**COVID-19 VACCINE ANALYSIS**

**PHASE 4**: Development part 2

**PHASE DESCRIPTION**: Conducting the COVID-19 vaccines analysis by performing exploratory data analysis,statistical analysis and visualization.

**Load the dataset:**

* Data sets are available in .csv format. A CSV file stores tabular data in plain text.
* Each line of the file is a data record. We use the read\_csv method of the pandas library to read a local CSV file as a dataframe.
* Load our customer data from the CSV file

**Code:**

import pandas as pd

# Try reading the file with different encodings

encodings = ['utf-8', 'latin1', 'ISO-8859-1']

for encoding in encodings:

try:

dataset = pd.read\_csv(r'C:\\Users\\KISHORE\\OneDrive\\Documents\\country\_vaccinations.csv', encoding=encoding)

print(f"Successfully read with encoding: {encoding}")

break  # If successful, no need to try other encodings

except UnicodeDecodeError:

print(f"Failed to read with encoding: {encoding}")

# Now 'dataset' should contain your data

**Output:**

****

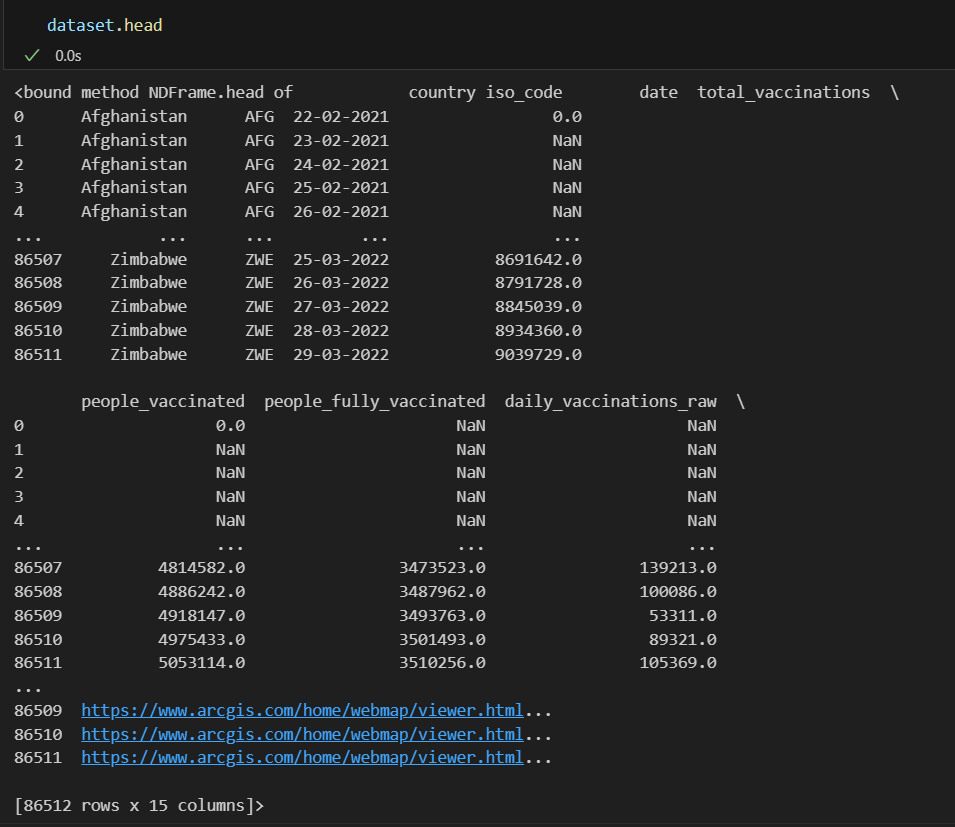
**Head() Function:**

* The head() function is used to get the first n rows.
* This function returns the first n rows for the object based on position.
* It is useful for quickly testing if your object has the right type of data in it.
* If the value of the n is not assigned it returns a default value of first 5 rows

**Code:**

dataset.head()

**Output:**



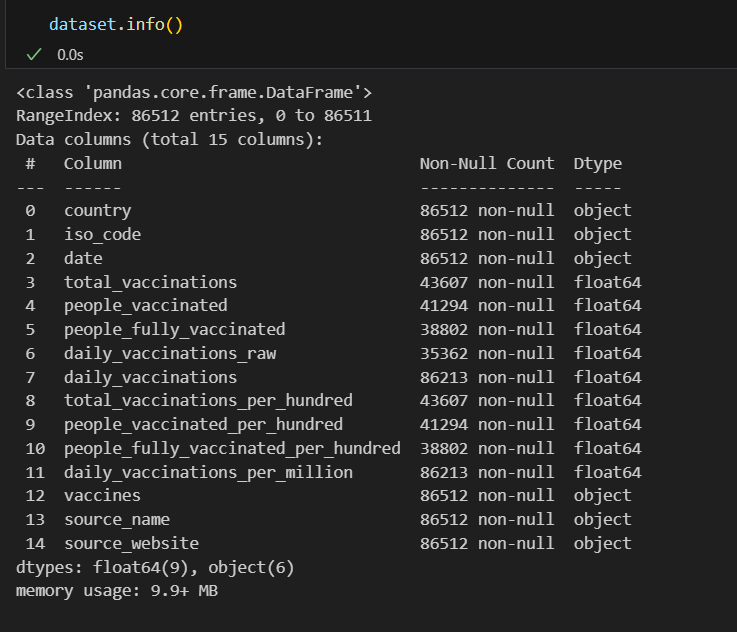
**Info() Function:**

* The info() method prints information about the DataFrame.
* The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-nullvalues).

**Code:**

dataset.info()

**Output:**



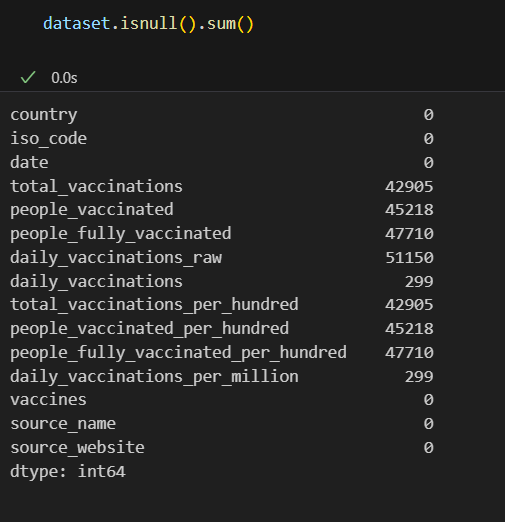
**Df.isnull().sum() Function:**

* This code is used to count the number of missing (null) values in each column of a DataFrame, denoted as df.
* It returns a summary of the missing data for each column, showing how many missing values are there in each column.
* This information is essential in data preprocessing and analysis to identify and handle missing data appropriately.Top of Form

**Code:**

dataset.isnull().sum()

**Output:**



**Describe Function:**

* The describe() function in pandas, a popular Python data analysis library, is used to generate summary statistics of a DataFrame or Series.

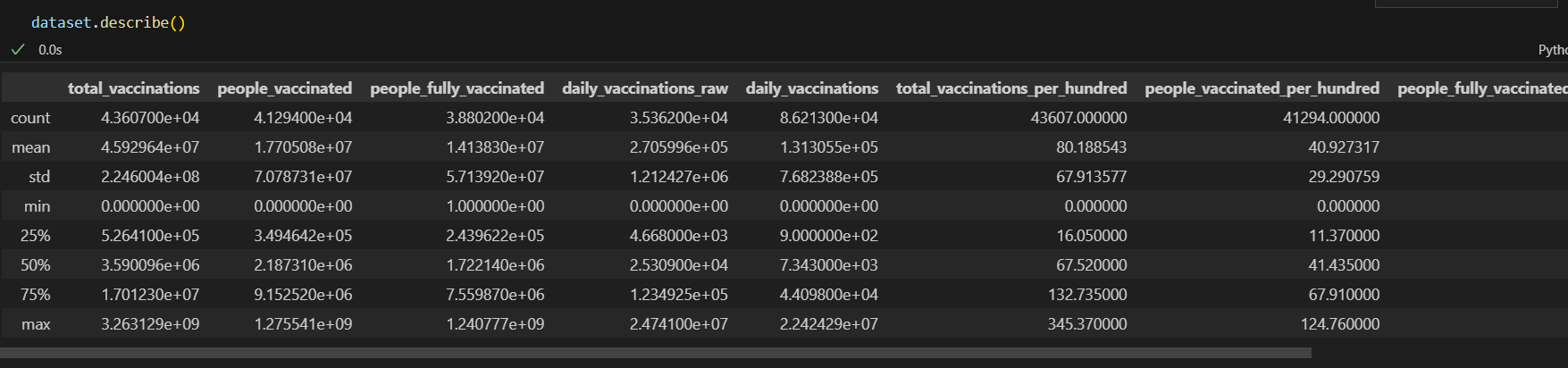
It provides a quick overview of the key statistics for numerical data in the dataset, including:

* **Count:** The number of non-null values.
* **Mean:** The average of the values.
* **Standard Deviation (std):** A measure of the spread or dispersion of the data.
* **Minimum:** The minimum value in the dataset.
* **25th Percentile (25%):** The value below which 25% of the data falls (the first quartile).
* **Median (50% or the 2nd quartile):** The middle value when the data is sorted.
* **75th Percentile (75%):** The value below which 75% of the data falls (the third quartile).
* **Maximum:** The maximum value in the dataset.

**Code:**

dataset.describe()

**Output:**



**WCSS Function:**

* WCSS is the sum of the squared distance between each point and the centroid in a cluster.
* When we plot the WCSS with the K value, the plot looks like an Elbow.
* As the number of clusters increases, the WCSS value will start to decrease.

**Code:**

plt.plot(range(1,11),wcss)

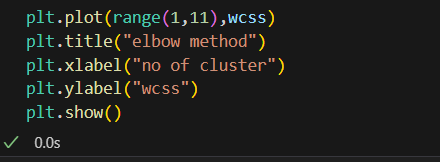
plt.title("elbow method")

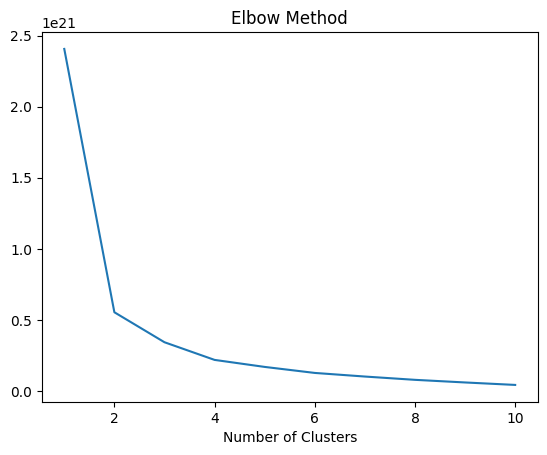
plt.xlabel("no of cluster")

plt.ylabel("wcss")

plt.show()

**Output:**





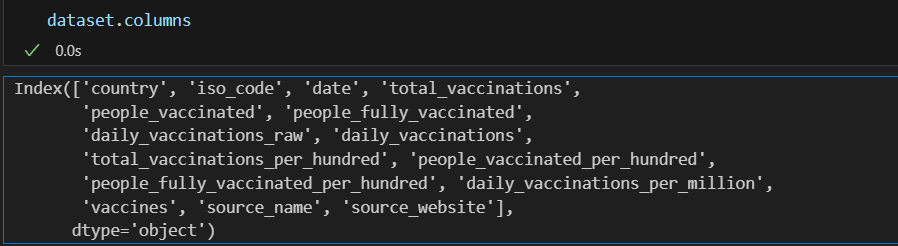
**Dataset.columns :**

* We can use the loc and iloc functions to access columns in a Pandas DataFrame.
* for example the Grades column, we could simply use the loc function and specify the name of the column in order to retrieve it.

**Code:**

dataset.columns

**Output:**



**Memory Function:**

* Pandas**dataframe.memory\_usage()** function return the memory usage of each column in bytes.
* The memory usage can optionally include the contribution of the index and

elements of object dtype. This value is displayed in DataFrame.info by default.

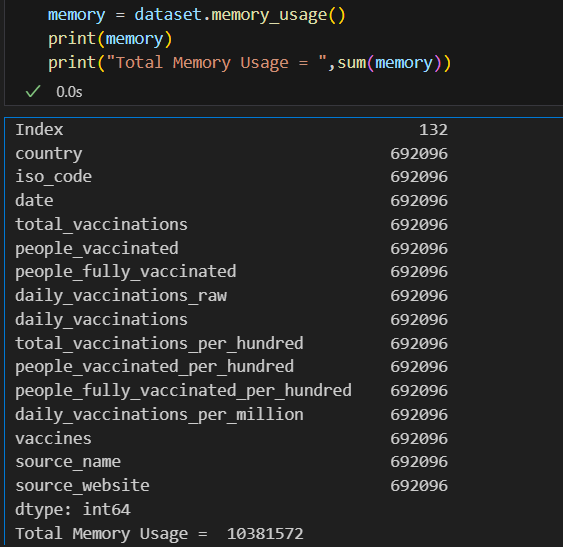
**Code:**

memory = dataset.memory\_usage()

print(memory)

print("Total Memory Usage = ",sum(memory))

**Output:**



**Dropna() Function:**

* dropna() is a function used in data preprocessing, often in the context of data analysis and cleaning, to remove or drop rows or columns with missing (NaN or null) values from a dataset.
* It's a method to eliminate incomplete or unreliable data from your dataset, which can be important to ensure the quality of your analysis or machine learning models

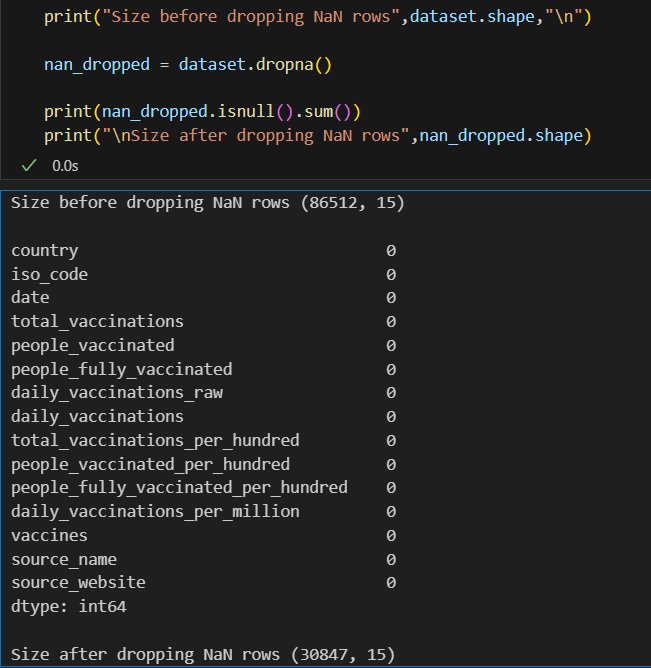
**Code:**

print("Size before dropping NaN rows",dataset.shape,"\n")

nan\_dropped = dataset.dropna()

print(nan\_dropped.isnull().sum())

print("\nSize after dropping NaN rows",nan\_dropped.shape)

**Output:**

**Iloc() Function:**

The iloc() function is a method in pandas, a popular Python library for data manipulation and analysis. It is primarily used to select and access data in a DataFrame by integer-based indexing.

* Select specific rows and columns from a DataFrame using integer-based indexing.
* Provide a way to slice and filter data by row and column positions.

**Code:**

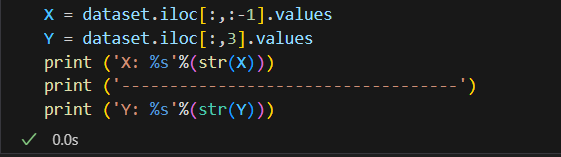
X = dataset.iloc[:,:-1].values

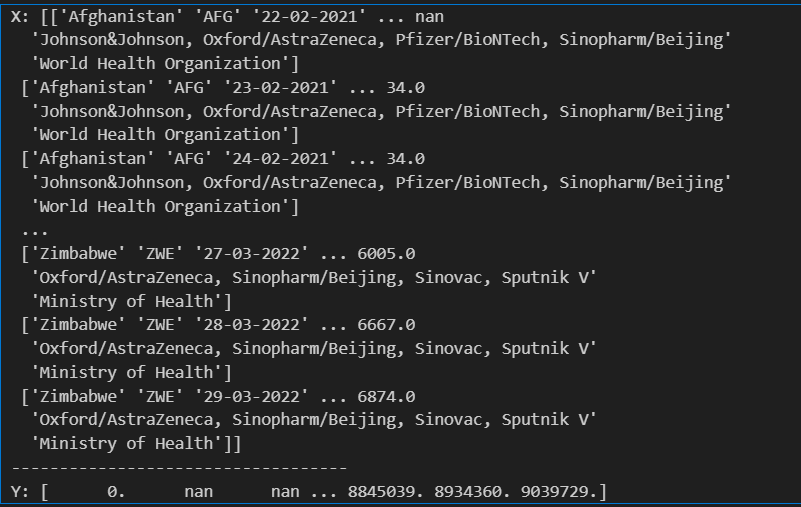
Y = dataset.iloc[:,3].values

print ('X: %s'%(str(X)))

print ('-----------------------------------')

print ('Y: %s'%(str(Y)))

**Output:**



**Subplots Function:**

* Subplots are a feature in data visualization that allow you to create multiple smaller plots within a larger figure.
* They are useful for displaying multiple related visualizations side by side, making it easier to compare and analyze data.
* subplots help you arrange and present multiple charts, graphs, or plots in a single figure, improving the overall clarity and readability of your data visualizations.

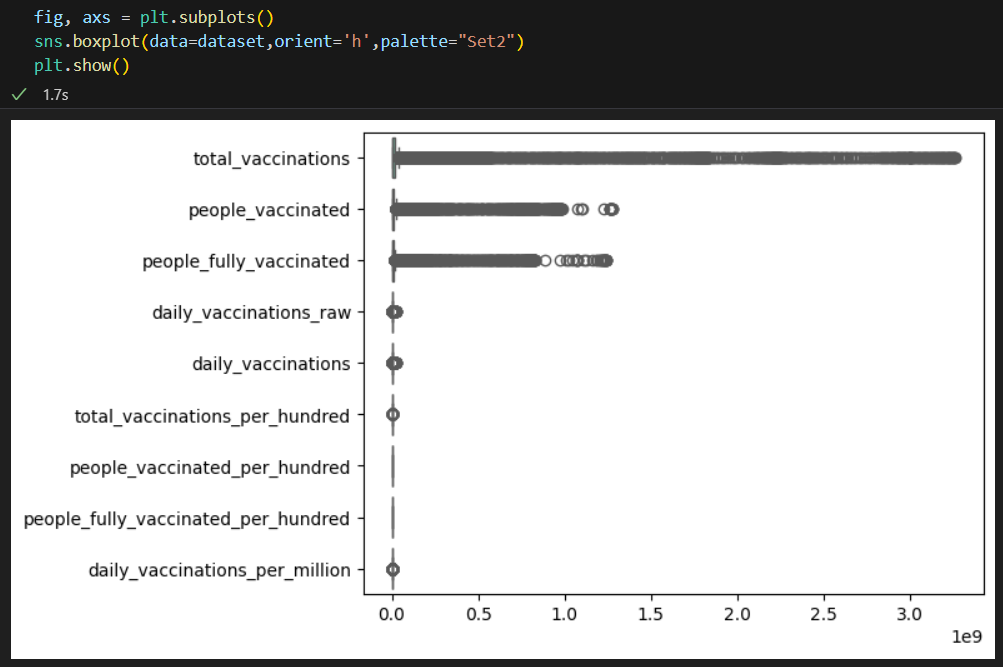
**Code:**

fig, axs = plt.subplots()

sns.boxplot(data=dataset,orient='h',palette="Set2")

plt.show()

**Output:**



**Missingno Function:**

* "missingno" is a Python library used for visualizing and analyzing missing data in a dataset.
* It provides various visualization tools to quickly understand and identify missing values in your data, allowing you to make informed decisions on how to handle or impute missing data.

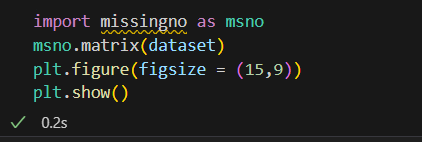
**Code:**

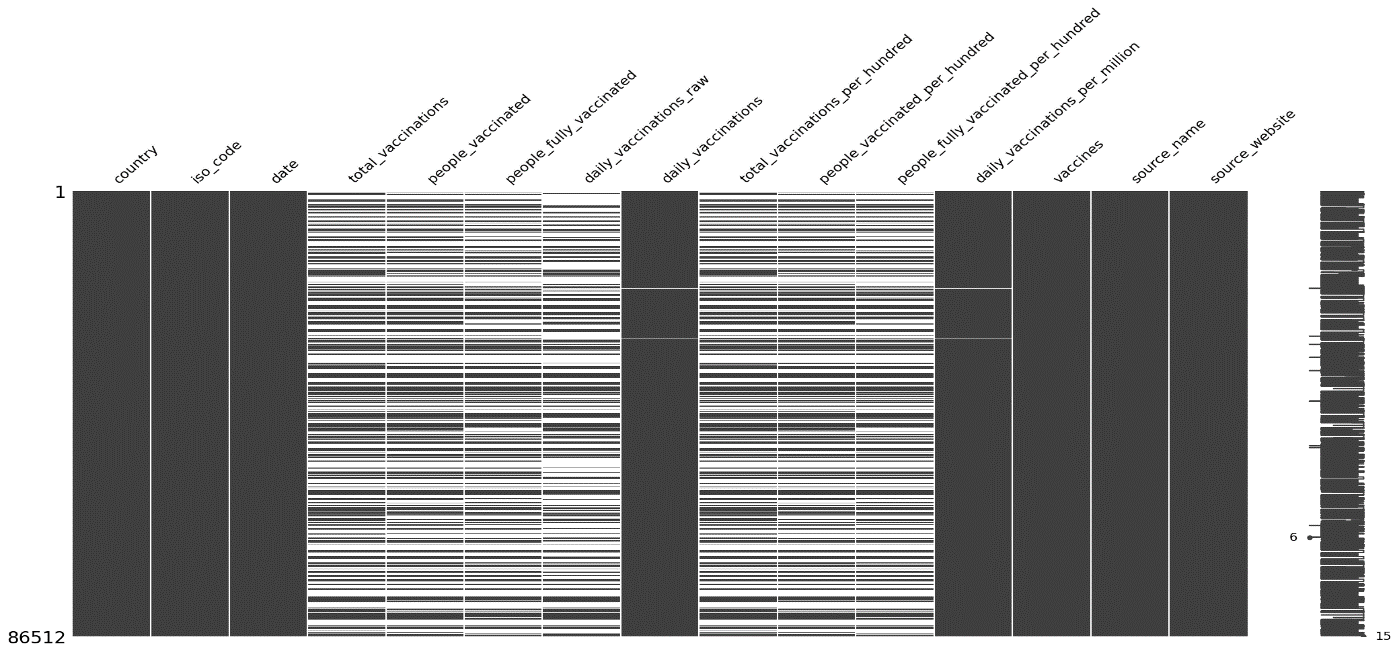
import missingno as msno

msno.matrix(dataset)

plt.figure(figsize = (15,9))

plt.show()

**Output:**

****

**Code:**

msno.heatmap(dataset, labels = True)

**Output:**

