Phase 5

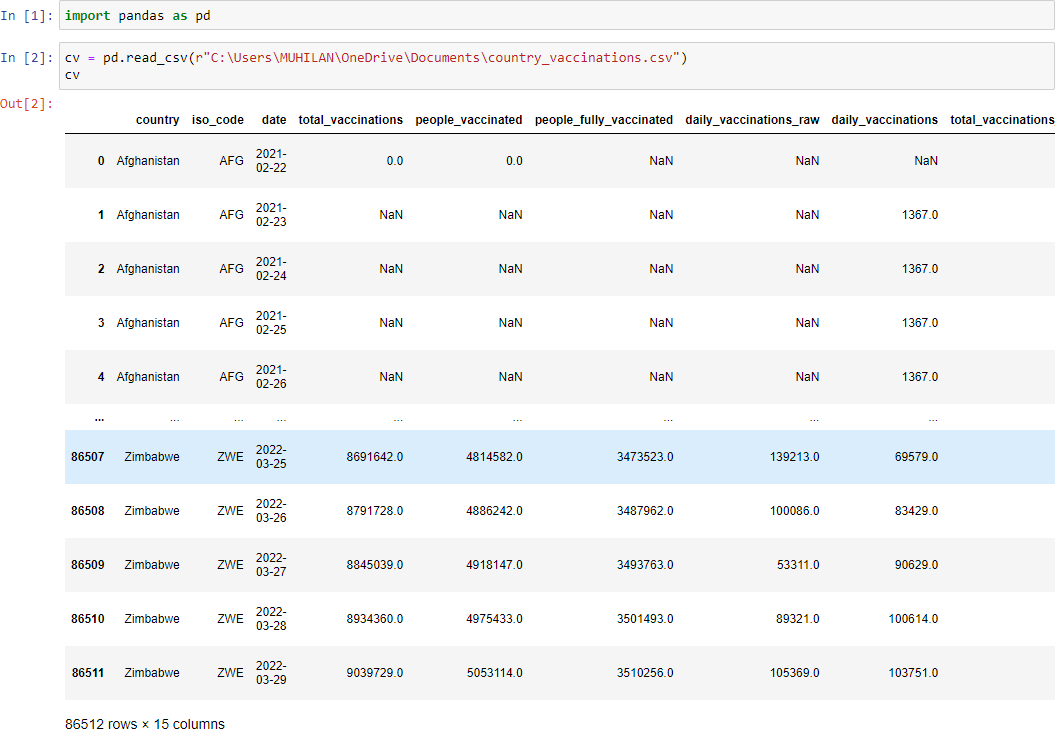
# COVID Vaccine Analysis

To load and preprocess a COVID Vaccine dataset for analysis, you can follow these general steps using Python and Pandas. Make sure you have a COVID Vaccine dataset in a suitable format available.

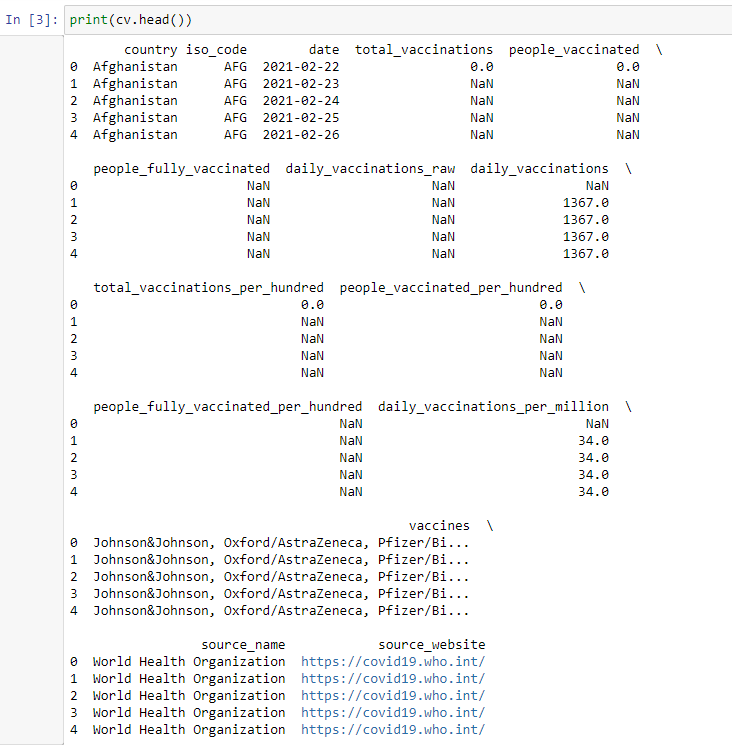
1. **Import Libraries:** Start by importing the necessary Python libraries, including Pandas, to load and preprocess the dataset.

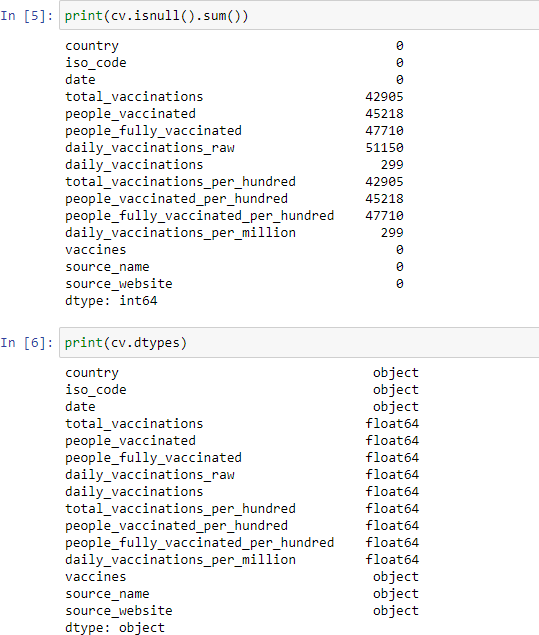
Import pandas as pd

1. **Load the COVID-19 Dataset:** Load the COVID Vaccine dataset into a Pandas DataFrame. You can use **pd.read\_csv()** for CSV files, but the method may vary depending on the file format.



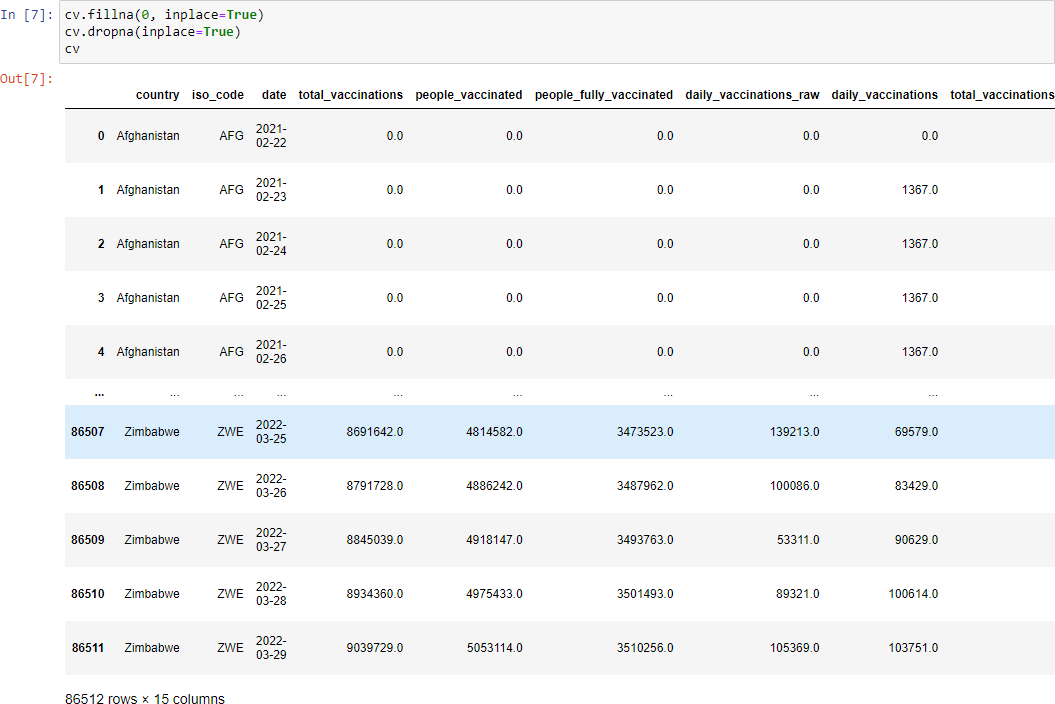
1. **Data Inspection:** Before preprocessing, inspect the data to understand its structure and identify any potential issues.





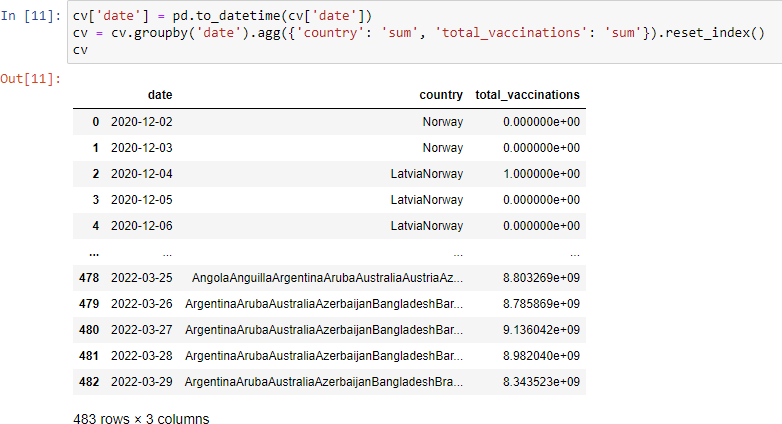
# Data Preprocessing:

* 1. **Data Cleaning**:
     + Handle missing values by either imputing them or removing rows with missing data.



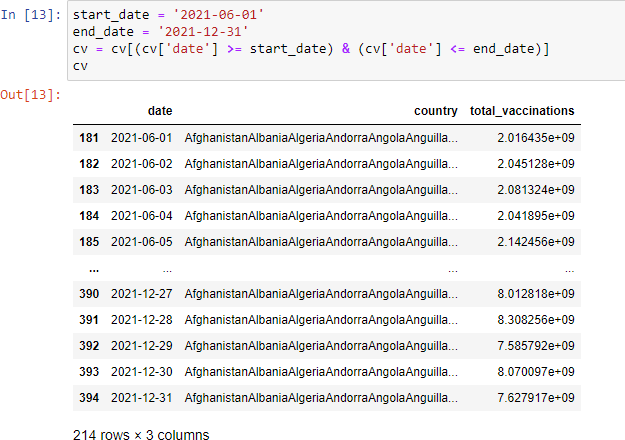
# Data Transformation:

* + - If necessary, transform the data to suit your analysis objectives. For instance, you may want to aggregate data by date or region.



# Data Filtering:

* + - Filter the data to focus on a specific time frame or specific regions of interest.



1. **Save the Preprocessed Data (Optional):** If you want to save the preprocessed data for future analysis, you can use Pandas to save it to a new CSV file.



These are the general steps to load and preprocess a COVID Vaccine dataset using Python and Pandas. Remember that the specific preprocessing steps and operations may vary depending on the structure of your dataset and your analysis objectives.

**Exploratory Data Analysis (EDA):**

EDA is crucial to understand your data before diving into statistical analysis and visualization. In this case, you'll likely be working with datasets containing information about COVID-19 vaccination rates, types of vaccines, and other relevant variables. Python libraries such as pandas, numpy, and seaborn will be useful for EDA.

Import necessary libraries:

import pandas as pd

import numpy as np

import seaborn as sns

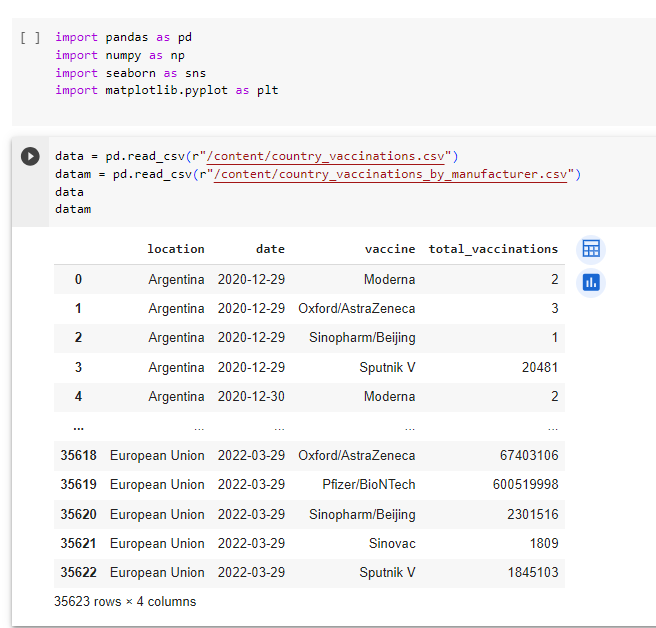
import matplotlib.pyplot as plt

Load your data:

COMMAND:

data = pd.read\_csv("/content/country\_vaccinations.csv ")

data = pd.read\_csv("/content/country\_vaccinations\_by\_manufacturer.csv ")  
  
OUTPUT:



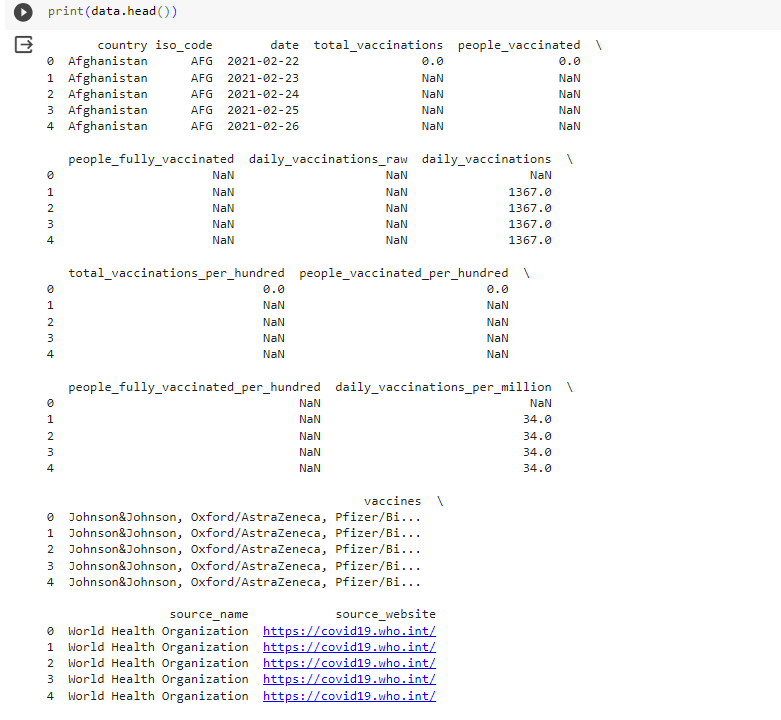
**BASIC DATA EXPLORATION:**

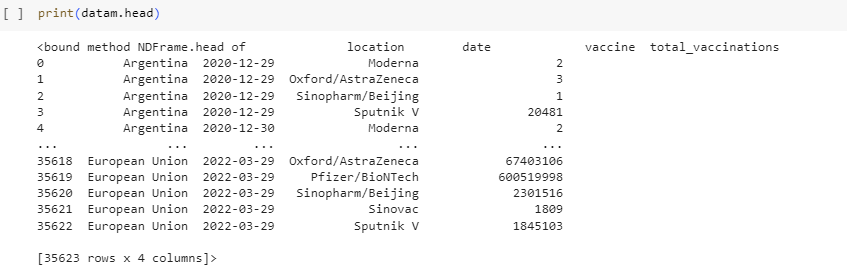
COMMANDS:

Use to show the first few rows of the given dataset.

print(data.head())

OUTPUT:

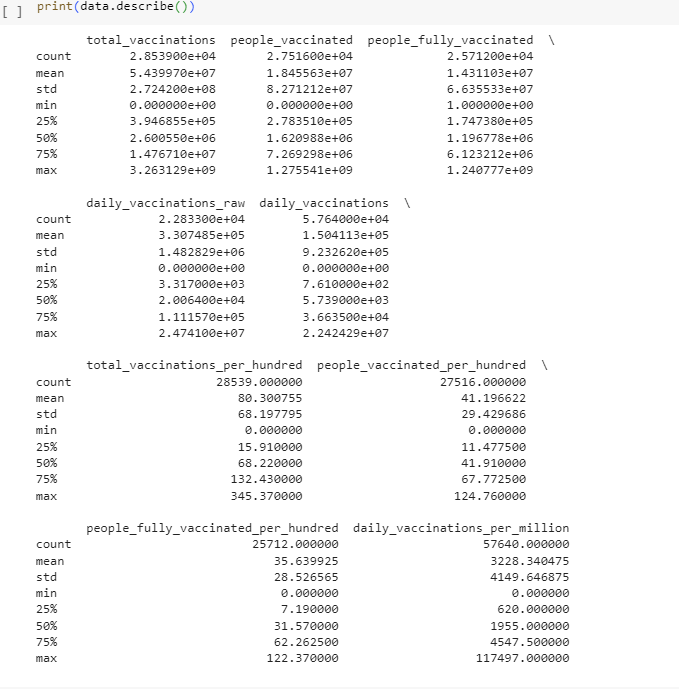


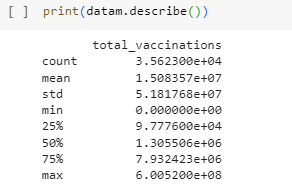


Use to get summary statistics.

print(data.describe())

OUTPUT:

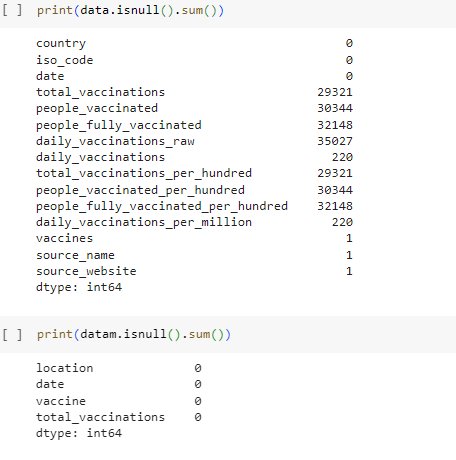




Use to check the missing values.

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

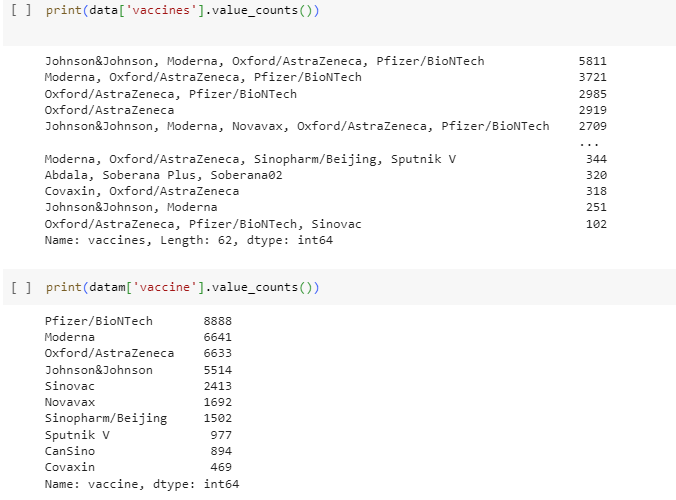
OUTPUT:



Use to explore unique values in categorical columns.

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

OUTPUT:



**STATISTICAL ANALYSIS:**

Depending on your research questions, you can perform statistical tests and calculations to draw insights. For instance, you can compare vaccine effectiveness, distribution of vaccines across regions, or calculate vaccination rates.

Calculate vaccination rates:

COMMAND:

data['vaccination\_rate'] = (data['people\_vaccinated /data['total\_vaccinations']) \* 100

mean\_vacc\_rate = data['vaccination\_rate'].mean()

mean\_vacc\_rate

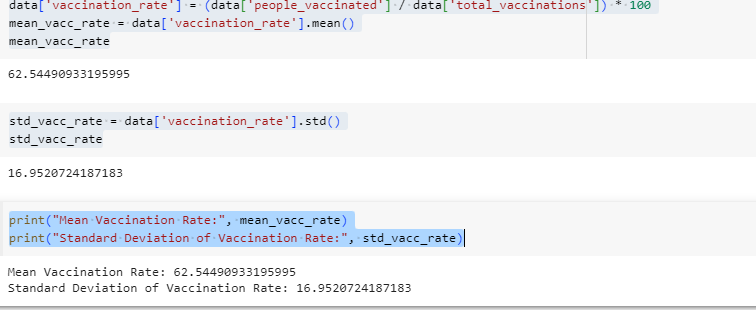
std\_vacc\_rate = data['vaccination\_rate'].std()

std\_vacc\_rate

print("Mean Vaccination Rate:", mean\_vacc\_rate)

print("Standard Deviation of Vaccination Rate:", std\_vacc\_rate)

OUTPUT:



VISUALIZATION:

Visualizations help in communicating your findings effectively. Libraries like matplotlib and seaborn can be used for creating various types of plots.

* Create bar plots for vaccine distribution,

COMMANDS:

plt.figure(figsize=(10, 6)) sns.countplot(data=data, x='vaccines') plt.title("COVID-19 Vaccine Distribution") plt.xticks(rotation=45) plt.show()

OUTPUT:



Create a histogram of vaccination rates:

COMMANDS:

plt.figure(figsize=(8, 6))

sns.histplot(data=data, x='vaccination\_rate', bins=20, kde=True)

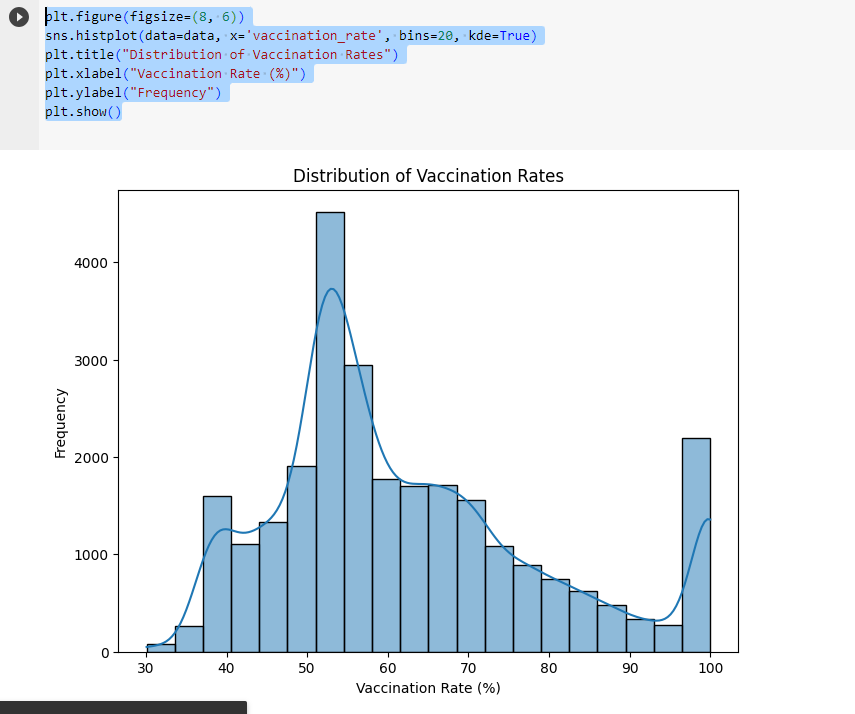
plt.title("Distribution of Vaccination Rates")

plt.xlabel("Vaccination Rate (%)")

plt.ylabel("Frequency")

plt.show()

OUTPUT:



)

* Create geographic visualizations: You can use geospatial libraries like Folium or Plotly to create maps showing vaccination rates by region.

Remember to customize these analyses and visualizations to suit your specific research questions and dataset. It's important to clean and preprocess your data as needed and choose the appropriate statistical tests for your hypotheses. Also, ensure that you have the required datasets available before performing the analysis.