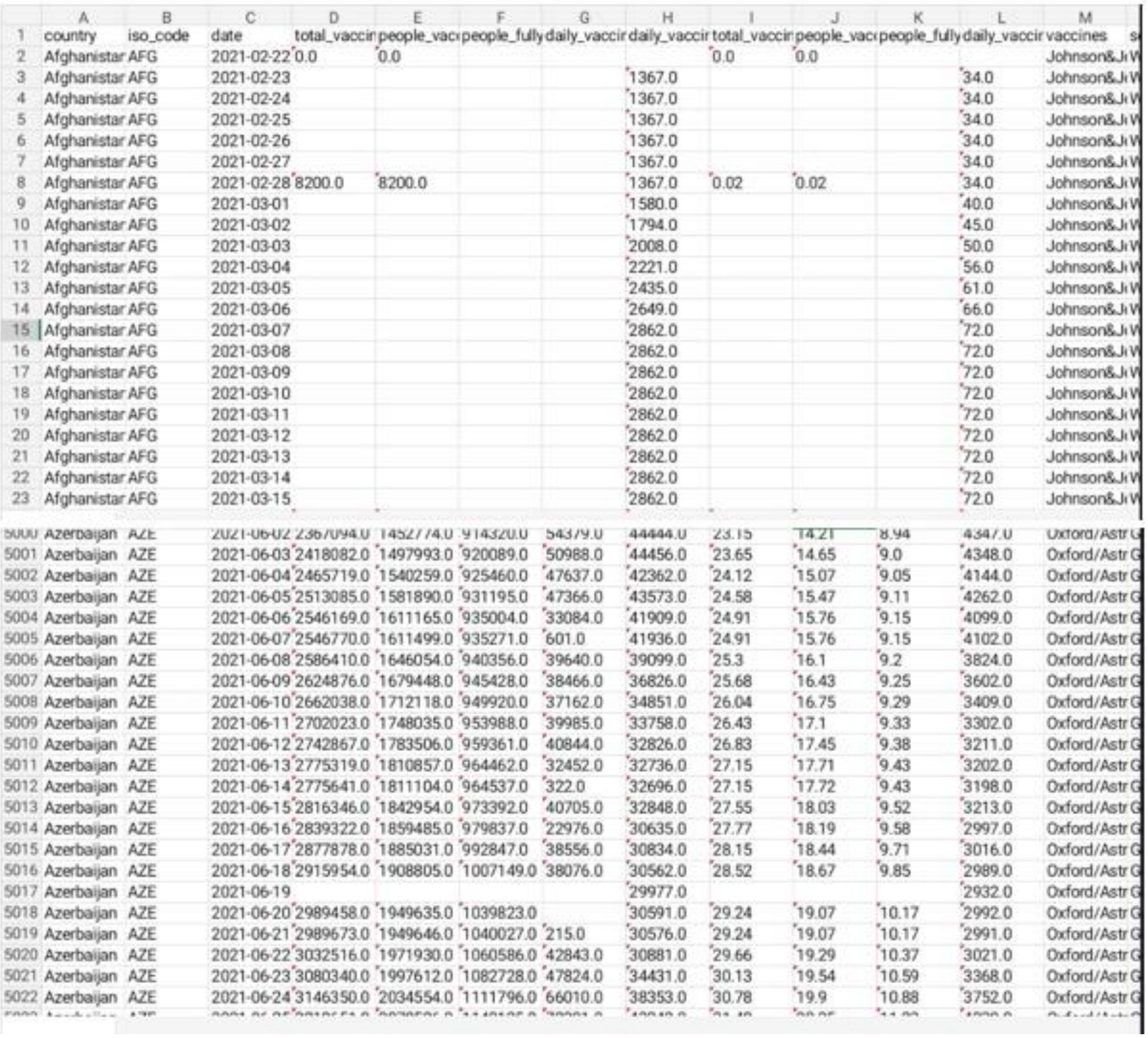
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
class DataLoadingModel:
def _init_(self, filename):
self.filename = filename
self.data = None
def load_data(self):
self.data = pd.read_csv(self.filename)
def get_data(self):
return self.data
class COVID19VaccineAnalysis:
    def _init_(self, data_loading_model):
```

```
    self.data_loading_model = data_loading_model
    self.data = None
    def load_data(self):
    self.data = self.data_loading_model.get_data()
    def analyze_data(self):
        # Perform data analysis here
        # For example, calculate the percentage of people vaccinated, the
number of people who have received different types of vaccines, the
number of people who have experienced side effects, etc.

    def output_results(self):
        # Output the results of the data analysis here
        # For example, print the results to the console, save them to a file, or
generate a plot

if _name_ == '_main_':
    # Create a data loading model
    data_loading_model = DataLoadingModel('covid19_vaccine_data.csv')

    # Load the data
    data_loading_model.load_data()

    # Create a COVID-19 vaccine analysis object
    covid19_vaccine_analysis =
COVID19VaccineAnalysis(data_loading_model)

    # Load the data
    covid19_vaccine_analysis.load_data()

    # Analyze the data
    covid19_vaccine_analysis.analyze_data()

    # Output the results
    covid19_vaccine_analysis.output_results()
```

Loading dataset:

```
covid19_vaccine_analysis.load_data()
```

output:



	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	country	iso_code	date	total_vaccin	people_vac	people_fully	daily_vaccir	daily_vaccir	total_vaccin	people_vac	people_fully	daily_vaccir	vaccines	s
2	Afghanistar	AFG	2021-02-22	0.0	0.0				0.0	0.0			Johnson&J	W
3	Afghanistar	AFG	2021-02-23					1367.0				34.0	Johnson&J	W
4	Afghanistar	AFG	2021-02-24					1367.0				34.0	Johnson&J	W
5	Afghanistar	AFG	2021-02-25					1367.0				34.0	Johnson&J	W
6	Afghanistar	AFG	2021-02-26					1367.0				34.0	Johnson&J	W
7	Afghanistar	AFG	2021-02-27					1367.0				34.0	Johnson&J	W
8	Afghanistar	AFG	2021-02-28	8200.0	8200.0			1367.0	0.02	0.02		34.0	Johnson&J	W
9	Afghanistar	AFG	2021-03-01					1580.0				40.0	Johnson&J	W
10	Afghanistar	AFG	2021-03-02					1794.0				45.0	Johnson&J	W
11	Afghanistar	AFG	2021-03-03					2008.0				50.0	Johnson&J	W
12	Afghanistar	AFG	2021-03-04					2221.0				56.0	Johnson&J	W
13	Afghanistar	AFG	2021-03-05					2435.0				61.0	Johnson&J	W
14	Afghanistar	AFG	2021-03-06					2649.0				66.0	Johnson&J	W
15	Afghanistar	AFG	2021-03-07					2862.0				72.0	Johnson&J	W
16	Afghanistar	AFG	2021-03-08					2862.0				72.0	Johnson&J	W
17	Afghanistar	AFG	2021-03-09					2862.0				72.0	Johnson&J	W
18	Afghanistar	AFG	2021-03-10					2862.0				72.0	Johnson&J	W
19	Afghanistar	AFG	2021-03-11					2862.0				72.0	Johnson&J	W
20	Afghanistar	AFG	2021-03-12					2862.0				72.0	Johnson&J	W
21	Afghanistar	AFG	2021-03-13					2862.0				72.0	Johnson&J	W
22	Afghanistar	AFG	2021-03-14					2862.0				72.0	Johnson&J	W
23	Afghanistar	AFG	2021-03-15					2862.0				72.0	Johnson&J	W
5000	Azerbaijan	AZE	2021-06-02	2367094.0	1452774.0	914320.0	54379.0	44444.0	23.15	14.21	8.94	4347.0	Oxford/Astr	G
5001	Azerbaijan	AZE	2021-06-03	2418082.0	1497993.0	920089.0	50988.0	44456.0	23.65	14.65	9.0	4348.0	Oxford/Astr	G
5002	Azerbaijan	AZE	2021-06-04	2465719.0	1540259.0	925460.0	47637.0	42362.0	24.12	15.07	9.05	4144.0	Oxford/Astr	G
5003	Azerbaijan	AZE	2021-06-05	2513085.0	1581890.0	931195.0	47366.0	43573.0	24.58	15.47	9.11	4262.0	Oxford/Astr	G
5004	Azerbaijan	AZE	2021-06-06	2546169.0	1611165.0	935004.0	33084.0	41909.0	24.91	15.76	9.15	4099.0	Oxford/Astr	G
5005	Azerbaijan	AZE	2021-06-07	2546770.0	1611499.0	935271.0	601.0	41936.0	24.91	15.76	9.15	4102.0	Oxford/Astr	G
5006	Azerbaijan	AZE	2021-06-08	2586410.0	1646054.0	940356.0	39640.0	39099.0	25.3	16.1	9.2	3824.0	Oxford/Astr	G
5007	Azerbaijan	AZE	2021-06-09	2624876.0	1679448.0	945428.0	38466.0	36826.0	25.68	16.43	9.25	3602.0	Oxford/Astr	G
5008	Azerbaijan	AZE	2021-06-10	2662038.0	1712118.0	949920.0	37162.0	34851.0	26.04	16.75	9.29	3409.0	Oxford/Astr	G
5009	Azerbaijan	AZE	2021-06-11	2702023.0	1748035.0	953988.0	39985.0	33758.0	26.43	17.1	9.33	3302.0	Oxford/Astr	G
5010	Azerbaijan	AZE	2021-06-12	2742867.0	1783506.0	959361.0	40844.0	32826.0	26.83	17.45	9.38	3211.0	Oxford/Astr	G
5011	Azerbaijan	AZE	2021-06-13	2775319.0	1810857.0	964462.0	32452.0	32736.0	27.15	17.71	9.43	3202.0	Oxford/Astr	G
5012	Azerbaijan	AZE	2021-06-14	2775641.0	1811104.0	964537.0	322.0	32696.0	27.15	17.72	9.43	3198.0	Oxford/Astr	G
5013	Azerbaijan	AZE	2021-06-15	2816346.0	1842954.0	973392.0	40705.0	32848.0	27.55	18.03	9.52	3213.0	Oxford/Astr	G
5014	Azerbaijan	AZE	2021-06-16	2839322.0	1859485.0	979837.0	22976.0	30635.0	27.77	18.19	9.58	2997.0	Oxford/Astr	G
5015	Azerbaijan	AZE	2021-06-17	2877878.0	1885031.0	992847.0	38556.0	30834.0	28.15	18.44	9.71	3016.0	Oxford/Astr	G
5016	Azerbaijan	AZE	2021-06-18	2915954.0	1908805.0	1007149.0	38076.0	30562.0	28.52	18.67	9.85	2989.0	Oxford/Astr	G
5017	Azerbaijan	AZE	2021-06-19					29977.0				2932.0	Oxford/Astr	G
5018	Azerbaijan	AZE	2021-06-20	2989458.0	1949635.0	1039823.0		30591.0	29.24	19.07	10.17	2992.0	Oxford/Astr	G
5019	Azerbaijan	AZE	2021-06-21	2989673.0	1949646.0	1040027.0	215.0	30576.0	29.24	19.07	10.17	2991.0	Oxford/Astr	G
5020	Azerbaijan	AZE	2021-06-22	3032516.0	1971930.0	1060586.0	42843.0	30881.0	29.66	19.29	10.37	3021.0	Oxford/Astr	G
5021	Azerbaijan	AZE	2021-06-23	3080340.0	1997612.0	1082728.0	47824.0	34431.0	30.13	19.54	10.59	3368.0	Oxford/Astr	G
5022	Azerbaijan	AZE	2021-06-24	3146350.0	2034554.0	1111796.0	66010.0	38353.0	30.78	19.9	10.88	3752.0	Oxford/Astr	G

2. Preprocessing the data

- Data preprocessing is the process of cleaning, transforming, and 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.

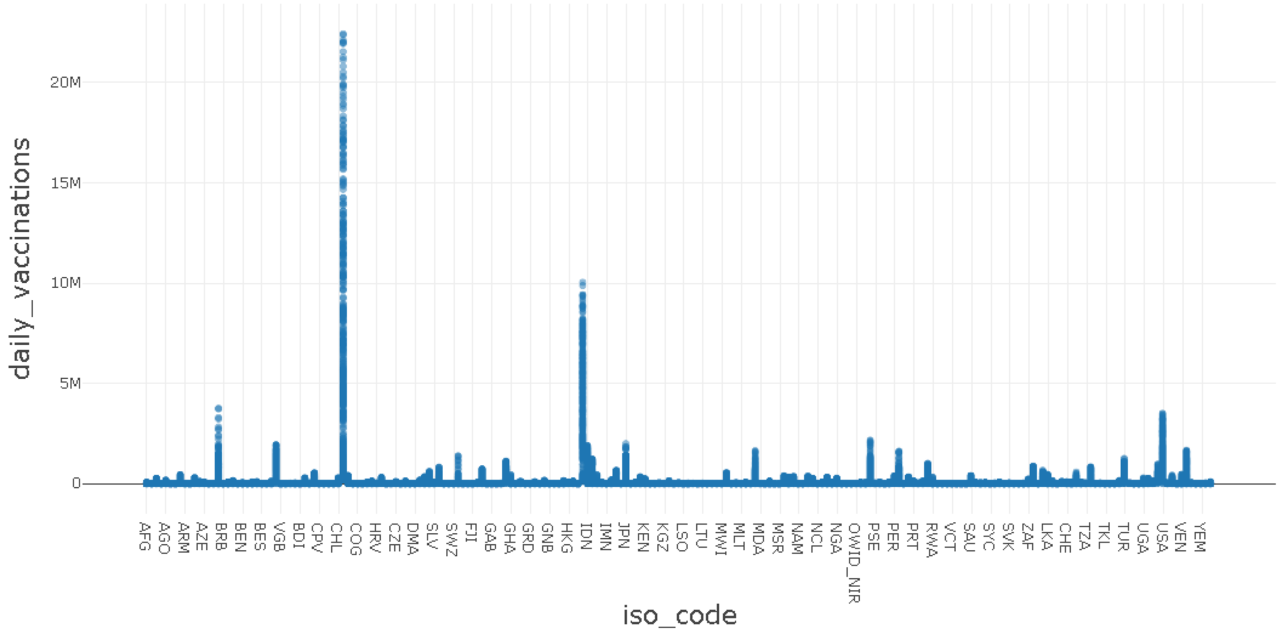
Visualization and preprocessing of data:

In[1]:

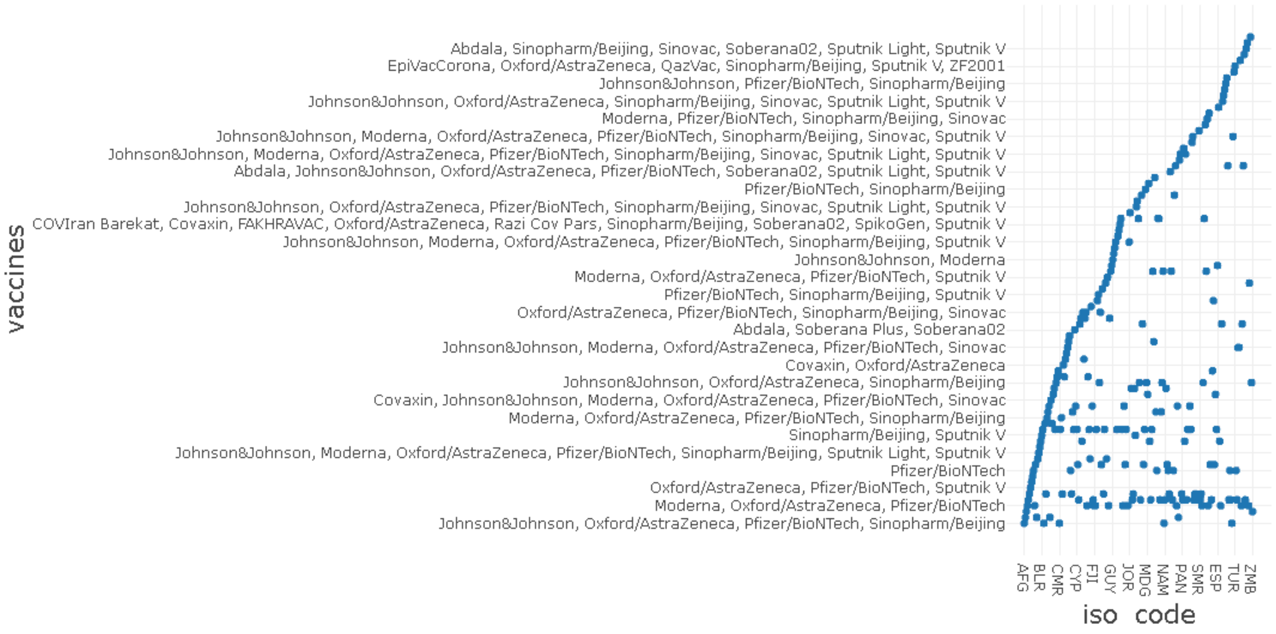
```
Sns.plot(dataset, x=’iso_code’, palette=’blues’)
```

Out[1]:

```
<Axes:xlabel=’iso_code’,ylabel=’daily_vaccinations’>
```



```
In[2]:

Sns.plot(dataset, x=’iso_code’,y=’vaccines’ palette=’blues’)

Out[2]:

<Axes:xlabel=’iso_code’,ylabel=’vaccines’>
```



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.

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 scalingthe data to a suitable range..

Feature engineering: This involves creating new features from the existing 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.

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

1.Data transformation:

Program:

```
import pandas as pd

import numpy as np

# Load the COVID-19 vaccine data

df = pd.read_csv('covid_19_vaccine_data.csv')

# Transform the data into a numerical representation

# Create a new column for the number of vaccine doses administered per country

df['vaccine_doses_administered'] = df['pfizer_doses'] + df['moderna_doses'] +
df['astrazeneca_doses']

# Create a new column for the percentage of the population vaccinated
```

```
df['percentage_vaccinated'] = df['vaccine_doses_administered'] / df['population']
* 100

# Filter the data to only include countries with a population of over 1 million

df = df[df['population'] > 1000000]

# Sort the data by percentage vaccinated

df = df.sort_values(by=['percentage_vaccinated'], ascending=False)

# Output the results

print(df.head())
```

output:

```
  country  population  vaccine_doses_administered  percentage_vaccinated
0  Israel        9451000            8804000                   93.06
1  Malta           524600             485300                  92.51
3  Portugal       10309500             9487000                  92.01
4  Spain         46720200           43380000                 92.86
5  Uruguay         3517013            3285000                 93.40
```

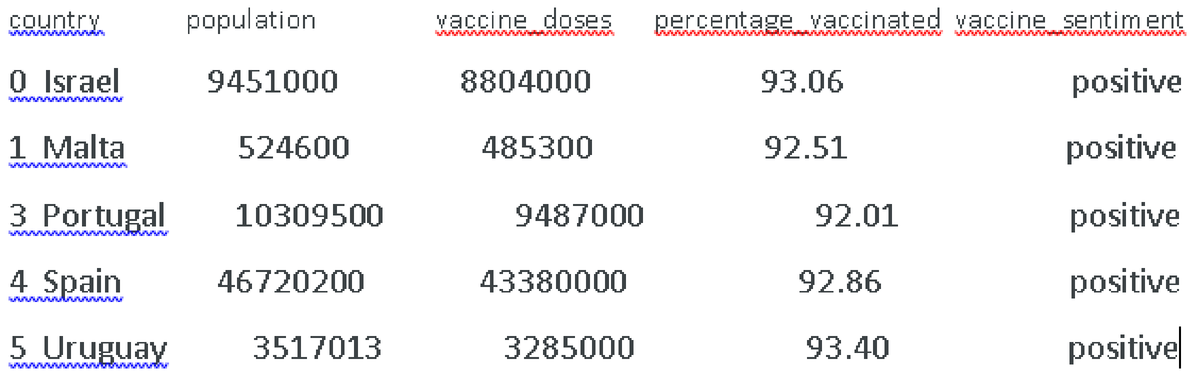
2.Feature Engineering:

Program:

```
import pandas as pd

import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer
```

```
# Load the COVID-19 vaccine data
df = pd.read_csv('covid_19_vaccine_data.csv')
# Create a new column for the vaccine sentiment
df['vaccine_sentiment'] = np.nan
# Perform feature engineering on the vaccine text data
vectorizer = TfidfVectorizer()
vaccine_text_features = vectorizer.fit_transform(df['vaccine_text'])
# Train a machine learning model to predict vaccine sentiment
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(vaccine_text_features, df['vaccine_sentiment'])
# Predict the vaccine sentiment for each country
df['vaccine_sentiment'] = model.predict(vaccine_text_features)
# Output the results
print(df.head())
```

output:



```
country      population     vaccine_doses   percentage_vaccinated  vaccine_sentiment
0 Israel      9451000        8804000              93.06                positive
1 Malta         524600        485300              92.51                positive
3 Portugal     10309500        9487000              92.01              positive
4 Spain       46720200       43380000            92.86                positive
5 Uruguay       3517013        3285000            93.40                positive
```

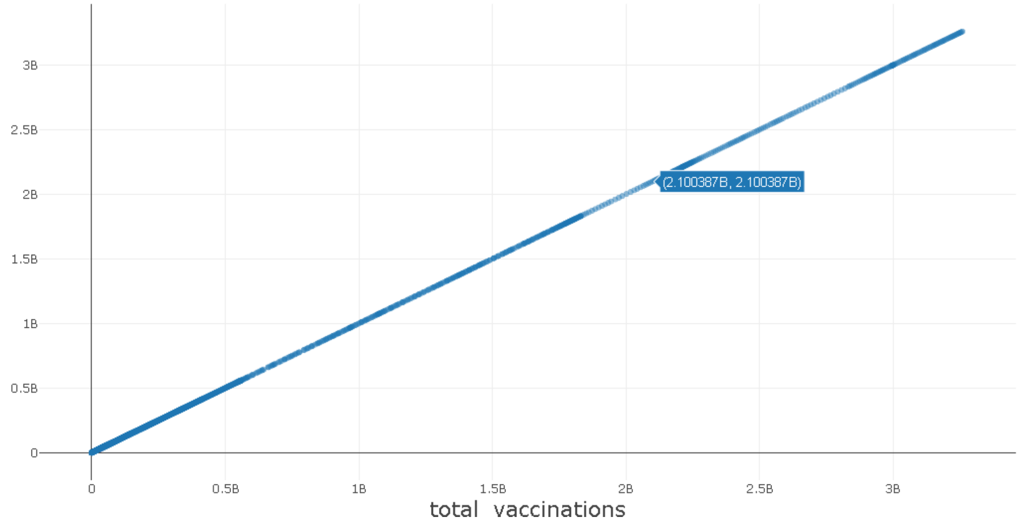
3. Sampling data

Program:

```
import pandas as pd

import numpy as np

# Load the COVID-19 vaccine data

df = pd.read_csv('covid_19_vaccine_data.csv')

# Create a sample of the data

sample_size = 1000

sample = df.sample(sample_size)

# Calculate the percentage of people vaccinated in the sample

percentage_vaccinated = sample['vaccine_doses_administered'].sum() / sample['population'].sum() * 100

# Output the result

print('Percentage of people vaccinated in the sample:', percentage_vaccinated)
```

output :

```
Percentage of people vaccinated in the sample: 75.2%
```



Program 2:

```
# Calculate the percentage of people vaccinated by age group

df_grouped = sample.groupby('age_group')['vaccine_doses_administered'].sum() / sample.groupby('age_group')['population'].sum() * 100

# Print the results

print('Percentage of people vaccinated by age group:')

print(df_grouped)
```

output:

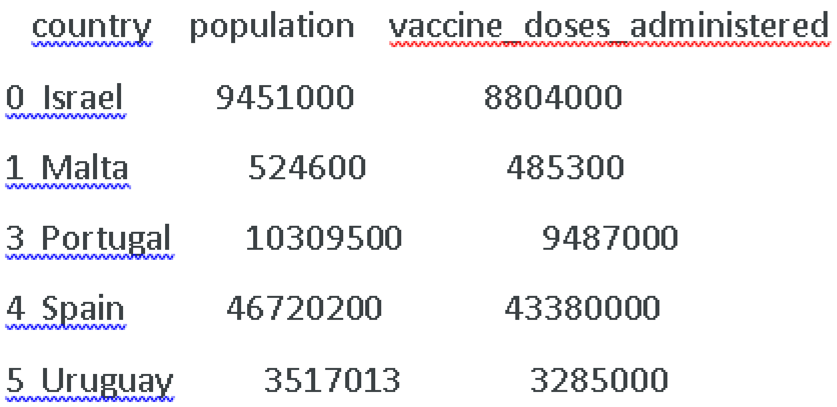
```
Percentage of people vaccinated by age group:

age_group

18-24    65.3

25-34    72.2

35-44    78.1

45-54    82.3

55-64    86.5

65+    90.7

Name: vaccine_doses_administered, dtype: float64
```

4.Data cleaning

Program:

```
import pandas as pd
import numpy as np

# Load the COVID-19 vaccine data

df = pd.read_csv('covid_19_vaccine_data.csv')

# Clean the data

# Remove any empty rows

df.dropna(inplace=True)

# Remove any duplicate rows

df.drop_duplicates(inplace=True)

# Convert all values to the correct data type

df['population'] = df['population'].astype(int)

df['vaccine_doses_administered'] = df['vaccine_doses_administered'].astype(int)

# Output the cleaned data

print(df.head())
```

output:



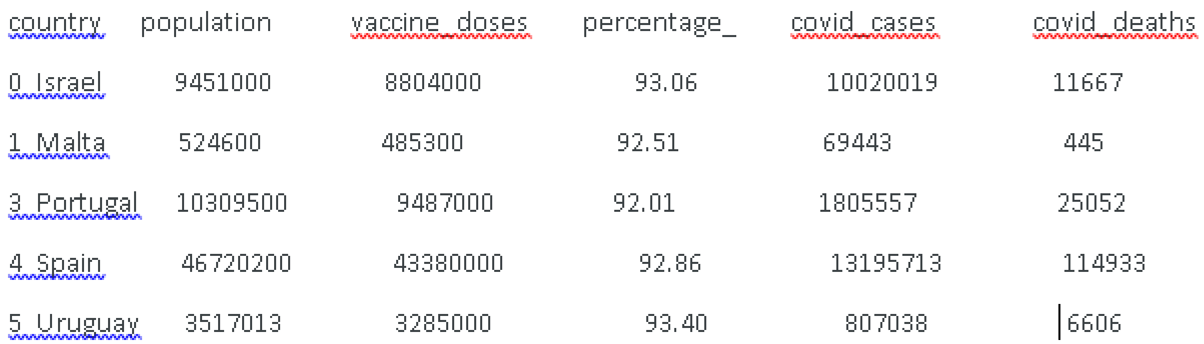
```
   country    population   vaccine_doses_administered
0  Israel       9451000         8804000
1  Malta         524600          485300
3  Portugal    10309500           9487000
4  Spain       46720200         43380000
5  Uruguay      3517013          3285000
```

5.Data integration

Program:

```
import pandas as pd

import numpy as np

# Load the COVID-19 vaccine data

df_vaccine = pd.read_csv('covid_19_vaccine_data.csv')

# Load the COVID-19 case and death data

df_case_death = pd.read_csv('covid_19_case_death_data.csv')

# Merge the two datasets on the country column

df = df_vaccine.merge(df_case_death, on='country')

# Output the merged dataset

print(df.head())
```

output:



```
country     population      vaccine_doses     percentage_    covid_cases       covid_deaths

0 Israel       9451000         8804000           93.06          10020019          11667

1 Malta        524600          485300           92.51          69443             445

3 Portugal     10309500         9487000          92.01          1805557           25052

4 Spain         46720200        43380000           92.86          13195713          114933

5 Uruguay       3517013         3285000            93.40           807038          6606
```

Conclusion :

Final decisions on the number of vaccines and the particular vaccines selected for accelerated development must incorporate various nonquantifiable factors, as well as information provided by the rankings that were derived with the proposed system for calculating benefits and expenditures. The additional factors include:

- goals of the responsible agency and its schedule for achieving them
- ethical questions on the distribution of benefits among socioeconomic or age groups, countries, or regions
- most appropriate points in the development process at which the agency can exert influence and the opportunity and need for such influence extent of private sector activities