**Covid-19 Vaccines Analysis**

**Phase 4: Development Part 2**

**Topic:** Continue Building the project Covid-19 Vaccine Analysis by Performing exploratory data analysis, Statistical analysis and Visualization



**Overview of the process:**

The following is the overview of the Covid-19 Vaccine Analysis project by performing exploratory data analysis, Statistical analysis and Visualization. Analyzing COVID-19 vaccine data through exploratory data analysis (EDA), statistical analysis, and visualization is a critical aspect of understanding the vaccine's effectiveness and distribution.

**1.Data Collection:**

The first step in this project is to gather the relevant data. This data can include information on vaccine distribution, administration, vaccine types (e.g., Pfizer, Moderna, Johnson & Johnson), demographics of vaccinated individuals, adverse events, and more. Sources of data can include government health agencies, research institutions, and vaccination clinics.’

**2. Data Cleaning and Preprocessing:**

Raw data often contains errors or missing values. Data cleaning involves handling missing data, removing duplicates, and converting data into a format suitable for analysis.

**3. Exploratory Data Analysis (EDA)**:

* **Descriptive Statistics**: Compute basic statistics like mean, median, standard deviation, and percentiles to summarize the data.
* **Data Visualization**: Create visualizations such as bar charts, histograms, and box plots to understand the distribution of key variables. For instance, you might visualize the age distribution of vaccine recipients or the regional distribution of vaccinations.
* **Correlation Analysis**: Identify relationships between variables. For example, you can analyze the correlation between vaccine coverage and COVID-19 case rates

.

* **Time Series Analysis**: Examine how vaccine distribution and COVID-19 cases have changed over time.

**4. Statistical Analysis:**

Hypothesis Testing: Formulate hypotheses and conduct statistical tests to determine if there are significant differences or associations.

For example, you can test whether there's a significant difference in vaccine efficacy between age groups.

**5. Data Visualization:**

Create more complex visualizations to present key insights. For instance, you can use choropleth maps to show vaccination rates by region, or you can create time series plots to visualize vaccine rollout progress.

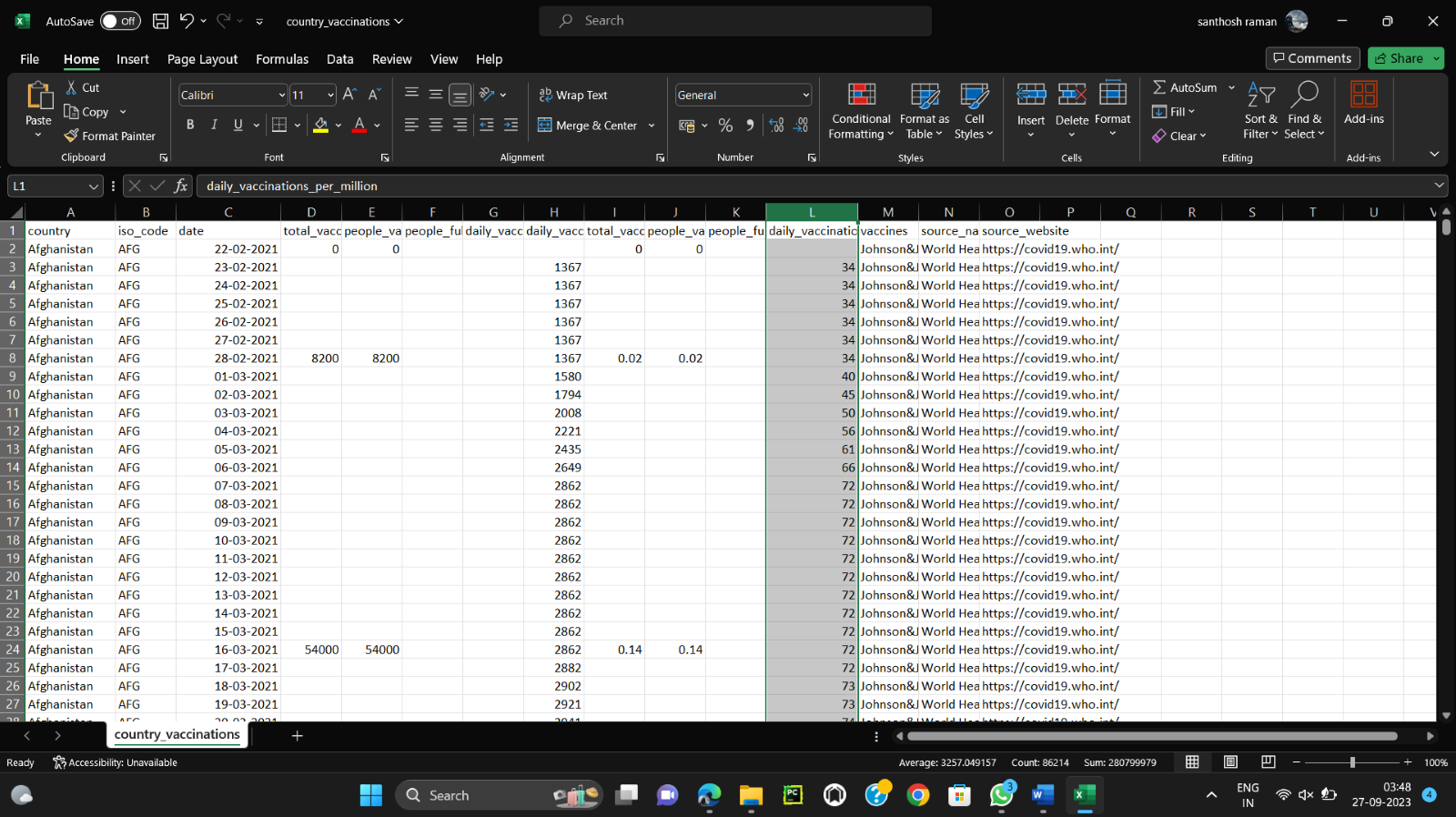
**Exploratory Data Analysis (EDA):**

**I**t is an approach to analyzing and visualizing data sets to summarize their main characteristics, often with the help of statistical graphics and various data visualization techniques. Here's a general outline of the steps typically involved in EDA along with their source code and output:

**1.Data Collection**:

In this stage we simply collect the dataset that we are in need to Analyse. This can be done by making using of the dataset link that is given below:

“<https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress> "



**2.Data Loading:**

To load a dataset in Python, we make use of various libraries, such as Pandas, NumPy, and scikit-learn, depending on the dataset's format and our specific requirements. The term "dataset" is quite broad, so the method that we use may vary depending on whether we have a CSV file, Excel file, SQL database, or some other format. Since we are given a csv file, we make use of the read\_csv() method to read the given dataset file.

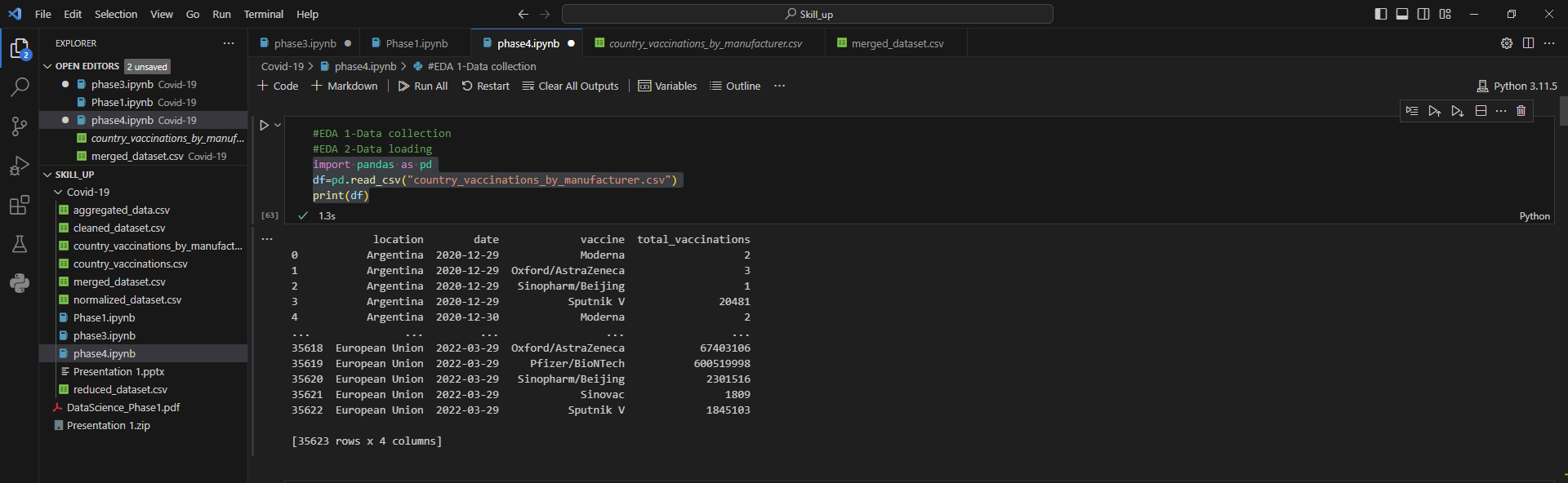
***Program:***

import pandas as pd

df=pd.read\_csv("country\_vaccinations\_by\_manufacturer.csv")

print(df)

***Output:***



**3.Data Inspection:**  
 Data inspection is an important step in the Exploratory Data Analysis (EDA) process. It involves examining and understanding the raw data we are working with before diving into more complex analysis. This can be started by inspecting the dataset to understand its basic characteristics. To do it so we make use of commands like head(), info(), and describe() in pandas to check the first few rows, data types, and summary statistics.

***Program:***

import pandas as pd

data = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

print("First few rows of the dataset:")

print(data.head())

print("\Information about the dataset:")

print(data.info ())

print("\nSummary statistics:")

print(data.describe())

print("\nMissing values:")

print(data.isnull().sum())

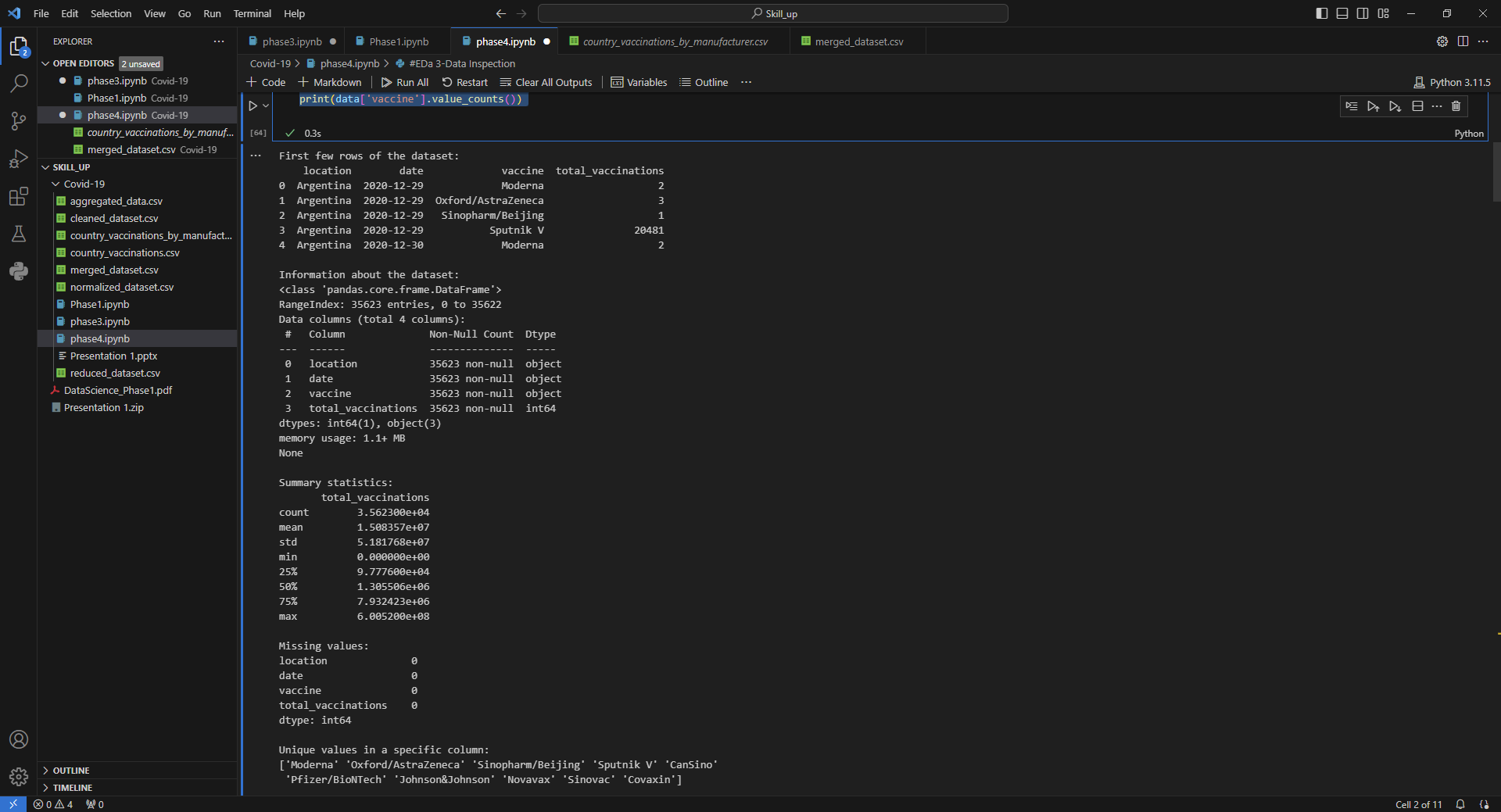
print("\nUnique values in a specific column:")

print(data['vaccine'].unique())

print("\nCount of unique values in a specific column:")

print(data['vaccine'].value\_counts())

***Output:***



**4. Handling Missing Data:**

Identify and address missing data. Use methods like dropping rows with missing values or imputing missing values with appropriate strategies.

Check for missing or null values in the dataset. Missing data can impact your analysis, and you may need to decide how to handle it.

Functions like isnull() or isna() in Pandas can be used to identify missing values.

Handling missing data is a critical aspect of data analysis and is often necessary during the data preprocessing phase of an analysis project.

Missing data can arise for various reasons, including errors in data collection, data entry issues, or non-responses in surveys.

***Program:***

import pandas as pd

data = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

print("Missing values before handling:")

print(data.isnull().sum())

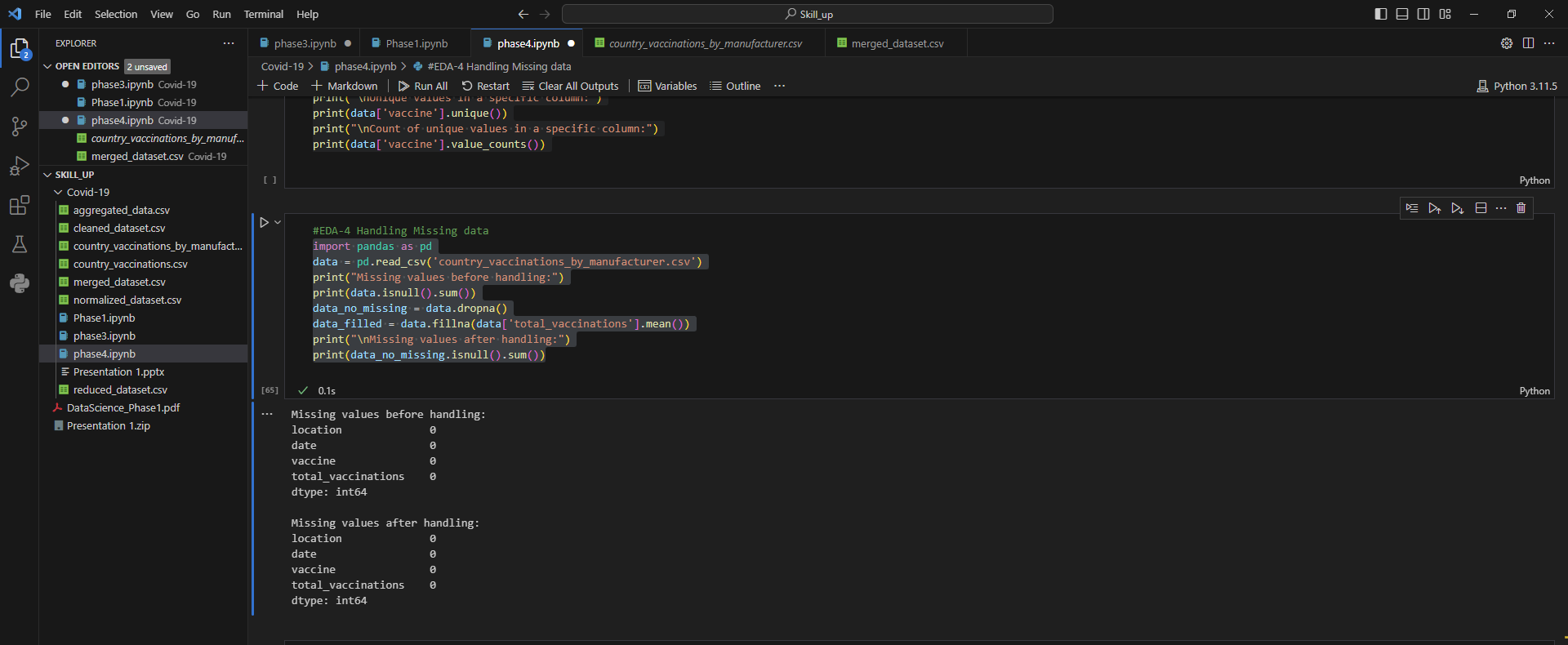
data\_no\_missing = data.dropna()

data\_filled = data.fillna(data['total\_vaccinations'].mean())

print("\nMissing values after handling:")

print(data\_no\_missing.isnull().sum())

***Output:***



**5. Handling Outliers**:

Detect and deal with outliers, which are extreme values that can affect your analysis. Visualizations like box plots or scatter plots can help in identifying outliers. Handling outliers is an important step in data analysis to ensure that extreme or erroneous data points do not unduly influence the results of statistical analyses or data visualizations. Outliers can be caused by data entry errors, measurement errors, or rare but legitimate observations

***Program:***

import pandas as pd

import numpy as np

data = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

def handle\_outliers(data, column\_name):

mean = data[column\_name].mean()

std = data[column\_name].std()

z\_score\_threshold = 3

z\_scores = (data[column\_name] - mean) / std

is\_outlier = np.abs(z\_scores) > z\_score\_threshold

data\_no\_outliers = data[~is\_outlier]

data\_no\_outliers = data. Copy()

data\_no\_outliers.loc[is\_outlier, column\_name] = data[column\_name].median()

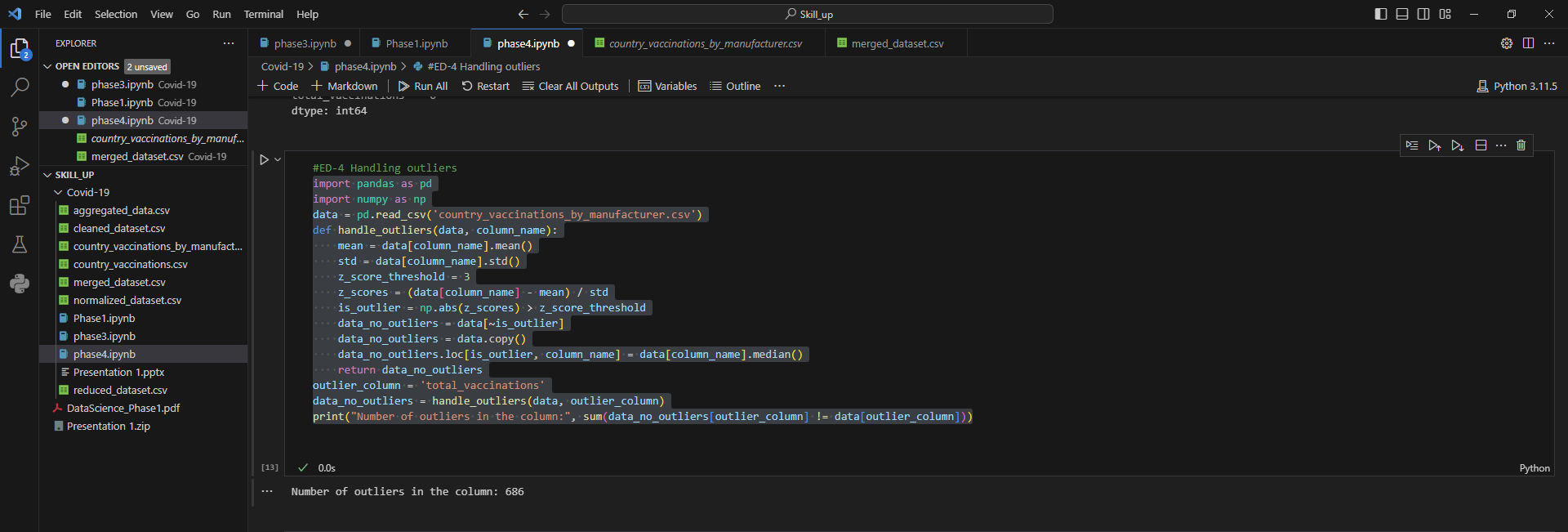
return data\_no\_outliers

outlier\_column = 'total\_vaccinations'

data\_no\_outliers = handle\_outliers(data, outlier\_column)

print("Number of outliers in the column:", sum(data\_no\_outliers[outlier\_column] != data[outlier\_column]))

***Output:***



**6.Data Transformation**:   
 Data transformation is a fundamental step in data preprocessing and analysis. It involves modifying the original data in various ways to make it more suitable for analysis, modeling, and visualization. Data transformation can help address issues like skewness, non-linearity, and heteroscedasticity, among others.

***Program:***

import pandas as pd

df = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

print(df.dtypes)

df['total\_vaccinations'] = df['total\_vaccinations'].astype(object)

print(df.dtypes)

df = pd.get\_dummies(df, columns=['total\_vaccinations'])

print(df)

import pandas as pd

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

columns\_to\_normalize = ['total\_vaccinations']

scaler = StandardScaler()

df[columns\_to\_normalize] = scaler.fit\_transform(df[columns\_to\_normalize])

df.to\_csv('normalized\_dataset.csv', index=False)

print("Normalized dataset:")

print(df.head())

import pandas as pd

df = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

grouped\_data = df.groupby('date')

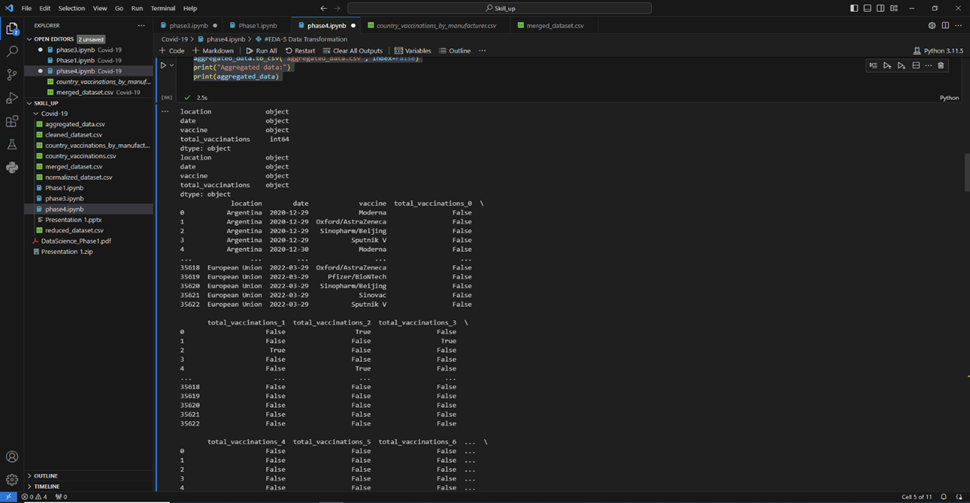
aggregated\_data = grouped\_data['total\_vaccinations'].mean().reset\_index()

aggregated\_data.to\_csv('aggregated\_data.csv', index=False)

print("Aggregated data:")

print(aggregated\_data)

***Output:***

******

**8.Univariant Analysis:**

Explore individual variables using summary statistics and visualizations. Create histograms, box plots, or bar charts to understand the distribution of each variable. Univariate analysis is a statistical and data analysis technique that focuses on analyzing and summarizing the characteristics of a single variable in isolation. In other words, it examines one variable at a time to understand its distribution, central tendency, dispersion, and other key characteristics. Univariate analysis is often the first step in exploratory data analysis (EDA) and provides fundamental insights into the data.

***Program:***

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("country\_vaccinations\_by\_manufacturer.csv")

column\_name = "total\_vaccinations"

summary = df[column\_name].describe()

print(f"Summary statistics for {column\_name}:\n{summary}\n")

plt.hist(df[column\_name], bins=20, edgecolor='k')

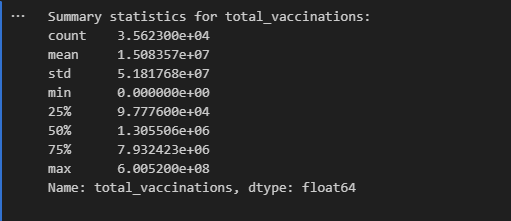
plt.xlabel(column\_name)

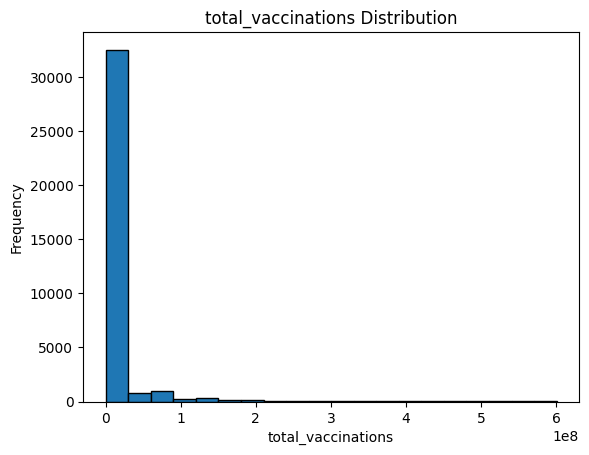
plt.ylabel('Frequency')

plt.title(f'{column\_name} Distribution')

plt.show()

***Output:***





***Basic Exploratory Analysis:***  

Basic exploratory data analysis (EDA) is an essential initial step in data analysis that involves summarizing, visualizing, and understanding the main characteristics of your dataset.

exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

***Program:***

import pandas as pd

import matplotlib.pyplot as plt

df=pd.read\_csv("country\_vaccinations.csv",encoding="unicode\_escape")

print(df.head())

print(df.describe())

print(df['people\_vaccinated'].value\_counts())

df.hist()

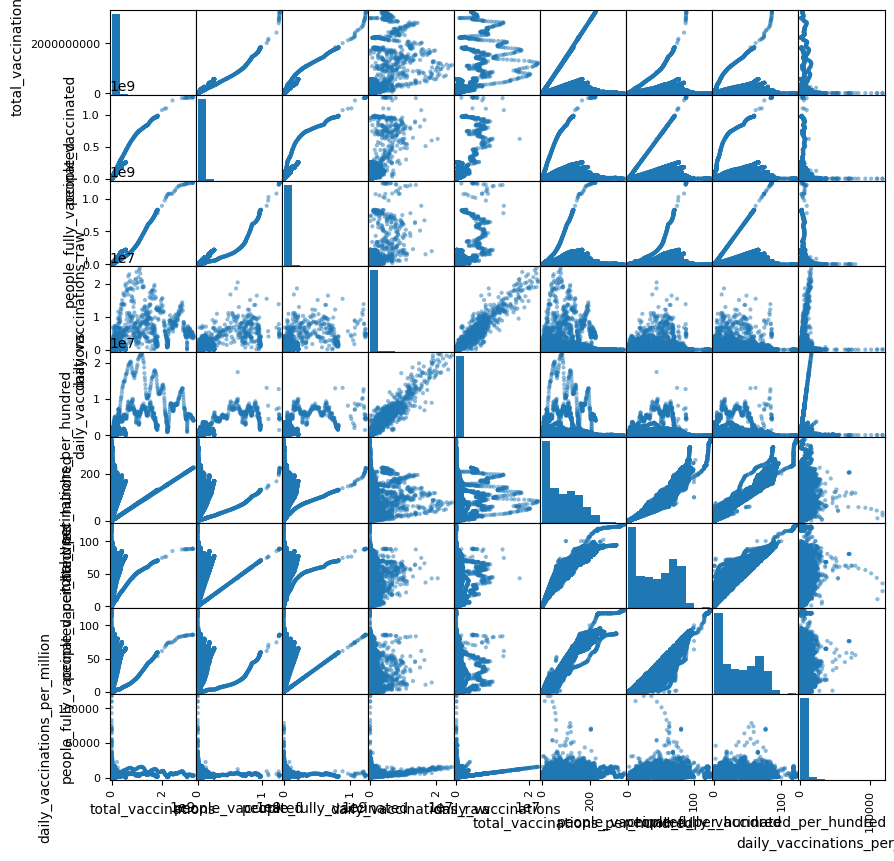
plt.show()

from pandas.plotting import scatter\_matrix

scatter\_matrix(df, figsize=(10, 10))

plt.show()

***Output:***



**Statistical Analysis:**

      Statistical analysis is a systematic approach to understanding data through the application of mathematical and statistical techniques. It plays a crucial role in making sense of complex information, identifying patterns, and drawing meaningful insights. The process typically begins with data collection, followed by data cleaning and preprocessing to ensure data quality. Descriptive statistics are employed to provide an initial summary of the dataset, revealing central tendencies and variability. Inferential statistics, on the other hand, are used to make predictions and test hypotheses about the population from which the data was collected. This branch of analysis encompasses a wide array of methods, including hypothesis testing, regression analysis, and analysis of variance, among others. Statistical analysis is a cornerstone in fields ranging from science and business to healthcare and social sciences, aiding in decision-making, problem-solving, and evidence-based reasoning.

***Program:***

 import pandas as pd

import numpy as np

from scipy import stats

df = pd.read\_csv("merged\_dataset.csv")

df=df.tail(10)

summary = df.describe()

group1\_data = df['total\_vaccinations\_x']

group2\_data = df['people\_vaccinated']

t\_statistic, p\_value = stats.ttest\_ind(group1\_data, group2\_data)

correlation\_coefficient = df['people\_fully\_vaccinated'].corr(df['daily\_vaccinations\_raw'])

print("Descriptive Statistics:")

print(summary)

print("\nT-Test Results:")

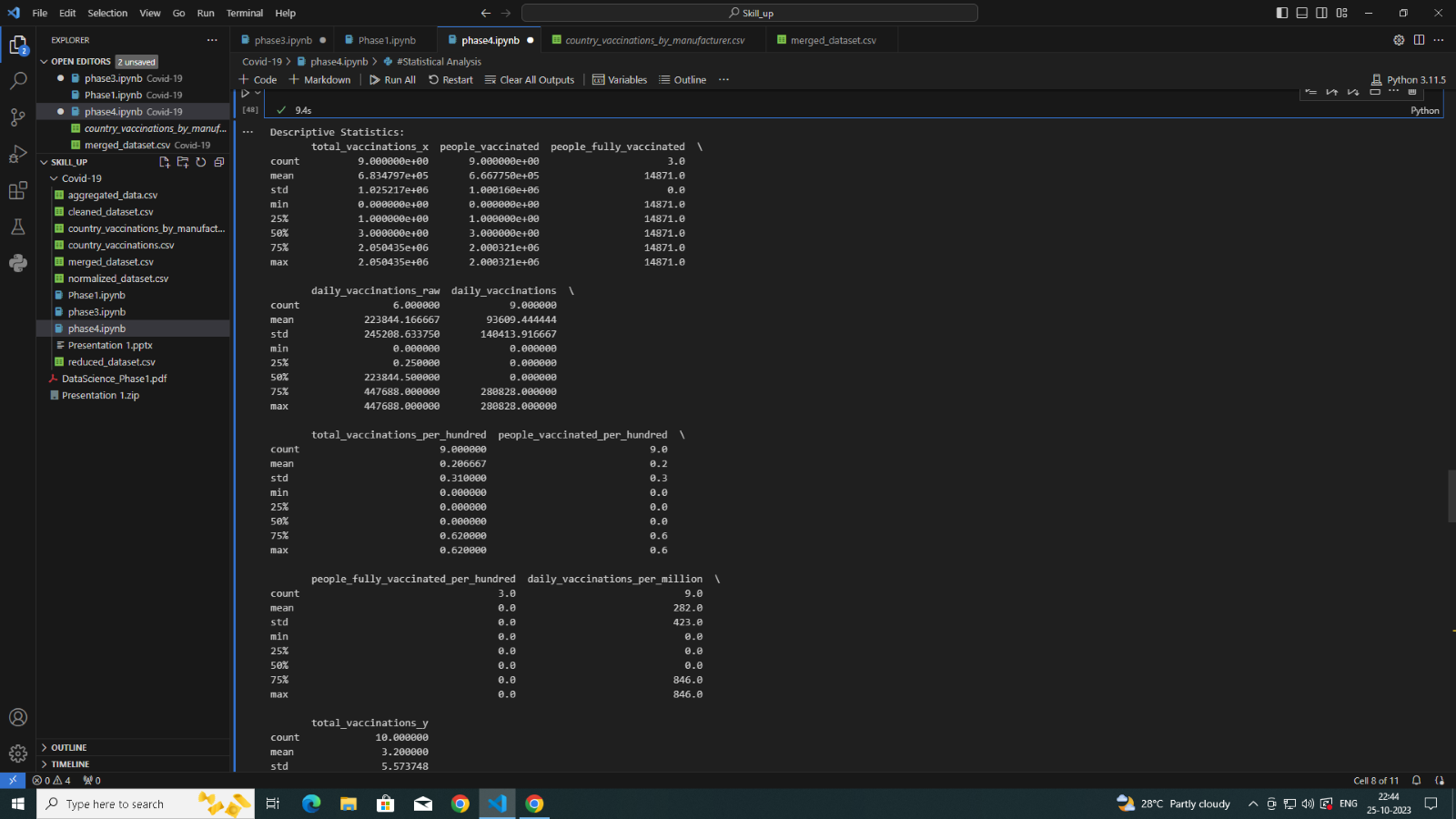
print(f"T-Statistic: {t\_statistic}")

print(f"P-Value: {p\_value}")

print("\nPearson Correlation Coefficient:")

print(correlation\_coefficient)

***Output:***



***Data Visualization:***

Data visualization is a powerful method of representing complex data in a visual and understandable format. It involves the creation of charts, graphs, and diagrams to illustrate patterns, trends, and relationships within the data. By presenting information visually, data visualization makes it easier for individuals to comprehend and interpret large datasets. Data visualizations can take many forms, from simple bar charts and pie graphs to intricate heat maps and interactive dashboards. They are particularly useful for identifying outliers, correlations, and data distributions, making data more accessible and actionable. In an era of data abundance, data visualization has become an indispensable tool for turning data into knowledge and communicating findings effectively to both experts and non-experts.

***Program:***

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("country\_vaccinations.csv")

df=df.tail(50)

categories = df["date"]

values = df["total\_vaccinations"]

plt.figure(figsize=(10, 6))

plt.bar(categories, values)

plt.xlabel("Categories")

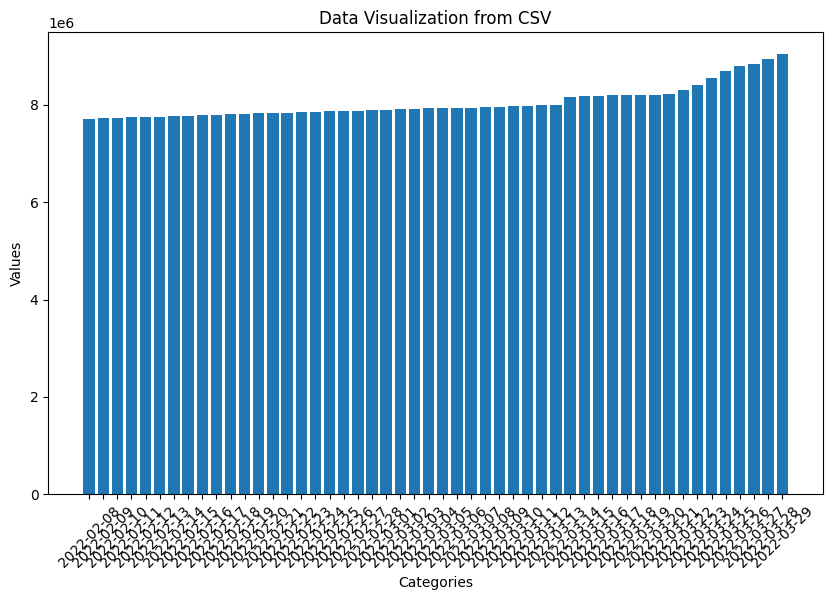
plt.ylabel("Values")

plt.title("Data Visualization from CSV")

plt.xticks(rotation=45)

plt.show()

***Output:***



***Program2:***

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

from sklearn.datasets import load\_iris

df = pd.read\_csv("country\_vaccinations\_by\_manufacturer.csv")

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

scatter = ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y, cmap=plt.cm.Set1)

ax.set\_xlabel('total\_vaccinations')

ax.set\_ylabel('people\_vaccinated')

ax.set\_zlabel('people\_vaccinated\_per\_100')

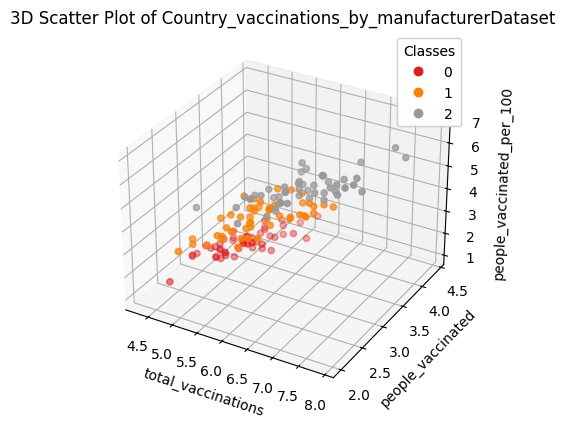
legend = ax.legend(\*scatter.legend\_elements(), title="Classes")

ax.add\_artist(legend)

ax.set\_title('3D Scatter Plot of Country\_vaccinations\_by\_manufacturerDataset')

plt.show()

***Output:***



***Program 3:***

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib as plt

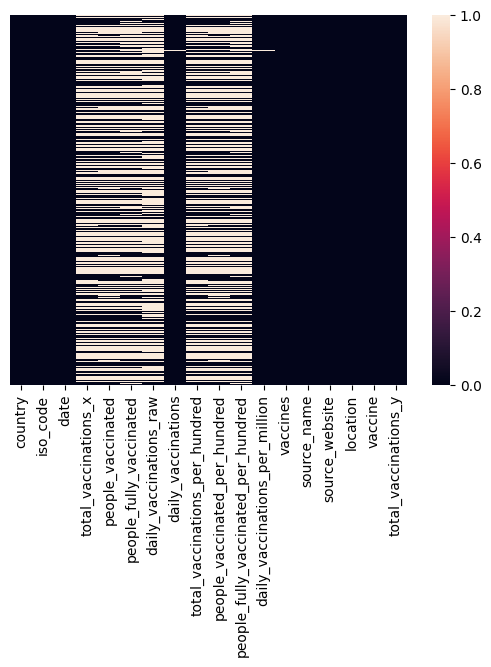
df=pd.read\_csv("merged\_dataset.csv",encoding="unicode\_escape")

missing\_values=["N/a","na",np.nan]

df=pd.read\_csv("merged\_dataset.csv",na\_values=missing\_values,encoding="unicode\_escape")

print(sns.heatmap(df.isnull(),yticklabels=False))

***Output:***



**Conclusion:**

                  In conclusion, the comprehensive analysis of COVID-19 vaccines through a combination of exploratory data analysis (EDA), statistical analysis, and visualization has proven instrumental in understanding the pandemic's trajectory and the impact of vaccination efforts. EDA allowed for the initial assessment of the vaccine's distribution, demographics of recipients, and adverse events, shedding light on important patterns and disparities. Statistical analysis enabled hypothesis testing, identification of vaccine efficacy, and the exploration of associations, offering valuable insights for public health decision-makers. Furthermore, data visualization played a pivotal role in making complex information accessible and aiding in the communication of critical findings to the public. The holistic approach of these analytical methods has not only contributed to informed decision-making but has also underscored the importance of data-driven strategies in addressing global health crises. As we continue to navigate the challenges posed by COVID-19, the integration of these analytical techniques remains essential in monitoring, evaluating, and optimizing vaccination campaigns, ultimately striving for a safer and healthier world.