COVID VACCINE ANALYSIS

Data Analytics with cognos – Phase 3

DOCUMENTATION

Team Members:

1.SATHANA S(au613021205046)

2.RASIGA J (au613021205040)

3.RAMYA R(au613021205038)

4.KAVIYA R(au613021205024)

5.SAJEETHA BRITTY E(au613021205042)

Problem Definition:

Start the data analysis by loading and preprocessing the dataset. Load the dataset using python and data manipulation libraries (e.g., pandas).

Dataset Link:

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

Overview of the process:

1.Import Libraries:

Begin by importing the necessary libraries, such as pandas for data manipulation.

2.Load the Dataset:

Use pd.read_csv() or other appropriate methods to load your dataset into a pandas DataFrame.

3.Explore the Dataset:

Display the initial rows, check for missing values, and explore basic statistics to understand the structure and content of the data.

4.Handle Missing Values:

Decide on an appropriate strategy for dealing with missing values, such as dropping rows or filling values based on a specific strategy.

5.Additional Preprocessing Steps:

Depending on the nature of your data, consider additional preprocessing steps such as feature scaling, handling outliers, processing date-time features, dealing with text data, feature engineering, or discretization.

6.Save Preprocessed Dataset (Optional):

Save the preprocessed dataset to a new file if significant changes have been made.

Loading the dataset:

1.Importing libraries

Here, for preprocessing the dataset and manipulate the data, pandas is the library used to frame the data.

Code:

```
import pandas as pd
```

2.Loading the dataset

In this step, we are framing the data into the table using DataFrame in pandas, and display the head or 5 rows of the dataset.

Code:

```
# Replace with the actual filename
file_path="C:/Users/91962/Documents/country_vaccinations.csv"
df = pd.read_csv(file_path)
```

Preprocessing the dataset

3.Explore the dataset:

After framing data, the first few or five rows of the data in displayed using the head() function.

Code:

```
print(df.head())
```

OUTPUT:

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

   people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \

0                  NaN                NaN                    NaN
1                  NaN                NaN                   1367.0
2                  NaN                NaN                   1367.0
3                  NaN                NaN                   1367.0
4                  NaN                NaN                   1367.0

   total_vaccinations_per_hundred  people_vaccinated_per_hundred  \

0                       0.0                               0.0
1                       NaN                               NaN
2                       NaN                               NaN
3                       NaN                               NaN
4                       NaN                               NaN

   people_fully_vaccinated_per_hundred  daily_vaccinations_per_million  \

0                            NaN                              NaN
1                            NaN                              34.0
2                            NaN                              34.0
3                            NaN                              34.0
4                            NaN                              34.0

                                  vaccines  \

0  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
1  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
2  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
3  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
4  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
```

```
   source_name                 source_website

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

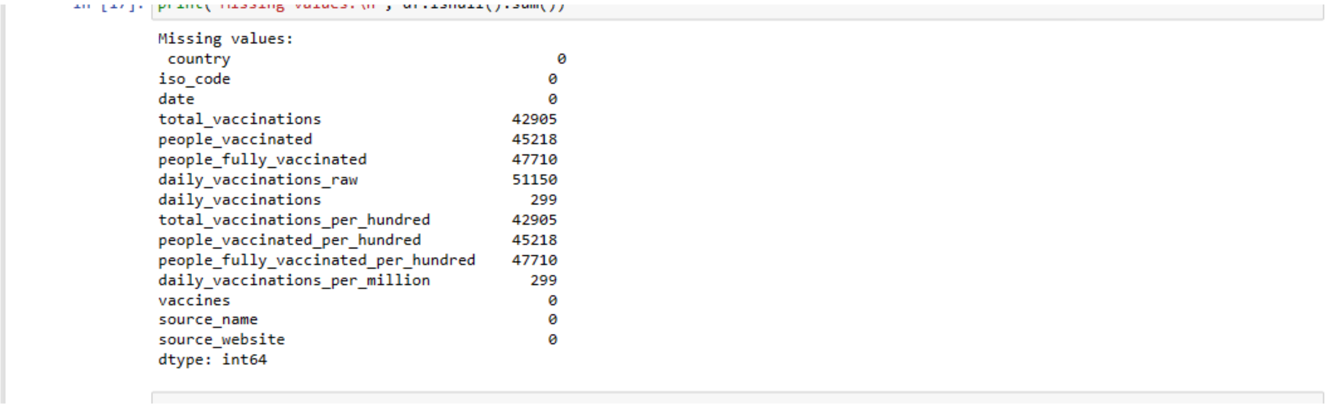
4.Check for missing values:

In this step, the missing values or null values, if it present in the data are separated and number of null values are shown through this code.

Code:

print("Missing values:\n", df.isnull().sum())

OUTPUT:



```
In [17]: print("Missing values:\n", df.isnull().sum())

Missing values:
 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
```

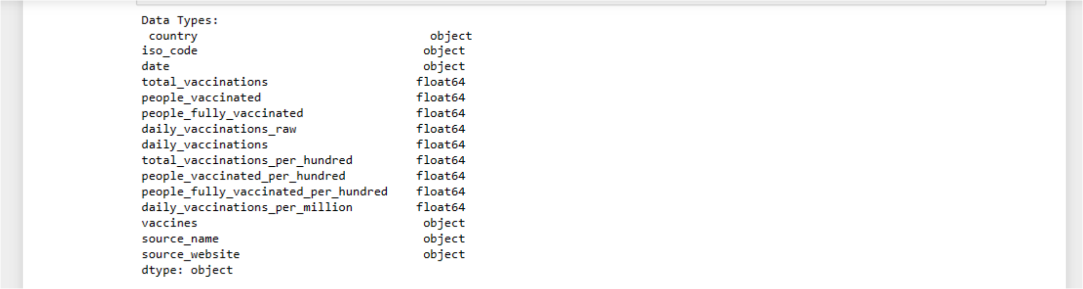
5.Check datatype:

In this step, the data type of the columns are discussed

Code:

print("Data Types:\n", df.dtypes)

OUTPUT:



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

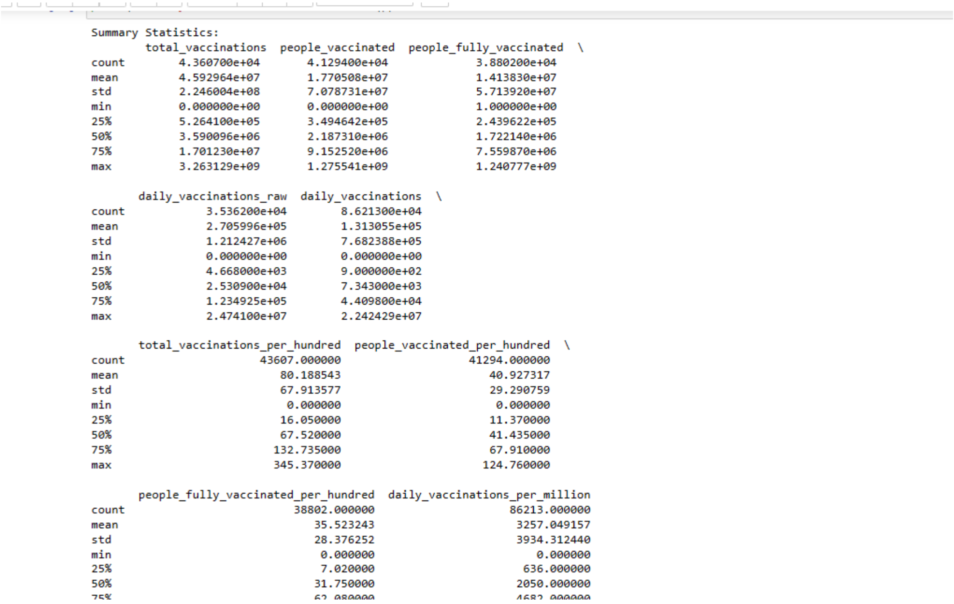
6.Check basic statistics:

The statistics of the columns such as count, mean, std, min, max, 25%, 50%, 75% are shown through the describe() function command.

Code:

```
print("Summary Statistics:\n", df.describe())
```

OUTPUT:



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

      daily_vaccinations_raw  daily_vaccinations  \
count           3.536200e+04        8.621300e+04
mean            2.705996e+05        1.313055e+05
std             1.212427e+06        7.682388e+05
min             0.000000e+00        0.000000e+00
25%             4.668000e+03        9.000000e+02
50%             2.530900e+04        7.343000e+03
75%             1.234925e+05        4.409800e+04
max             2.474100e+07        2.242429e+07

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

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

7.Additional Preprocessing steps:

Perform any other preprocessing steps that are specific to your dataset and analysis goals. This may include scaling numeric features, handling outliers, or creating new features.

8.Saving Preprocessed dataset:

In this step, if we made substantial changes to the dataset and want to save the preprocessed version, you can use the following Code.

Code:

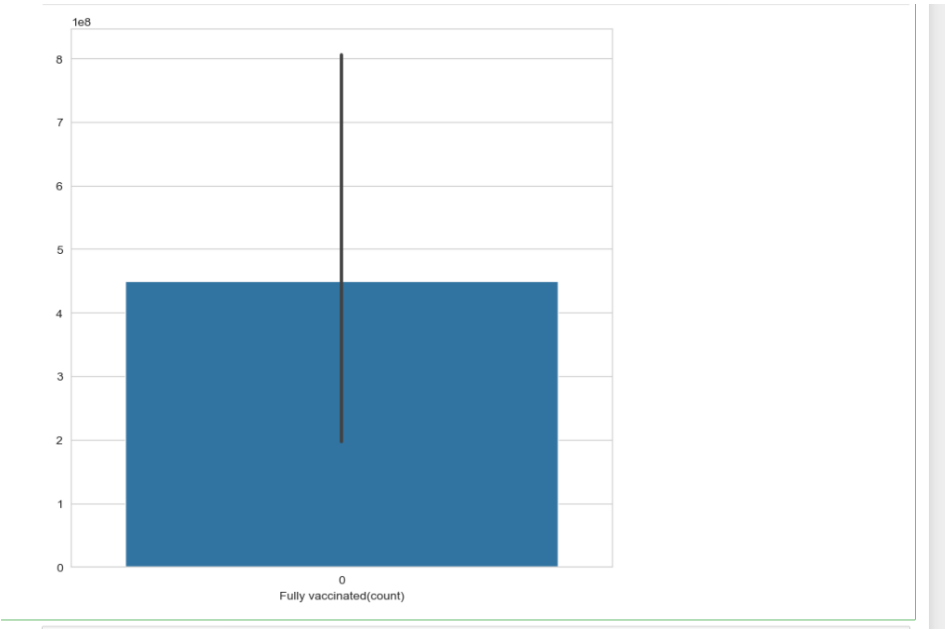
```
 # Save the preprocessed dataset to a new CSV file
df.to_csv('preprocessed_dataset.csv', index=False)
```

DATA VISUALIZATION:

BAR PLOT:

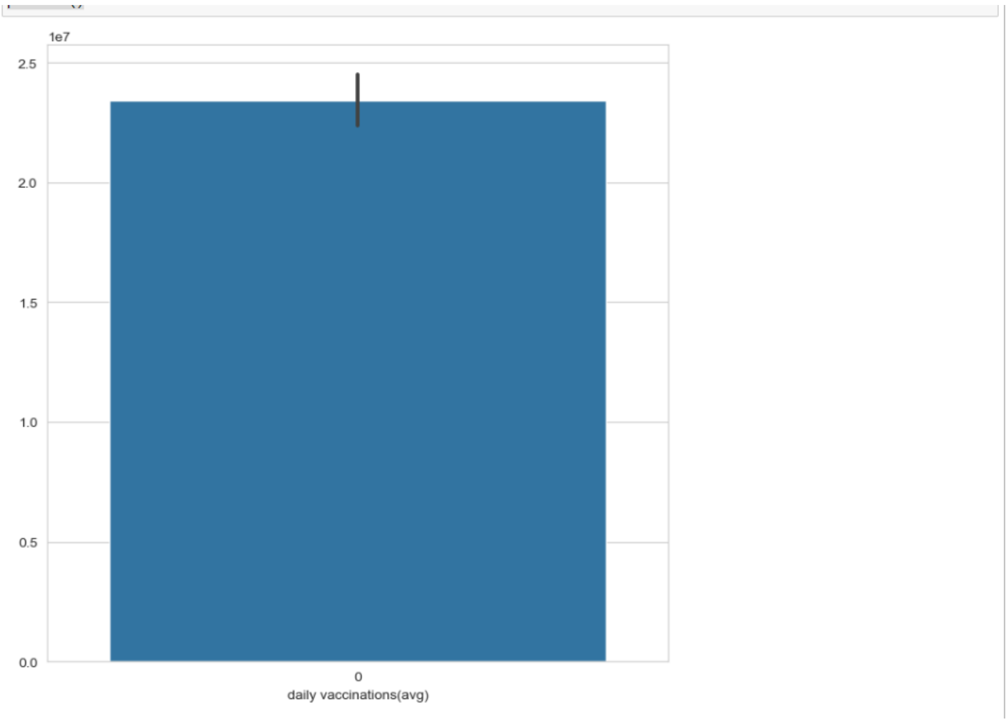
```
sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax= sns.barplot(x.values)
ax.set_xlabel("Fully vaccinated(count)")
plt.show()
```

OUTPUT:



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

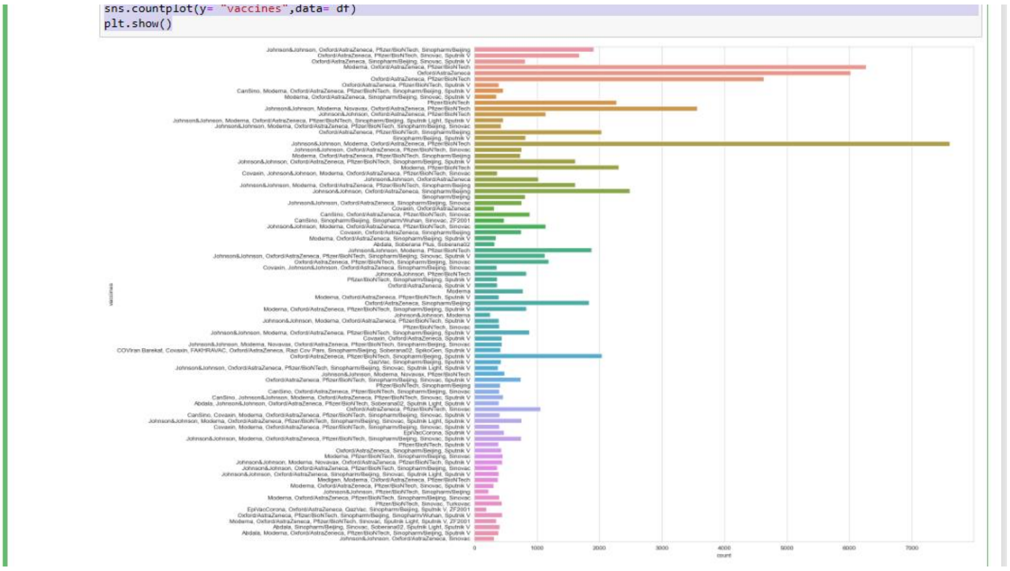
OUTPUT:



COUNT PLOT:

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

OUTPUT:

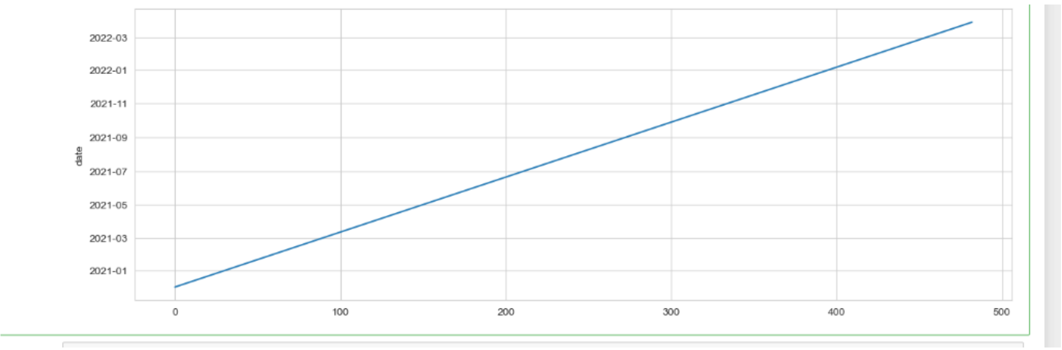


```
sns.countplot(y= "vaccines",data= df)
plt.show()
```

LINE PLOT:

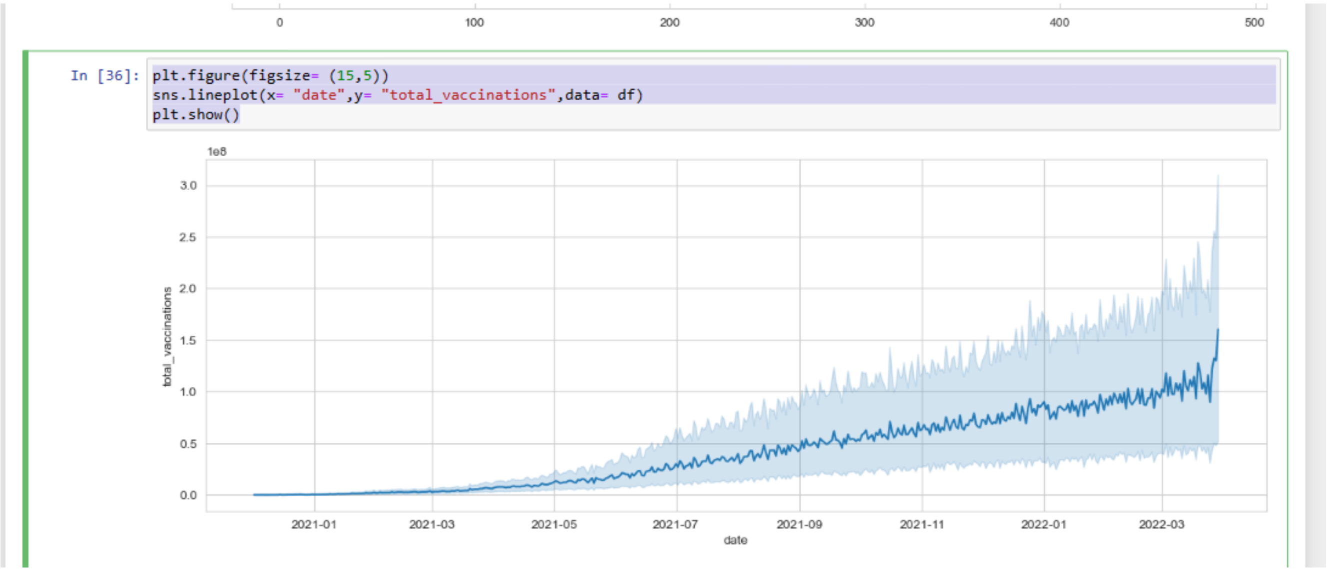
x= df.groupby("date").daily_vaccinations.sum()
plt.figure(figsize= (15,5))
sns.lineplot(x.index)
plt.show()

OUTPUT:



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

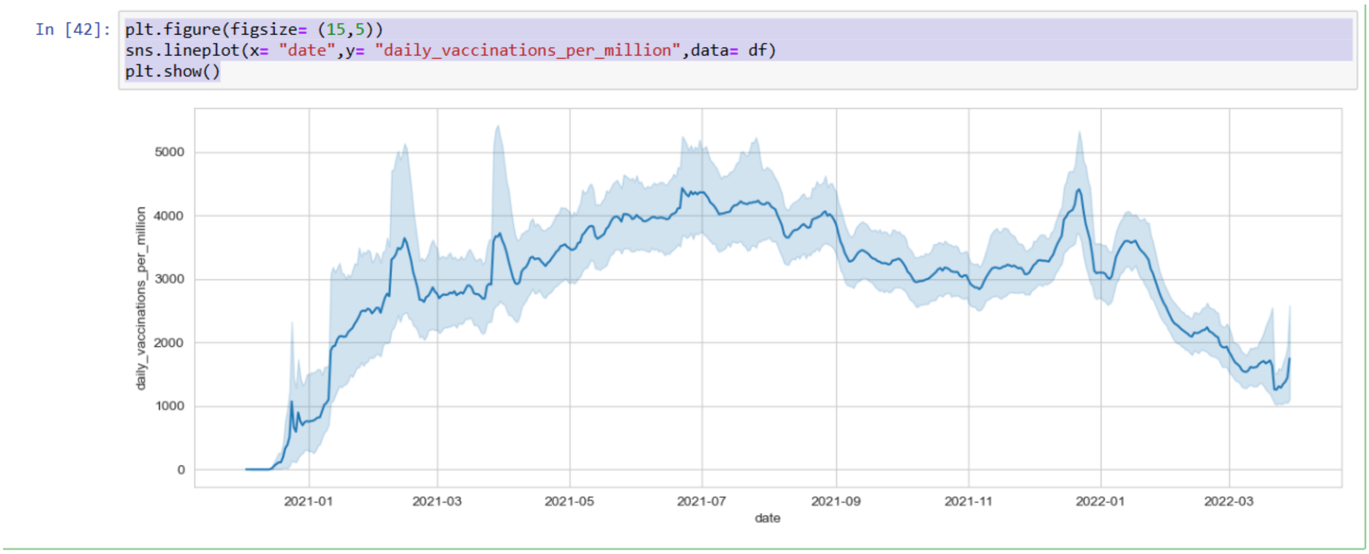
OUTPUT:



```
In [36]: plt.figure(figsize= (15,5))
         sns.lineplot(x= "date",y= "total_vaccinations",data= df)
         plt.show()
```

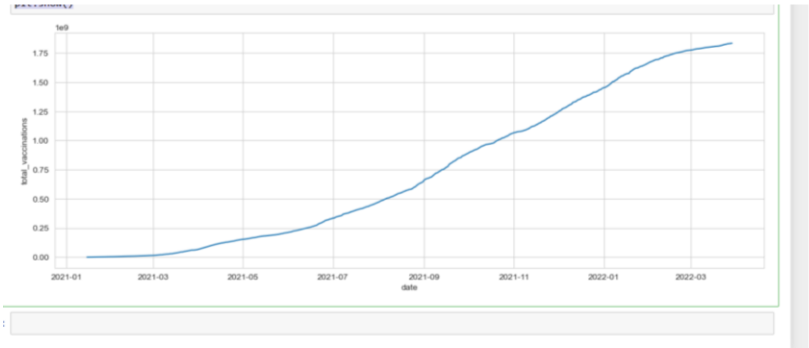
```
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "daily_vaccinations_per_million",data= df)
plt.show()
```

OUTPUT:



```
In [42]: plt.figure(figsize= (15,5))
         sns.lineplot(x= "date",y= "daily_vaccinations_per_million",data= df)
         plt.show()
         plt.figure(figsize= (15,5))
         sns.lineplot(x= "date",y= "total_vaccinations",data= df[df["country"]=="India"])
         plt.show()
```

OUTPUT:



CONCLUSION:

In conclusion, the outlined data loading and preprocessing steps provide a foundational framework for preparing a dataset for analysis in Python using the pandas library. By following these steps, you can ensure that your data is in a suitable format and quality for further exploration and visualization tasks.