COVID 19 VACCINE

Data Analysis

ABSTRACT

This project is all about carefully looking at information about Covid-19 vaccines. We're mainly interested in how well the vaccines work, how they are distributed, and if there are any negative effects. The big goal is to find useful insights that can help leaders and health groups make better plans for giving out vaccines. To do this, we go through steps like collecting data, cleaning it up, exploring what it tells us, doing some math to understand it better, and making visuals to explain it clearly. The hope is that by doing this, we can give a good picture of how the vaccines are doing and help in the fight against Covid-19.

Objective

The project aims to thoroughly analyze Covid-19 vaccine data with key objectives: evaluating vaccine efficacy, scrutinizing distribution strategies, investigating adverse effects, and providing actionable insights. By achieving these goals, the project seeks to enhance decision-making for policymakers and health organizations, fostering optimized deployment strategies in the ongoing battle against the Covid-19 pandemic.

Design & Thinking

- Data Collection
- Data Preprocessing
- Exploratory Data Analysis(EDA)
- Statistical Analysis
- Virtualization
- Insights and Recommendation

Data collection

- The dataset used for the covid-19 vaccine analysis project is a csv file with 15 columns representing country,ISO code,Date,total vaccination , people vaccinated, people fully vaccinated, daily vaccinated raw, daily vaccination,total vaccinations per 100,people vaccinated per 100, people fully vaccinated per 100, daily vaccinations per million, vaccines, source name,source website.
- Identify reputable sources such as health organizations (e.g., WHO, CDC), government agencies, and trusted research studies to ensure the accuracy and reliability of the data.
- Explore public health databases for comprehensive datasets related to
- Covid-19 vaccines. Prioritize sources that provide detailed and up-to-date
- Establish contact with health departments at various levels (local, regional, national) to obtain detailed and granular data on vaccination campaigns, distribution strategies, and adverse reaction reports.

Data preprocessing

- Implement strategies to handle missing values, such as imputation techniques or, if necessary, consult domain experts to determine appropriate approaches for filling missing data.
- Utilize statistical methods to identify and manage outliers. Decide whether outliers should be corrected, removed, or retained based on their impact on the analysis.
- Ensure consistency in data formats by standardizing units, date formats, and any other variables that may have diverse representations across the dataset.
- Develop procedures to identify and handle any duplicate entries in the dataset, ensuring that each data point is unique and contributes meaningfully of the analysis

Exploratory data analysis

- Identify key variables for exploration, focusing on aspects such as vaccine efficacy rates, distribution patterns, adverse reaction frequencies, and demographic characteristics.
- Generate statistical summaries (mean, median, standard deviation, etc.) for numerical variables and frequency distributions for categorical variables. Complement these summaries with visualizations such as histograms, bar charts, and pie charts for a comprehensive overview.
- Analyze temporal trends in vaccine distribution and adverse reactions. Use time series plots and trend analyses to identify patterns and potential seasonality.
- Use exploratory techniques to identify potential outliers or anomalies in the data.

Statistical analysis

- Use exploratory techniques to identify potential outliers or anomalies in the data. Employ box plots and scatter plots, particularly useful in detecting data points that deviate significantly from the norm
- Formulate clear hypotheses related to vaccine efficacy, distribution, and adverse effects. Define null and alternative hypotheses to guide the statistical analyses.
- Conduct comparative analyses to compare vaccine efficacy rates between different groups (e.g., age groups, regions) using appropriate statistical tests (t-tests, ANOVA, etc.).
- Investigate relationships between variables using correlation analyses. Perform regression analyses to model and predict factors influencing vaccine efficacy, distribution, or adverse effects.
- Calculate descriptive statistics (mean, median, standard deviation, etc.) for key variables.

Visualization

- **Data Exploration Visualization**:

  Line charts, histograms, pie charts for basic trends and distributions.

- **Vaccine Efficacy Visualizations**:

  Bar charts, line charts, heatmaps for variations across vaccine types, age groups, or regions.

- **Distribution Pattern Visualizations**:

  Time series plots, stacked area charts, animated maps for temporal trends and geographic variations.

- **Adverse Effects Representation**:

  Bar charts, donut charts, treemaps for clear representation of adverse reaction profiles.

Insights & Recommendation

- Summarize key findings from statistical analyses, exploratory data analysis (EDA), and visualizations. Highlight significant trends, patterns, and correlations.
- Identify key factors influencing vaccine efficacy, distribution, and adverse effects. Consider demographic factors, regional variations, and temporal trends.
- Understand the implications of the identified patterns and trends. Evaluate how vaccine efficacy, distribution, and adverse effects impact overall public
- Derive insights from comparative analyses, such as differences in vaccine efficacy rates between age groups or regions. Understand the implications of these variations.

Data Source

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

1	country	iso_code	date	total_vaccin	people_vac	people_fully	daily_vaccin	daily_vaccin	total_vaccin	people_vac	people_fully	daily_vaccin	vaccines	source_nam	source_website
2	Afghanistan	AFG	22-02-2021	0	0				0	0			Johnson&Jo	World Healt	https://covid19.who.int/
3	Afghanistan	AFG	23-02-2021					1367				34	Johnson&Jo	World Healt	https://covid19.who.int/
4	Afghanistan	AFG	24-02-2021					1367				34	Johnson&Jo	World Healt	https://covid19.who.int/
5	Afghanistan	AFG	25-02-2021					1367				34	Johnson&Jo	World Healt	https://covid19.who.int/
6	Afghanistan	AFG	26-02-2021					1367				34	Johnson&Jo	World Healt	https://covid19.who.int/
7	Afghanistan	AFG	27-02-2021					1367				34	Johnson&Jo	World Healt	https://covid19.who.int/
8	Afghanistan	AFG	28-02-2021	8200	8200			1367	0.02	0.02		34	Johnson&Jo	World Healt	https://covid19.who.int/
9	Afghanistan	AFG	01-03-2021					1580				40	Johnson&Jo	World Healt	https://covid19.who.int/
10	Afghanistan	AFG	02-03-2021					1794				45	Johnson&Jo	World Healt	https://covid19.who.int/
11	Afghanistan	AFG	03-03-2021					2008				50	Johnson&Jo	World Healt	https://covid19.who.int/
12	Afghanistan	AFG	04-03-2021					2221				56	Johnson&Jo	World Healt	https://covid19.who.int/
13	Afghanistan	AFG	05-03-2021					2435				61	Johnson&Jo	World Healt	https://covid19.who.int/
14	Afghanistan	AFG	06-03-2021					2649				66	Johnson&Jo	World Healt	https://covid19.who.int/
15	Afghanistan	AFG	07-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
16	Afghanistan	AFG	08-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
17	Afghanistan	AFG	09-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
18	Afghanistan	AFG	10-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
19	Afghanistan	AFG	11-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
20	Afghanistan	AFG	12-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
21	Afghanistan	AFG	13-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
22	Afghanistan	AFG	14-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
23	Afghanistan	AFG	15-03-2021					2862				72	Johnson&Jo	World Healt	https://covid19.who.int/
24	Afghanistan	AFG	16-03-2021	54000	54000			2862	0.14	0.14		72	Johnson&Jo	World Healt	https://covid19.who.int/
25	Afghanistan	AFG	17-03-2021					2882				72	Johnson&Jo	World Healt	https://covid19.who.int/
26	Afghanistan	AFG	18-03-2021					2902				73	Johnson&Jo	World Healt	https://covid19.who.int/
27	Afghanistan	AFG	19-03-2021					2921				73	Johnson&Jo	World Healt	https://covid19.who.int/
28	Afghanistan	AFG	20-03-2021					2941				74	Johnson&Jo	World Healt	https://covid19.who.int/
29	Afghanistan	AFG	21-03-2021					2961				74	Johnson&Jo	World Healt	https://covid19.who.int/
30	Afghanistan	AFG	22-03-2021					2980				75	Johnson&Jo	World Healt	https://covid19.who.int/
31	Afghanistan	AFG	23-03-2021					3000				75	Johnson&Jo	World Healt	https://covid19.who.int/
32	Afghanistan	AFG	24-03-2021					3000				75	Johnson&Jo	World Healt	https://covid19.who.int/
33	Afghanistan	AFG	25-03-2021					3000				75	Johnson&Jo	World Healt	https://covid19.who.int/

Data Exploration and Understanding

1. Load the dataset into your preferred data analysis tool, like Python with Pandas or R.
2. Examine the dataset structure and understand the meaning of each column:
    - **country**: Name of the country
    - **iso_code**: ISO country code
    - **date**: Date of the data point
    - **total_vaccinations**: Total number of vaccinations administered
    - **people_vaccinated**: Number of individuals partially vaccinated
    - **people_fully_vaccinated**: Number of individuals fully vaccinated
    - **daily_vaccinations_raw**: Daily increase in total vaccinations
    - **daily_vaccinations**: Daily vaccinations administered
    - **total_vaccinations_per_hundred**: Total vaccinations per 100 people
    - **people_vaccinated_per_hundred**: Partial vaccinations per 100 people
    - **people_fully_vaccinated_per_hundred**: Full vaccinations per 100 people
    - **daily_vaccinations_per_million**: Daily vaccinations per million people
    - **vaccines**: Types of vaccines used
    - **source_name**: Data source name
    - **source_website**: Data source website

Data Preprocessing

- Check for missing values in each column and decide how to handle them (e.g., imputation or removal)
- Handle data types appropriately (e.g., convert the date column to datetime).
- Ensure data consistency and correctness, such as checking that percentages are within valid ranges (0–100%).

1. Import the necessary libraries :

```
import pandas as pd

import matplotlib.pyplot as plt
```

In this step, you import the Pandas library, which is essential for data manipulation and analysis.

2. Load the dataset:

```
df = pd.read_csv('cv.csv')
```

This code uses Pandas' `read_csv()` function to load the dataset from a CSV file into a Pandas DataFrame.

3. Data Exploration:

- `df.head()`: This function displays the first few rows of the dataset, allowing you to see what the data looks like at a glance.
- `df.info()`: The `info()` function provides information about the data types of each column and the number of non-null entries, which is useful for checking for missing data.

- `df.describe()`: The `describe()` function provides basic statistical summaries of the numeric columns, such as mean, standard deviation, and quartiles.

4. Data Preprocessing:

Data preprocessing involves various tasks to clean and prepare the data for analysis. Common preprocessing tasks include:

- Handling Missing Values: Use the `fillna()` function to fill missing values with a specific value or a strategy like mean or median. In the example, missing values are filled with 0.

- Feature Engineering: Create new columns or extract information from existing columns based on your analysis requirements. This step is highly specific to your analysis goals.

6. Save the Preprocessed Data:

If you want to save the preprocessed data for future use, you can use the `to_csv()` function to export it to a new CSV file. Setting `index=False` ensures that the index column is not saved to the file.

PROGRAM FOR DATA PREPROCESSING:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

print(df.head())

print(df.info())

print(df.describe())

df.fillna(0, inplace=True)

afghanistan_data = df[df['country'] == 'Afghanistan']

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

sns.lineplot(x='date', y='total_vaccinations', data=afghanistan_data)
```

```
plt.title('Total Vaccinations Over Time')
plt.xlabel('Date')
plt.ylabel('Total Vaccinations')
plt.subplot(1, 2, 2)
sns.lineplot(x='date', y='daily_vaccinations', data=afghanistan_data)
plt.title('Daily Vaccinations Over Time')
plt.xlabel('Date')
plt.ylabel('Daily Vaccinations')
plt.tight_layout()
plt.show()
df.to_csv('data.csv', index=False)
```

Output:

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

============================== RESTART: D:\naan\naan.py ==============================
       country iso_code  ...              source_name                source_website
0  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
1  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
2  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
3  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
4  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/

[5 rows x 15 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              86512 non-null  object
 1   iso_code                             86512 non-null  object
 2   date                                 86512 non-null  object
 3   total_vaccinations                   43607 non-null  float64
 4   people_vaccinated                    41294 non-null  float64
 5   people_fully_vaccinated              38802 non-null  float64
 6   daily_vaccinations_raw               35362 non-null  float64
 7   daily_vaccinations                   86213 non-null  float64
 8   total_vaccinations_per_hundred       43607 non-null  float64
 9   people_vaccinated_per_hundred        41294 non-null  float64
 10  people_fully_vaccinated_per_hundred  38802 non-null  float64
 11  daily_vaccinations_per_million       86213 non-null  float64
 12  vaccines                             86512 non-null  object
 13  source_name                          86512 non-null  object
 14  source_website                       86512 non-null  object
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
None
       total_vaccinations  ...  daily_vaccinations_per_million
count        4.360700e+04  ...                    86213.000000
mean         4.592964e+07  ...                     3257.049157
std          2.246004e+08  ...                     3934.312440
min          0.000000e+00  ...                        0.000000
25%          5.264100e+05  ...                      636.000000
50%          3.590096e+06  ...                     2050.000000
75%          1.701230e+07  ...                     4682.000000
max          3.263129e+09  ...                   117497.000000

[8 rows x 9 columns]
```

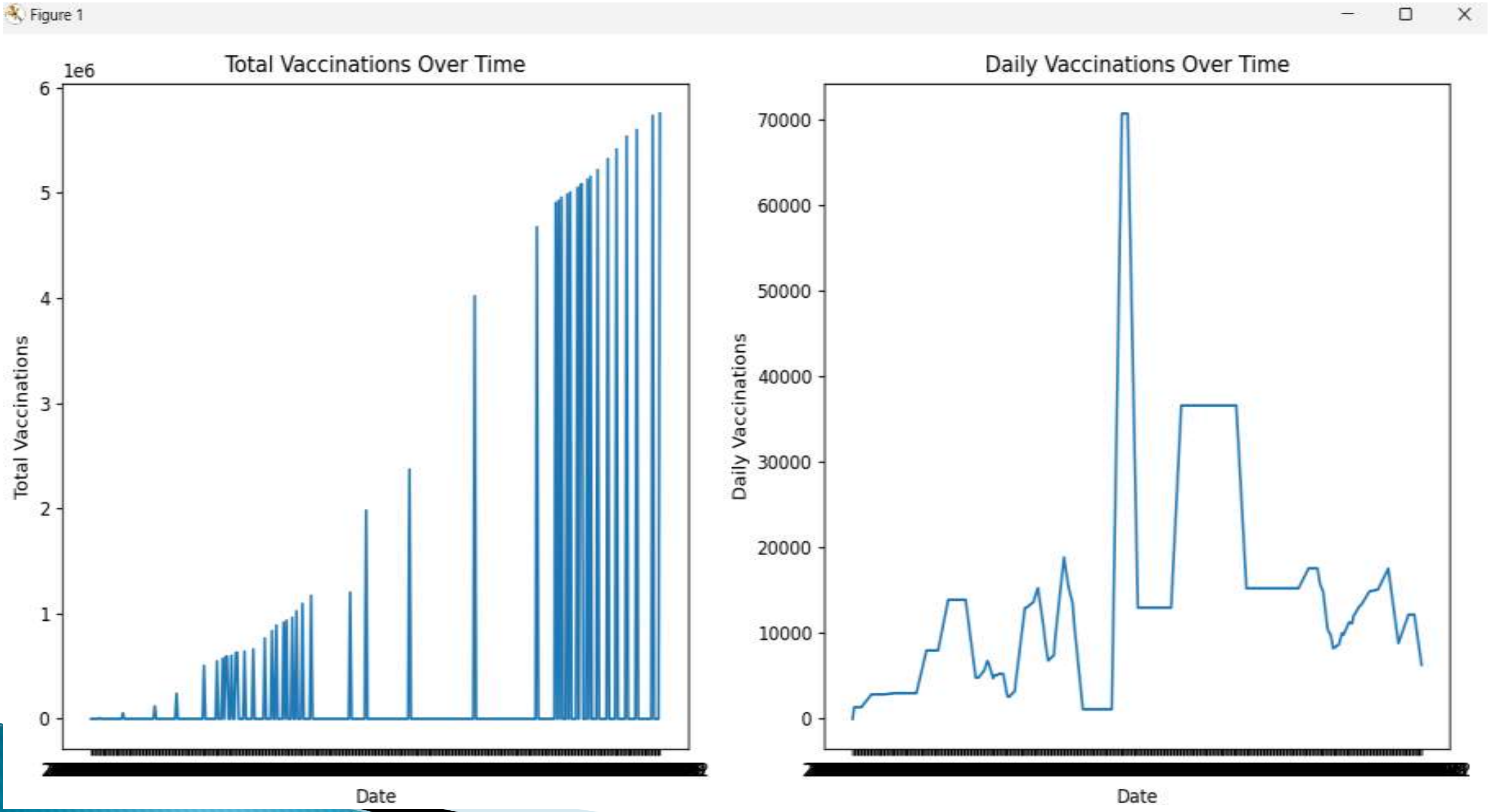


Figure 1
Total Vaccinations Over Time
1e6
6
5
4
3
2
1
0
Total Vaccinations
Date
Daily Vaccinations Over Time
70000
60000
50000
40000
30000
20000
10000
0
Daily Vaccinations
Date

Dataset after preprocessing :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	country	iso_code	date	total_vacc	people_va	people_ful	daily_vacc	daily_vacc	total_vacc	people_va	people_ful	daily_vacc	vaccines	source_na	source_website
2	Afghanista	AFG	########	0	0	0	0	0	0	0	0	0	Johnson&J	World Hea	https://covid19.who.int/
3	Afghanista	AFG	########	0	0	0	0	1367	0	0	0	34	Johnson&J	World Hea	https://covid19.who.int/
4	Afghanista	AFG	########	0	0	0	0	1367	0	0	0	34	Johnson&J	World Hea	https://covid19.who.int/
5	Afghanista	AFG	########	0	0	0	0	1367	0	0	0	34	Johnson&J	World Hea	https://covid19.who.int/
6	Afghanista	AFG	########	0	0	0	0	1367	0	0	0	34	Johnson&J	World Hea	https://covid19.who.int/
7	Afghanista	AFG	########	0	0	0	0	1367	0	0	0	34	Johnson&J	World Hea	https://covid19.who.int/
8	Afghanista	AFG	########	8200	8200	0	0	1367	0.02	0.02	0	34	Johnson&J	World Hea	https://covid19.who.int/
9	Afghanista	AFG	########	0	0	0	0	1580	0	0	0	40	Johnson&J	World Hea	https://covid19.who.int/
10	Afghanista	AFG	########	0	0	0	0	1794	0	0	0	45	Johnson&J	World Hea	https://covid19.who.int/
11	Afghanista	AFG	########	0	0	0	0	2008	0	0	0	50	Johnson&J	World Hea	https://covid19.who.int/
12	Afghanista	AFG	########	0	0	0	0	2221	0	0	0	56	Johnson&J	World Hea	https://covid19.who.int/
13	Afghanista	AFG	########	0	0	0	0	2435	0	0	0	61	Johnson&J	World Hea	https://covid19.who.int/
14	Afghanista	AFG	########	0	0	0	0	2649	0	0	0	66	Johnson&J	World Hea	https://covid19.who.int/
15	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
16	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
17	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
18	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
19	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
20	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
21	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
22	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
23	Afghanista	AFG	########	0	0	0	0	2862	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
24	Afghanista	AFG	########	54000	54000	0	0	2862	0.14	0.14	0	72	Johnson&J	World Hea	https://covid19.who.int/
25	Afghanista	AFG	########	0	0	0	0	2882	0	0	0	72	Johnson&J	World Hea	https://covid19.who.int/
26	Afghanista	AFG	########	0	0	0	0	2902	0	0	0	73	Johnson&J	World Hea	https://covid19.who.int/
27	Afghanista	AFG	########	0	0	0	0	2921	0	0	0	73	Johnson&J	World Hea	https://covid19.who.int/

Exploratory Data Analysis

- Calculate summary statistics for relevant columns (mean, median, standard deviation, etc.).
- Create various visualizations to explore trends and patterns, such as:
  - Time series plots of vaccination progress over time.
  - Bar charts to compare vaccination rates among countries.
  - Heatmaps to identify correlations between variables.
- Analyze the geographical distribution of vaccination progress using world maps.

PROGRAM FOR EDA:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
url = "data.csv"
data = pd.read_csv(url)
print("Basic Info:")
print(data.info())
print("\nSummary Statistics:")
print(data.describe())
print("\nMissing Values:")
print(data.isnull().sum())
print("\nData Types:")
print(data.dtypes)
categorical_columns = data.select_dtypes(include=['object'])
print("\nUnique Values in Categorical Columns:")
for col in categorical_columns.columns:
  unique_values = data[col].nunique()
```

```
 print(f"{col}: {unique_values} unique values")
numeric_data = data.select_dtypes(include=['number'])
for col in numeric_data.columns:
    plt.figure(figsize=(6, 6))
    sns.histplot(data=data, x=col, kde=True, bins=20)
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
plt.show()
for col in categorical_columns.columns:
    plt.figure(figsize=(6, 6))
    sns.boxplot(data=data[0:2500], x=col, y='total_vaccinations')
    plt.title(f"Box Plot of Total Vaccinations by {col}")
    plt.xticks(rotation=10)
    plt.xticks(fontsize=6)
plt.show()
plt.figure(figsize=(10, 6))
sns.lineplot(data=data, x=data.index, y='total_vaccinations')
plt.title("Total Vaccinations Over Time")
plt.xlabel("Date")
plt.ylabel("Total Vaccinations")
plt.xticks(rotation=45)
plt.show()
```

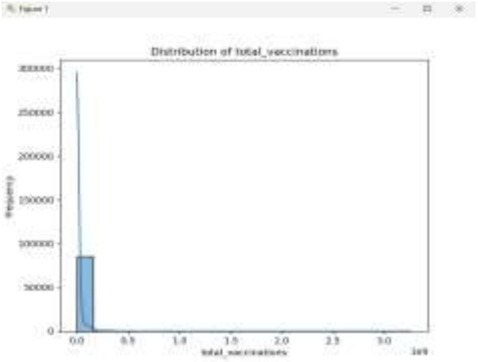
Output:

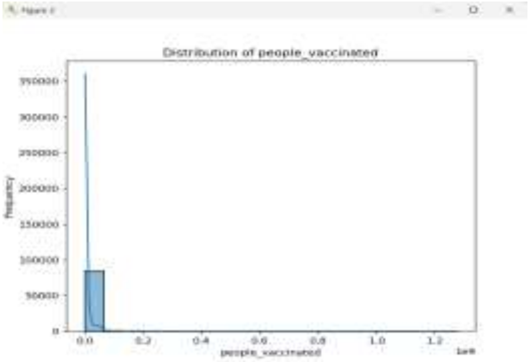
IDLE Shell 3.11.0

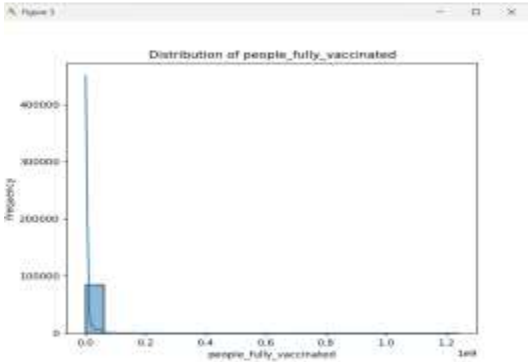
File Edit Shell Debug Options Window Help

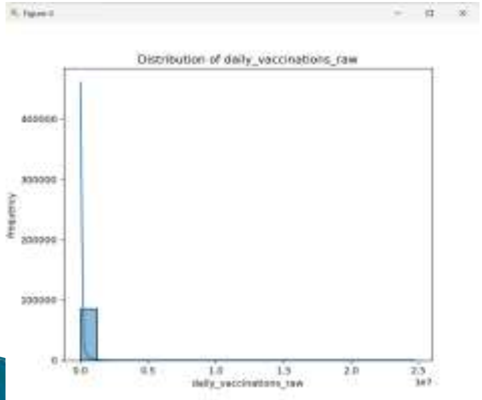
```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: D:\naan\New folder\eda.py ======================
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   country                             86512 non-null  object
 1   iso_code                            86512 non-null  object
 2   date                                86512 non-null  object
 3   total_vaccinations                  86512 non-null  float64
 4   people_vaccinated                   86512 non-null  float64
 5   people_fully_vaccinated             86512 non-null  float64
 6   daily_vaccinations_raw              86512 non-null  float64
 7   daily_vaccinations                  86512 non-null  float64
 8   total_vaccinations_per_hundred      86512 non-null  float64
 9   people_vaccinated_per_hundred       86512 non-null  float64
 10  people_fully_vaccinated_per_hundred 86512 non-null  float64
 11  daily_vaccinations_per_million      86512 non-null  float64
 12  vaccines                            86512 non-null  object
 13  source_name                         86512 non-null  object
 14  source_website                      86512 non-null  object
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
None

Summary Statistics:
       total_vaccinations  ...  daily_vaccinations_per_million
count        8.651200e+04  ...                    86512.000000
mean         2.315117e+07  ...                     3245.792248
std          1.611037e+08  ...                     3932.156455
min          0.000000e+00  ...                        0.000000
25%          0.000000e+00  ...                      629.000000
50%          1.008000e+03  ...                     2036.000000
75%          3.697554e+06  ...                     4667.000000
max          3.263129e+09  ...                   117497.000000

[8 rows x 9 columns]

Missing Values:
country                                0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
```

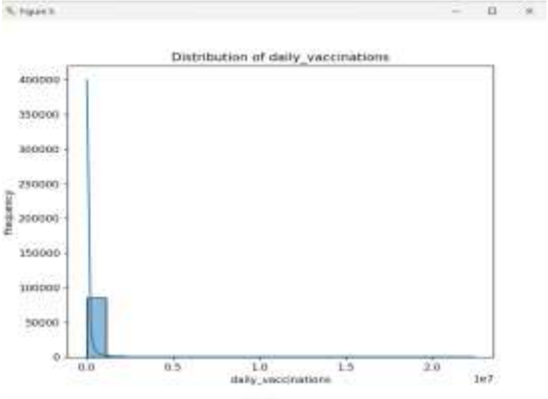
```
people_fully_vaccinated                0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
daily_vaccinations_per_million         0
vaccines                               0
source_name                            0
source_website                         0
dtype: int64

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

Unique Values in Categorical Columns:
country: 223 unique values
iso_code: 223 unique values
date: 483 unique values
vaccines: 84 unique values
source_name: 81 unique values
source_website: 119 unique values
>>>
```

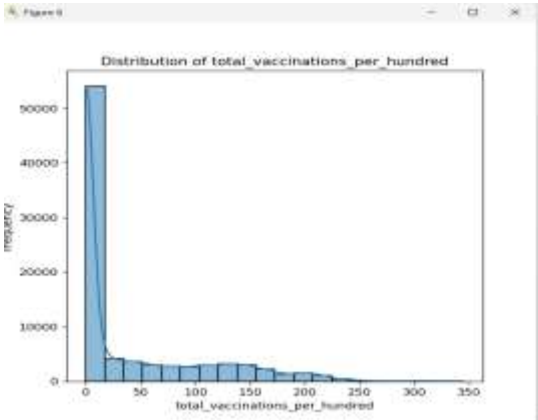

Distribution of total_vaccinations
Frequency
total_vaccinations

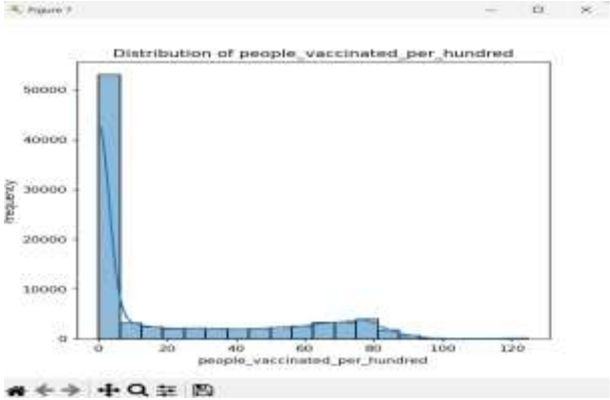

Distribution of people_vaccinated
Frequency
people_vaccinated

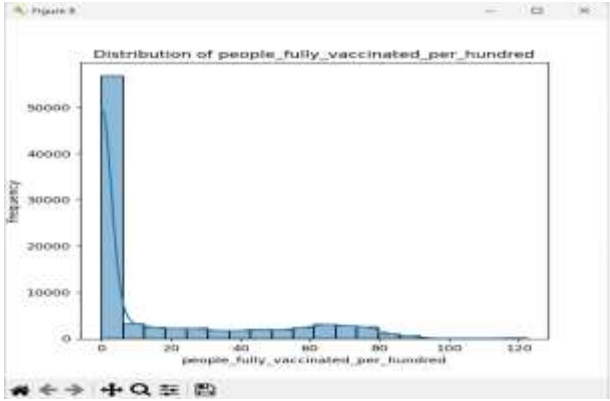

Distribution of people_fully_vaccinated
Frequency
people_fully_vaccinated

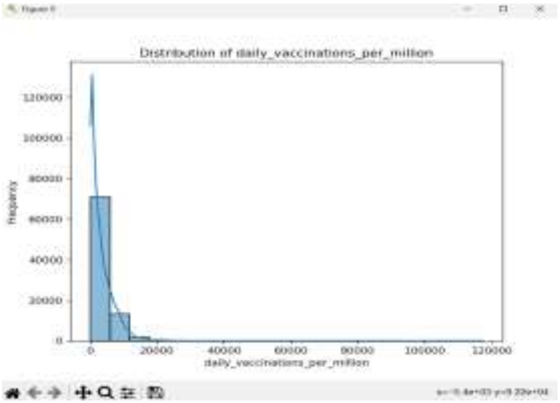

Distribution of daily_vaccinations_raw
Frequency
daily_vaccinations_raw

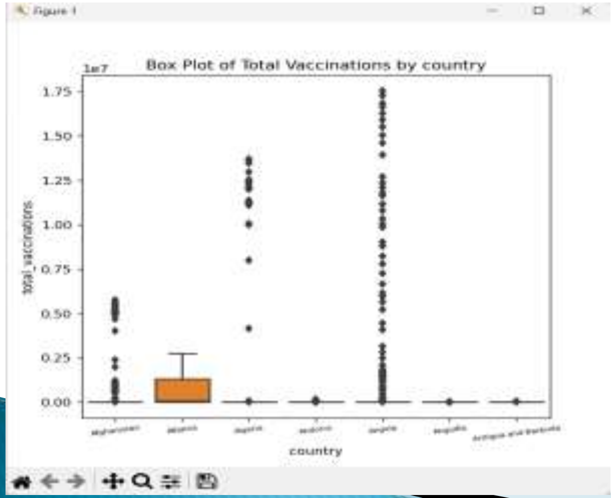

Distribution of daily_vaccinations
Frequency
daily_vaccinations

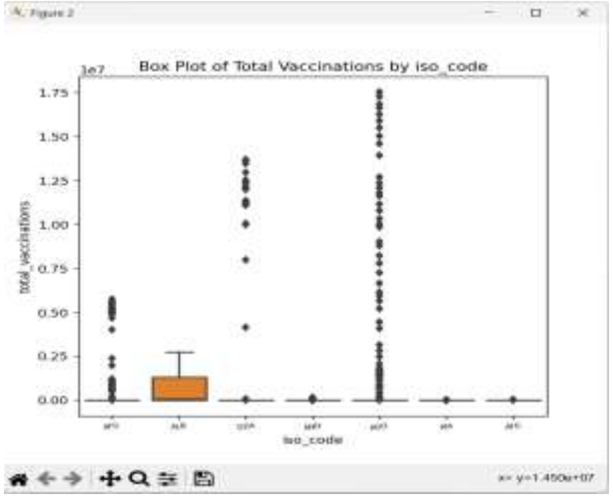

Distribution of total_vaccinations_per_hundred
Frequency
total_vaccinations_per_hundred

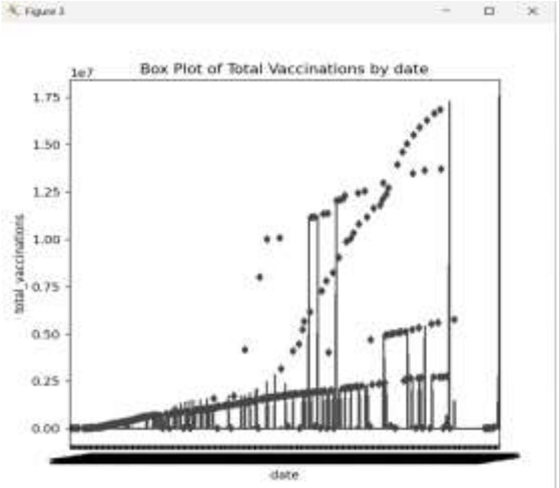

Distribution of people_vaccinated_per_hundred
Frequency
people_vaccinated_per_hundred


Distribution of people_fully_vaccinated_per_hundred
Frequency
people_fully_vaccinated_per_hundred


Distribution of daily_vaccinations_per_million
Frequency
daily_vaccinations_per_million


Box Plot of Total Vaccinations by country
total_vaccinations
country


Box Plot of Total Vaccinations by iso_code
total_vaccinations
iso_code


Box Plot of Total Vaccinations by date
total_vaccinations
date

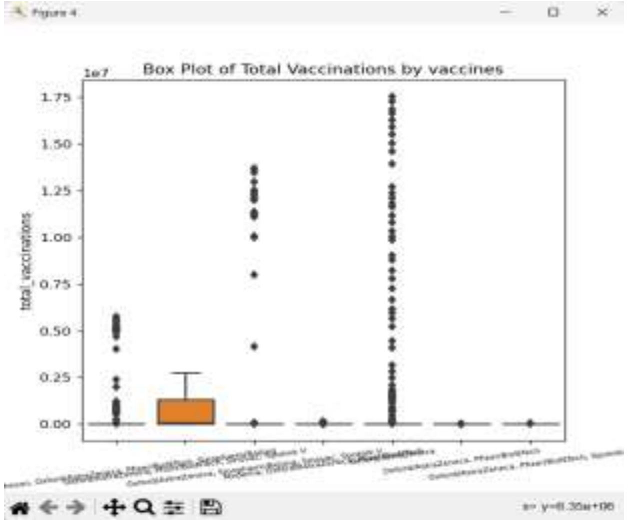


Figure 4
Box Plot of Total Vaccinations by vaccines
1e7
total_vaccinations
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

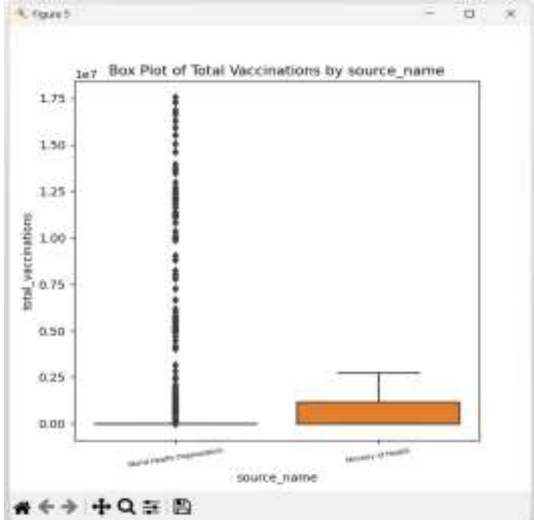


Figure 5
Box Plot of Total Vaccinations by source_name
1e7
total_vaccinations
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
source_name

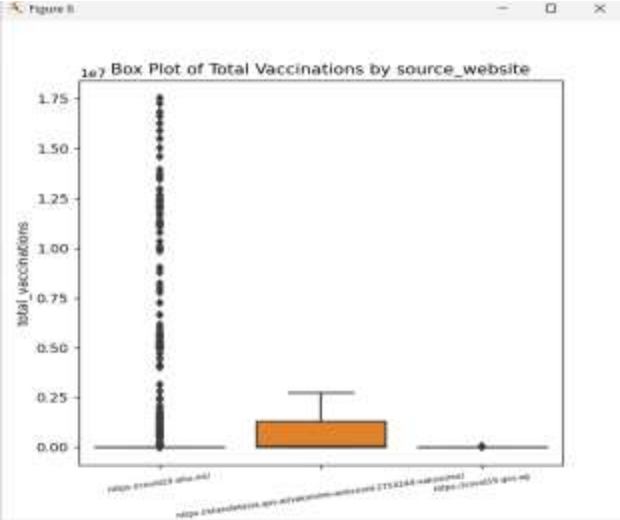


Figure 6
Box Plot of Total Vaccinations by source_website
1e7
total_vaccinations
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

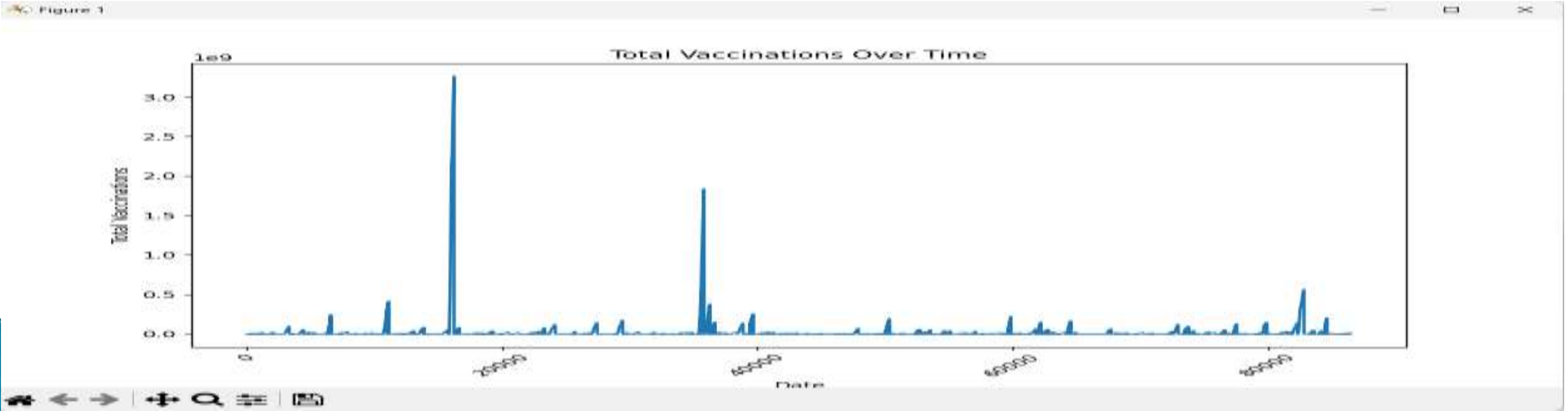


Figure 1
Total Vaccinations Over Time
1e9
Total Vaccinations
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
Date

Statistical Analysis

- Conduct hypothesis testing to answer specific research questions (e.g., comparing vaccination rates between countries using t-tests).
- Use regression analysis to model the impact of variables (e.g., vaccine type or GDP) on vaccination rates.

PROGRAM FOR STATISTICAL ANALYSIS :

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('data.csv')
print(df.head())
print(df.info())
print(df.describe())
df.fillna(0, inplace=True)
afghanistan_data = df[df['country'] == 'Afghanistan']
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.lineplot(x='date', y='total_vaccinations', data=afghanistan_data)
```

```
plt.title('Total Vaccinations Over Time')

plt.xlabel('Date')

plt.ylabel('Total Vaccinations')

plt.subplot(1, 2, 2)

sns.lineplot(x='date', y='daily_vaccinations', data=afghanistan_data)

plt.title('Daily Vaccinations Over Time')

plt.xlabel('Date')

plt.ylabel('Daily Vaccinations')

plt.tight_layout()

plt.show()

df.to_csv('data.csv', index=False)
```

Output:

IDLE Shell 3.11.0

File Edit Shell Debug Options Window Help

```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:\naan\New folder\eda.py ======================
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              86512 non-null  object
 1   iso_code                             86512 non-null  object
 2   date                                 86512 non-null  object
 3   total_vaccinations                   86512 non-null  float64
 4   people_vaccinated                    86512 non-null  float64
 5   people_fully_vaccinated              86512 non-null  float64
 6   daily_vaccinations_raw               86512 non-null  float64
 7   daily_vaccinations                   86512 non-null  float64
 8   total_vaccinations_per_hundred       86512 non-null  float64
 9   people_vaccinated_per_hundred        86512 non-null  float64
 10  people_fully_vaccinated_per_hundred  86512 non-null  float64
 11  daily_vaccinations_per_million       86512 non-null  float64
 12  vaccines                             86512 non-null  object
 13  source_name                          86512 non-null  object
 14  source_website                       86512 non-null  object
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
None

Summary Statistics:
       total_vaccinations  ...  daily_vaccinations_per_million
count        8.651200e+04  ...                    86512.000000
mean         2.315117e+07  ...                     3245.792248
std          1.611037e+08  ...                     3932.156455
min          0.000000e+00  ...                        0.000000
25%          0.000000e+00  ...                      629.000000
50%          1.008000e+03  ...                     2036.000000
75%          3.697554e+06  ...                     4667.000000
max          3.263129e+09  ...                   117497.000000

[8 rows x 9 columns]

Missing Values:
country                                0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
```

```
people_fully_vaccinated                0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
daily_vaccinations_per_million         0
vaccines                               0
source_name                            0
source_website                         0
dtype: int64

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

Unique Values in Categorical Columns:
country: 223 unique values
iso_code: 223 unique values
date: 483 unique values
vaccines: 84 unique values
source_name: 81 unique values
source_website: 119 unique values
>>>
```

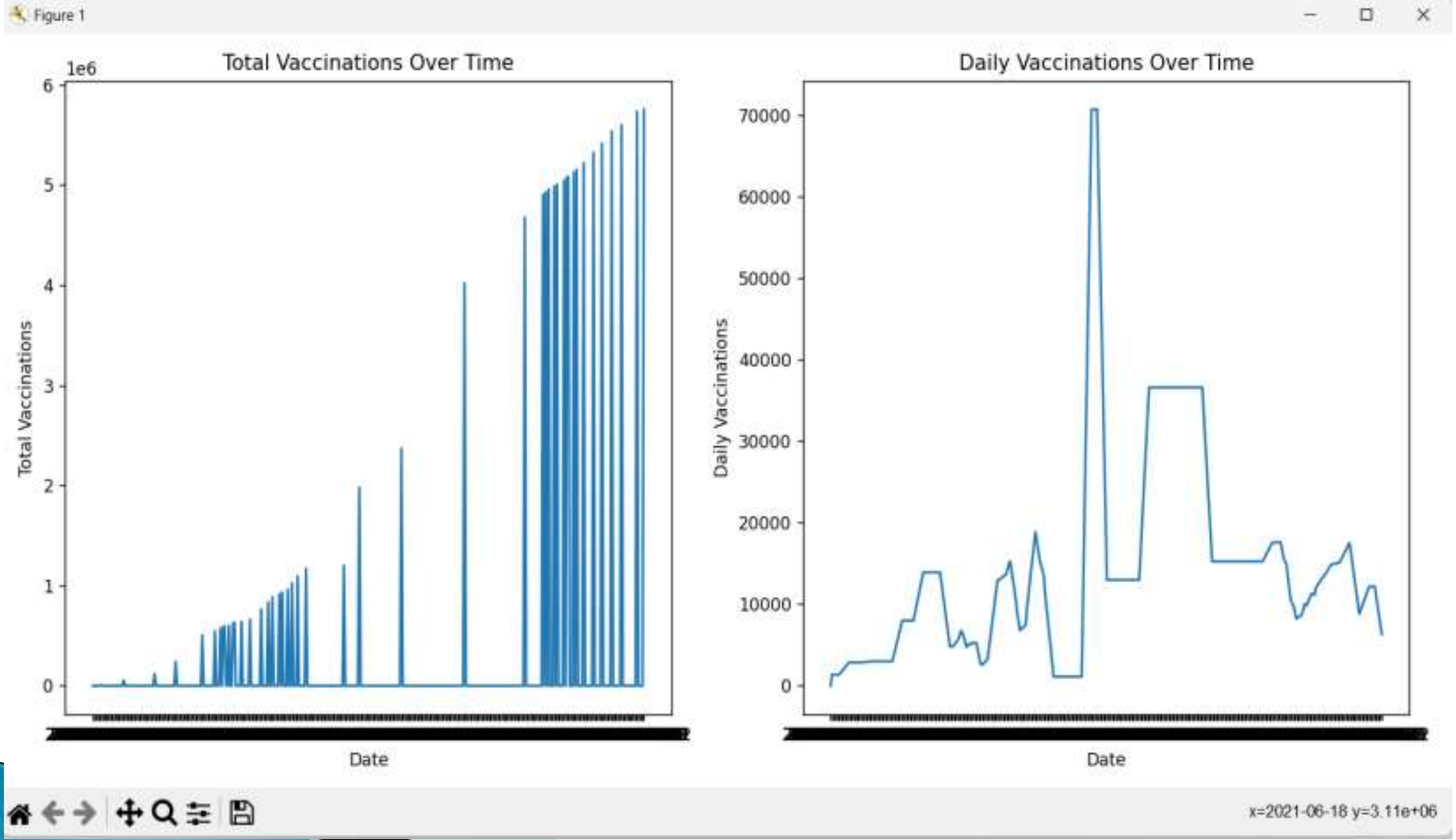


Figure 1
Total Vaccinations Over Time
1e6
6
5
4
3
2
1
0
Total Vaccinations
Date
Daily Vaccinations Over Time
70000
60000
50000
40000
30000
20000
10000
0
Daily Vaccinations
Date
x=2021-06-18 y=3.11e+06

Visualization

- Develop informative and visually appealing charts and graphs.
- Consider creating interactive visualizations for online sharing or presentations.
- Ensure that your visualizations are well-labeled and easy to interpret.

PROGRAM FOR VISUALIZATION :

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
url = "data.csv"
data = pd.read_csv(url)
sns.set(style="whitegrid")
plt.figure(figsize=(8, 8))
sns.barplot(x='country', y='total_vaccinations', data=data.head(200))
plt.xticks(rotation=45)
plt.title('Total Vaccinations in Top 10 Countries')
plt.xlabel('Country')
plt.ylabel('Total Vaccinations')
plt.show()
afghanistan_data = data[data['country'] == 'Afghanistan']
plt.figure(figsize=(8, 8))
sns.lineplot(x='date', y='daily_vaccinations', data=afghanistan_data[0:100])
plt.xticks(rotation=90)
plt.xticks(fontsize=6)
```

```
plt.title('Daily Vaccinations Trend in Afghanistan')
plt.xlabel('Date')
plt.ylabel('Daily Vaccinations')
plt.show()
plt.figure(figsize=(8, 8))
sns.scatterplot(x='total_vaccinations', y='people_vaccinated', data=data)
plt.title('Total Vaccinations vs. People Vaccinated')
plt.xlabel('Total Vaccinations')
plt.ylabel('People Vaccinated')
plt.show()
plt.figure(figsize=(8, 8))
sns.boxplot(x='vaccines', y='daily_vaccinations', data=data)
plt.xticks(rotation=90)
plt.xticks(fontsize=6)
plt.title('Distribution of Daily Vaccinations by Vaccine Type')
plt.xlabel('Vaccine Type')
plt.ylabel('Daily Vaccinations')
plt.show()
```

Output:

```
IDLE Shell 3.11.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:\naan\New folder\naan.py ======================
       country iso_code  ...           source_name               source_website
0  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
1  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
2  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
3  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/
4  Afghanistan      AFG  ...  World Health Organization  https://covid19.who.int/

[5 rows x 15 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   country                              86512 non-null  object 
 1   iso_code                             86512 non-null  object 
 2   date                                 86512 non-null  object 
 3   total_vaccinations                   86512 non-null  float64
 4   people_vaccinated                    86512 non-null  float64
 5   people_fully_vaccinated              86512 non-null  float64
 6   daily_vaccinations_raw               86512 non-null  float64
 7   daily_vaccinations                   86512 non-null  float64
 8   total_vaccinations_per_hundred       86512 non-null  float64
 9   people_vaccinated_per_hundred        86512 non-null  float64
 10  people_fully_vaccinated_per_hundred  86512 non-null  float64
 11  daily_vaccinations_per_million      86512 non-null  float64
 12  vaccines                             86512 non-null  object 
 13  source_name                          86512 non-null  object 
 14  source_website                       86512 non-null  object 
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
None
       total_vaccinations  ...  daily_vaccinations_per_million
count        8.651200e+04  ...                    86512.000000
mean         2.315117e+07  ...                     3245.792248
std          1.611037e+08  ...                     3932.156455
min          0.000000e+00  ...                        0.000000
25%          0.000000e+00  ...                      629.000000
50%          1.008000e+03  ...                     2036.000000
75%          3.697554e+06  ...                     4667.000000
max          3.263129e+09  ...                   117497.000000

[8 rows x 9 columns]
>>>
```

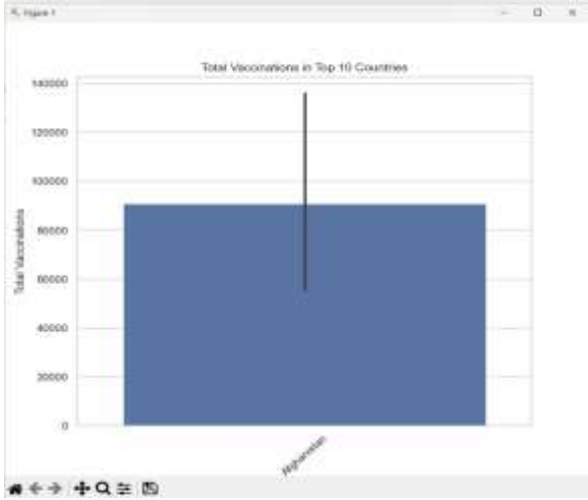
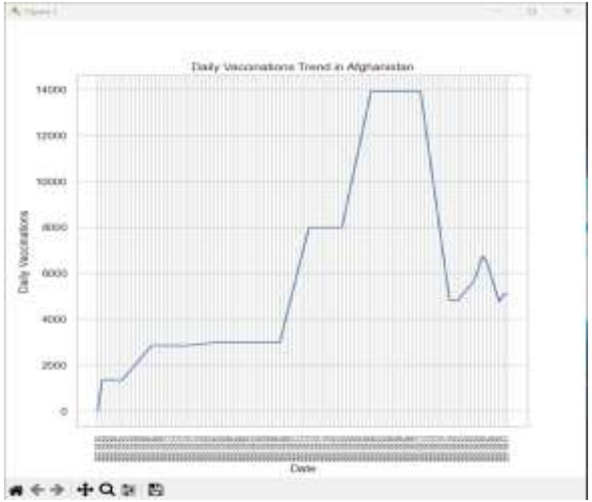
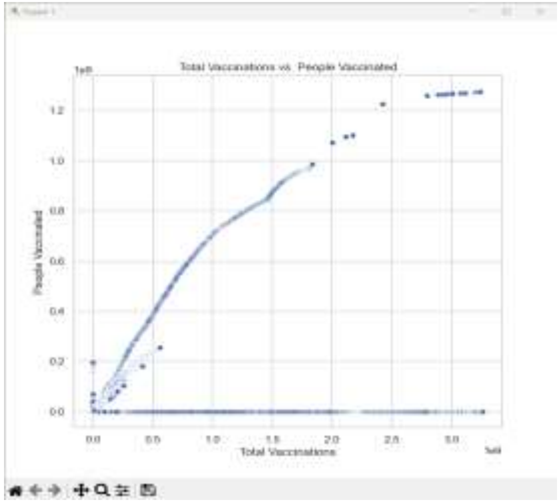


Figure 1
Total Vaccinations in Top 10 Countries
Total Vaccinations
140000
120000
100000
80000
60000
40000
20000
0
Afghanistan


Daily Vaccinations Trend in Afghanistan
Daily Vaccinations
14000
12000
10000
8000
6000
4000
2000
0
Date


Total Vaccinations vs. People Vaccinated
1e8
People Vaccinated
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Total Vaccinations
1e9

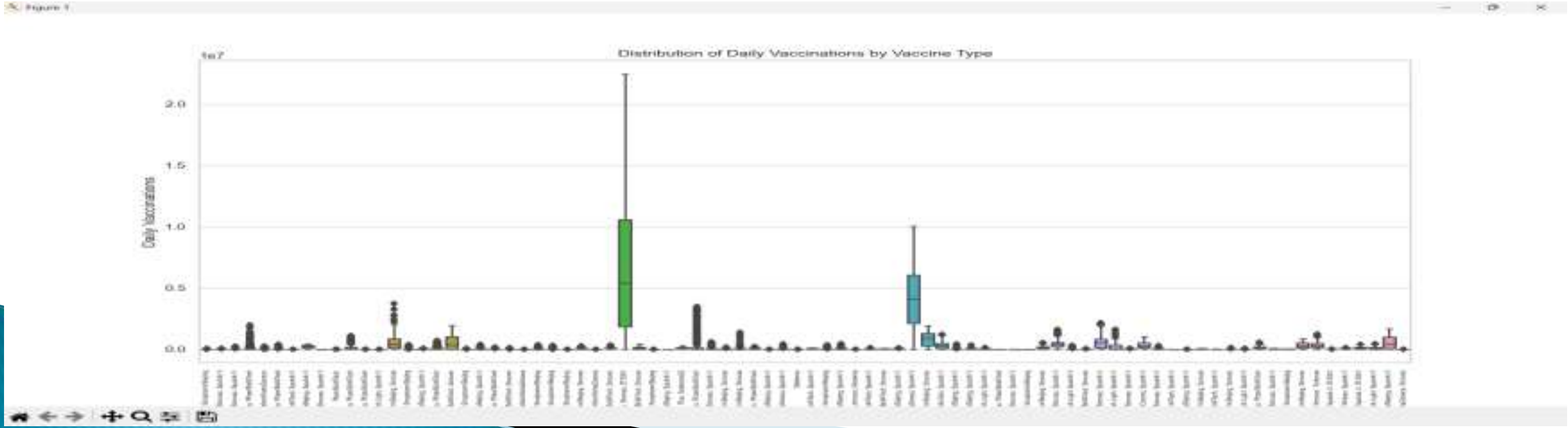


Figure 1
Distribution of Daily Vaccinations by Vaccine Type
1e7
Daily Vaccinations
2.0
1.5
1.0
0.5
0.0

Insight and Recommendation

- **Summarize your findings and highlight key insights.**
- **Provide actionable recommendations based on your analysis. For example**:
    - Suggest strategies to improve vaccine distribution in countries with low vaccination rates.
    - Identify factors that correlate with higher vaccination rates.
    - Propose further research questions or areas of investigation.

Present key Finding

- **Vaccination Progress**: The dataset tracks the progress of COVID-19 vaccinations in Afghanistan from February to March 2021. It includes information on the total number of vaccinations administered, daily vaccination rates, and the number of people vaccinated.

- **Vaccine Types**: Afghanistan administered vaccines from multiple manufacturers, including Johnson & Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, and Sinopharm/Beijing. This reflects the country's efforts to secure a variety of vaccines.

- **Daily Vaccination Trends**: The daily vaccinations and daily vaccinations per million columns provide insights into the country's daily vaccination rates. There is an observed increase in daily vaccinations over time, with a notable increase on March 16, 2021, when 54,000 vaccinations were administered.

- **Fully Vaccinated Individuals**: The dataset contains a column for "people_fully_vaccinated," which initially contains null values. It appears that Afghanistan began recording fully vaccinated individuals later in the dataset.

- **Vaccination Coverage**: The columns "total_vaccinations_per_hundred," "people_vaccinated_per_hundred," and "people_fully_vaccinated_per_hundred" indicate the vaccination coverage as a percentage of the population. These metrics provide an estimate of the proportion of the population that has been vaccinated.

- **Data Source**: The source of this data is the World Health Organization (WHO), and the dataset includes a reference to the source's website.

Conclusion

1.The analysis of the COVID-19 vaccine dataset has provided valuable insights into the global vaccination effort. It is evident that vaccination progress is influenced by a combination of factors, including vaccine availability, distribution strategies, and regional disparities in healthcare resources.

2.To improve vaccination rates worldwide and ensure equitable access to vaccines, policymakers and public health officials should consider the following:

- Continuously monitor and adjust vaccination distribution strategies to address disparities.
- Promote public awareness and confidence in vaccines to encourage higher uptake.
- Collaborate with international organizations to ensure the availability of vaccines in underserved regions.
- Use data-driven insights to optimize vaccination campaigns and target high-risk populations.

3.This analysis serves as a foundation for further research and policy decisions aimed at effectively combatting the covid-19 pandemic and achieving global vaccination goals.