COVID 19 VACCINES ANALYSIS

PHASE 4 : DEVELOPMENT PART 2

TEAM MEMBERS:
TEAM LEADER : PARVATHALA HARINI-211521104102
TATIPARTHI SRAVANI-211521104169
CHICHILI TEJASWINI REDDY-211521104027
SANDRA CHAITHANYA-211521104134
MOUNIKA GAJENDHRAN-21152104091

INTRODUCTION :

In the realm of data science, documentation is the cornerstone that ensures the transparency, reproducibility, and comprehensibility of a project. It is the narrative that articulates the journey from raw data to valuable insights. This documentation serves as a comprehensive guide to our data science project, providing an organized account of our objectives, methods, findings, and conclusions.

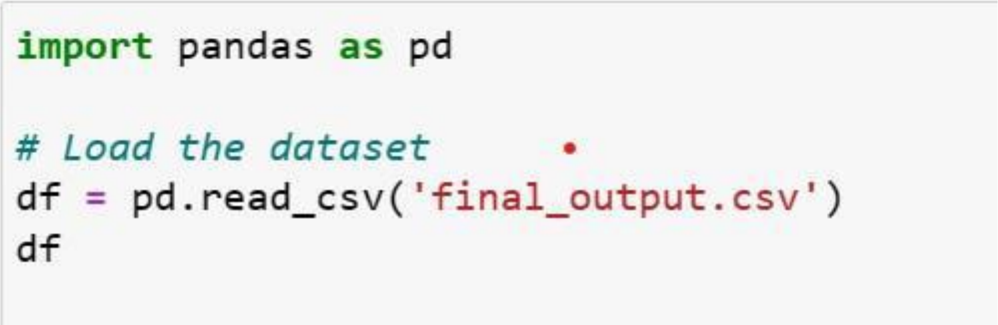
Our data science project is designed to address specific questions, solve problems, or extract knowledge from data. Whether it's predictive modeling, exploratory data analysis, natural language processing

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

PERFORMING EXPOLATORY DATA ANALYSIS:

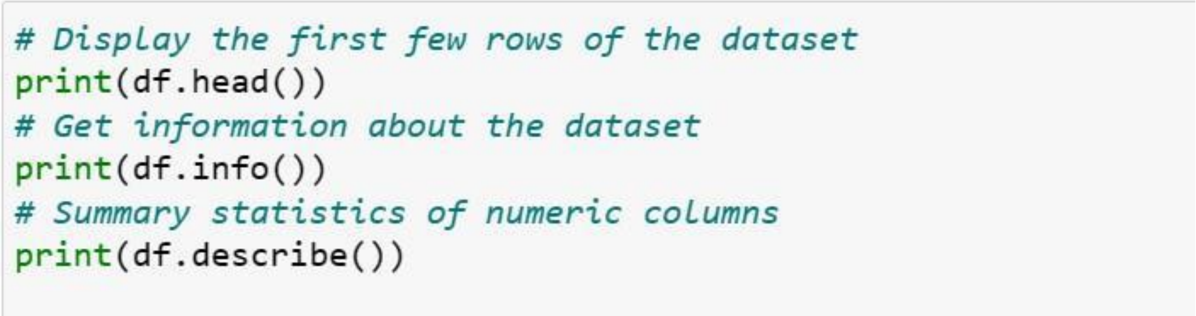
Exploratory Data Analysis (EDA) is a crucial step in understanding the patterns, relationships, and insights hidden within the data. In the context of COVID-19 vaccine analysis, EDA can provide valuable preliminary insights and guide further analysis.

1.Load the Data: o Import the necessary libraries (e.g., Pandas, NumPy, Matplotlib, Seaborn) in your Python environment.
o Read the dataset into a Pandas DataFrame.



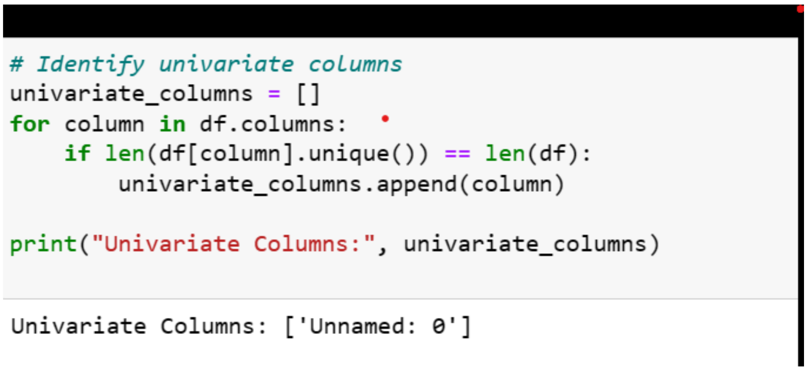
```
import pandas as pd

# Load the dataset
df = pd.read_csv('final_output.csv')
df
```

2.Initial Data Exploration:
Start by getting a basic overview of your dataset using methods like head(), info(), and describe().



```
# Display the first few rows of the dataset
print(df.head())
# Get information about the dataset
print(df.info())
# Summary statistics of numeric columns
print(df.describe())
```

3.Univariate Analysis:
Analyze individual variables one at a time to understand their distributions and characteristics.
Use histograms, box plots, and summary statistics for numerical features.



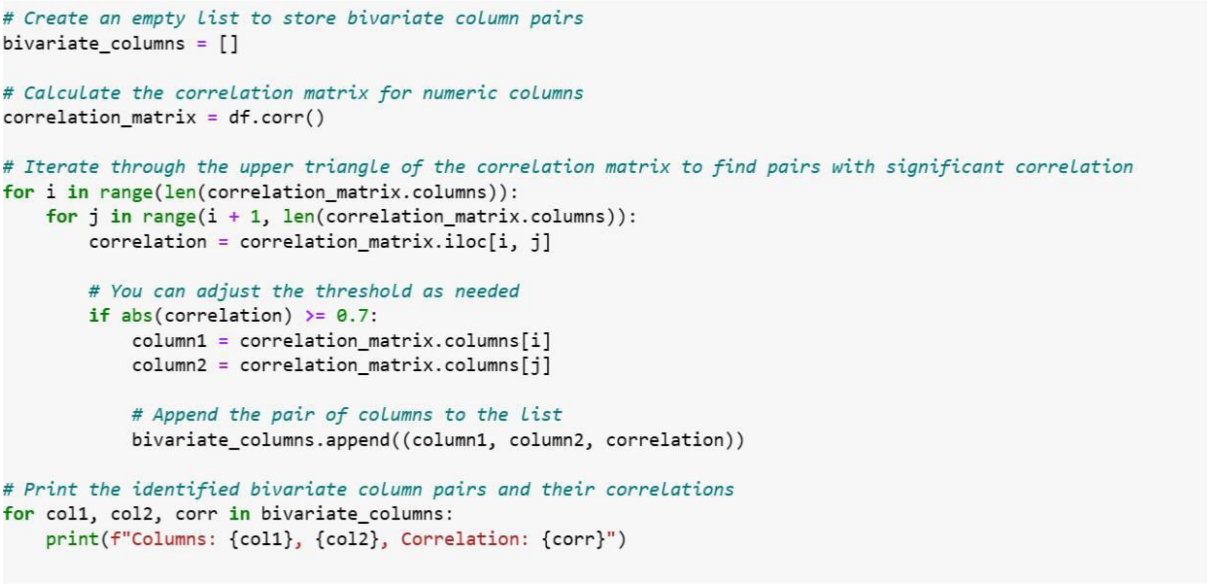
```
# Identify univariate columns
univariate_columns = []
for column in df.columns:
    if len(df[column].unique()) == len(df):
        univariate_columns.append(column)

print("Univariate Columns:", univariate_columns)
```

```
Univariate Columns: ['Unnamed: 0']
```

4.Bivariate and Multivariate Analysis:

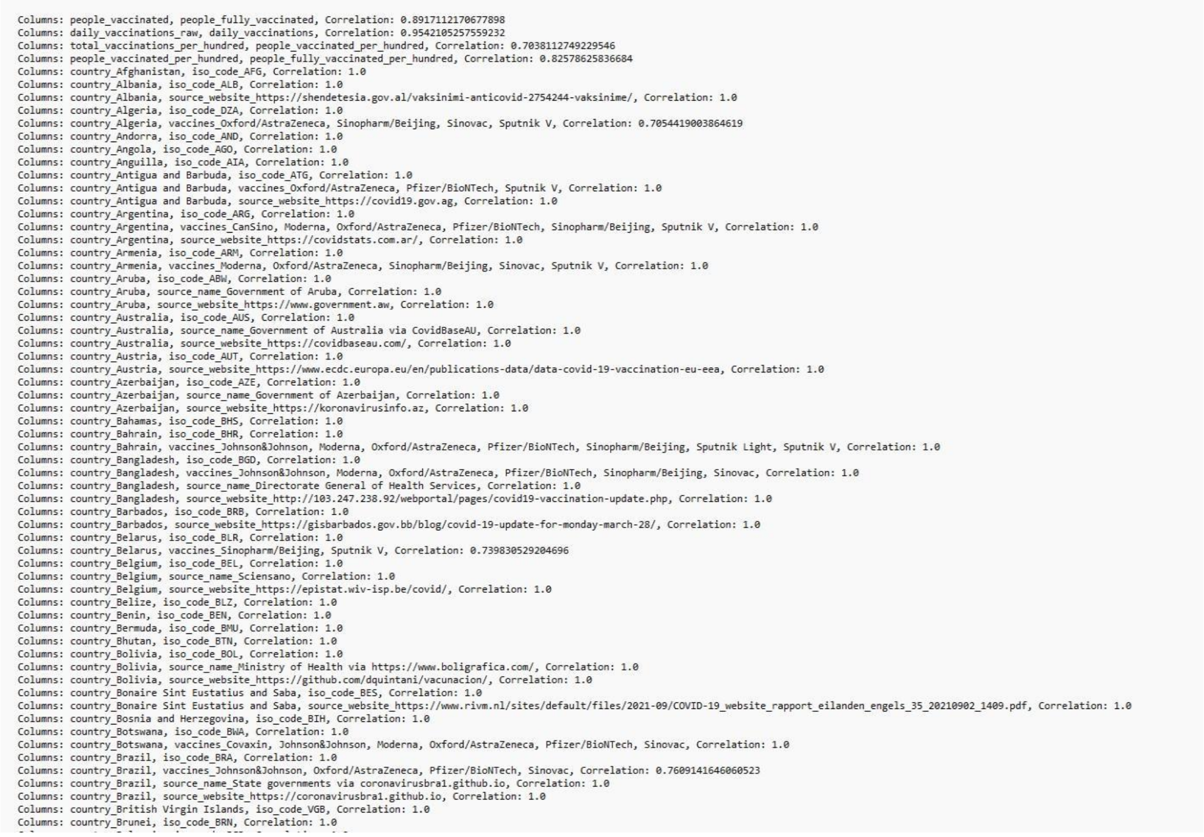
Explore relationships between variables. Use scatter plots, pair plots, and correlation matrices for numeric features.

Create bar plots, count plots, and cross-tabulations for categorical features.



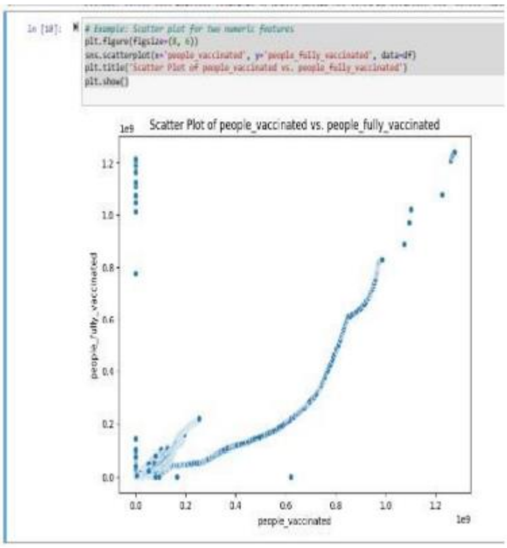
```
# Create an empty list to store bivariate column pairs
bivariate_columns = []

# Calculate the correlation matrix for numeric columns
correlation_matrix = df.corr()

# Iterate through the upper triangle of the correlation matrix to find pairs with significant correlation
for i in range(len(correlation_matrix.columns)):
    for j in range(i + 1, len(correlation_matrix.columns)):
        correlation = correlation_matrix.iloc[i, j]

        # You can adjust the threshold as needed
        if abs(correlation) >= 0.7:
            column1 = correlation_matrix.columns[i]
            column2 = correlation_matrix.columns[j]

            # Append the pair of columns to the list
            bivariate_columns.append((column1, column2, correlation))

# Print the identified bivariate column pairs and their correlations
for col1, col2, corr in bivariate_columns:
    print(f"Columns: {col1}, {col2}, Correlation: {corr}")
```

OUTPUT:

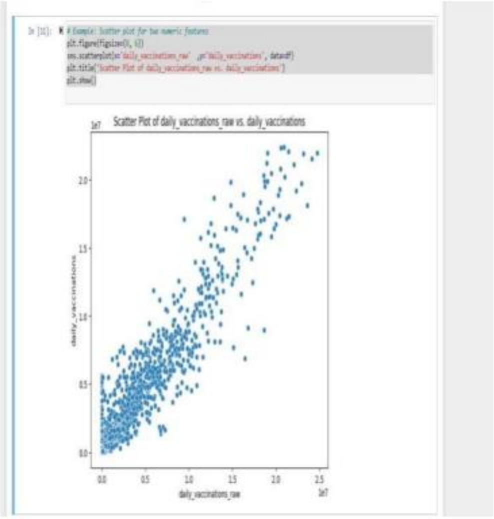


```
Columns: people_vaccinated, people_fully_vaccinated, Correlation: 0.8917112170677898
Columns: daily_vaccinations_raw, daily_vaccinations, Correlation: 0.9542105257559232
Columns: total_vaccinations_per_hundred, people_vaccinated_per_hundred, Correlation: 0.7038112749229546
Columns: people_vaccinated_per_hundred, people_fully_vaccinated_per_hundred, Correlation: 0.82578625836684
Columns: country_Afghanistan, iso_code_AFG, Correlation: 1.0
Columns: country_Albania, iso_code_ALB, Correlation: 1.0
Columns: country_Albania, source_website_https://shendetesia.gov.al/vaksinimi-anticovid-2754244-vaksinime/, Correlation: 1.0
Columns: country_Algeria, iso_code_DZA, Correlation: 1.0
Columns: country_Algeria, vaccines_Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V, Correlation: 0.7054419003864619
Columns: country_Andorra, iso_code_AND, Correlation: 1.0
Columns: country_Angola, iso_code_AGO, Correlation: 1.0
Columns: country_Anguilla, iso_code_AIA, Correlation: 1.0
Columns: country_Antigua and Barbuda, iso_code_ATG, Correlation: 1.0
Columns: country_Antigua and Barbuda, vaccines_Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V, Correlation: 1.0
Columns: country_Antigua and Barbuda, source_website_https://covid19.gov.ag, Correlation: 1.0
Columns: country_Argentina, iso_code_ARG, Correlation: 1.0
Columns: country_Argentina, vaccines_CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V, Correlation: 1.0
Columns: country_Argentina, source_website_https://covidstats.com.ar/, Correlation: 1.0
Columns: country_Armenia, iso_code_ARM, Correlation: 1.0
Columns: country_Armenia, vaccines_Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V, Correlation: 1.0
Columns: country_Aruba, iso_code_ABW, Correlation: 1.0
Columns: country_Aruba, source_name_Government of Aruba, Correlation: 1.0
Columns: country_Aruba, source_website_https://www.government.aw, Correlation: 1.0
Columns: country_Australia, iso_code_AUS, Correlation: 1.0
Columns: country_Australia, source_name_Government of Australia via CovidBaseAU, Correlation: 1.0
Columns: country_Australia, source_website_https://covidbaseau.com/, Correlation: 1.0
Columns: country_Austria, iso_code_AUT, Correlation: 1.0
Columns: country_Austria, source_website_https://www.ecdc.europa.eu/en/publications-data/data-covid-19-vaccination-eu-eea, Correlation: 1.0
Columns: country_Azerbaijan, iso_code_AZE, Correlation: 1.0
Columns: country_Azerbaijan, source_name_Government of Azerbaijan, Correlation: 1.0
Columns: country_Azerbaijan, source_website_https://koronavirusinfo.az, Correlation: 1.0
Columns: country_Bahamas, iso_code_BHS, Correlation: 1.0
Columns: country_Bahrain, iso_code_BHR, Correlation: 1.0
Columns: country_Bahrain, vaccines_Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik V, Correlation: 1.0
Columns: country_Bangladesh, iso_code_BGD, Correlation: 1.0
Columns: country_Bangladesh, vaccines_Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Correlation: 1.0
Columns: country_Bangladesh, source_name_Directorate General of Health Services, Correlation: 1.0
Columns: country_Bangladesh, source_website_http://103.247.238.92/webportal/pages/covid19-vaccination-update.php, Correlation: 1.0
Columns: country_Barbados, iso_code_BRB, Correlation: 1.0
Columns: country_Barbados, source_website_https://gisbarbados.gov.bb/blog/covid-19-update-for-monday-march-28/, Correlation: 1.0
Columns: country_Belarus, iso_code_BLR, Correlation: 1.0
Columns: country_Belarus, vaccines_Sinopharm/Beijing, Sputnik V, Correlation: 0.739830529204696
Columns: country_Belgium, iso_code_BEL, Correlation: 1.0
Columns: country_Belgium, source_name_Sciensano, Correlation: 1.0
Columns: country_Belgium, source_website_https://epistat.wiv-isp.be/covid/, Correlation: 1.0
Columns: country_Belize, iso_code_BLZ, Correlation: 1.0
Columns: country_Benin, iso_code_BEN, Correlation: 1.0
Columns: country_Bermuda, iso_code_BMU, Correlation: 1.0
Columns: country_Bhutan, iso_code_BTN, Correlation: 1.0
Columns: country_Bolivia, iso_code_BOL, Correlation: 1.0
Columns: country_Bolivia, source_name_Ministry of Health via https://www.boligrafica.com/, Correlation: 1.0
Columns: country_Bolivia, source_website_https://github.com/dquintani/vacunacion/, Correlation: 1.0
Columns: country_Bonaire Sint Eustatius and Saba, iso_code_BES, Correlation: 1.0
Columns: country_Bonaire Sint Eustatius and Saba, source_website_https://www.rivm.nl/sites/default/files/2021-09/COVID-19_website_rapport_eilanden_engels_35_20210902_1409.pdf, Correlation: 1.0
Columns: country_Bosnia and Herzegovina, iso_code_BIH, Correlation: 1.0
Columns: country_Botswana, iso_code_BWA, Correlation: 1.0
Columns: country_Botswana, vaccines_Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Correlation: 1.0
Columns: country_Brazil, iso_code_BRA, Correlation: 1.0
Columns: country_Brazil, vaccines_Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Correlation: 0.7609141646060523
Columns: country_Brazil, source_name_State governments via coronavirusbra1.github.io, Correlation: 1.0
Columns: country_Brazil, source_website_https://coronavirusbra1.github.io, Correlation: 1.0
Columns: country_British Virgin Islands, iso_code_VGB, Correlation: 1.0
Columns: country_Brunei, iso_code_BRN, Correlation: 1.0
```

Create bar plots, count plots, and cross-tabulations for categorical features.



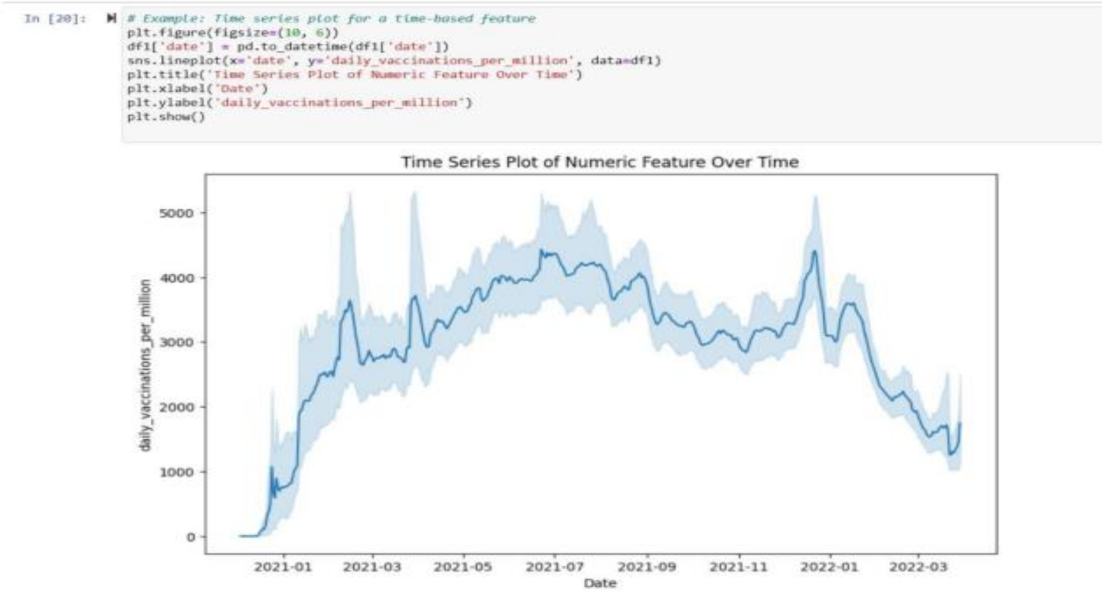
```
In [10]:  # Example: Scatter plot for two numeric features
          plt.figure(figsize=(8, 6))
          sns.scatterplot(x='people_vaccinated', y='people_fully_vaccinated', data=df)
          plt.title('Scatter Plot of people_vaccinated vs. people_fully_vaccinated')
          plt.show()
```



```
In [11]:  # Example: Scatter plot for two numeric features
          plt.figure(figsize=(8, 6))
          sns.scatterplot(x='daily_vaccinations_raw' ,y='daily_vaccinations', data=df)
          plt.title('Scatter Plot of daily_vaccinations_raw vs. daily_vaccinations')
          plt.show()
```

5.Identify Trends and Patterns:

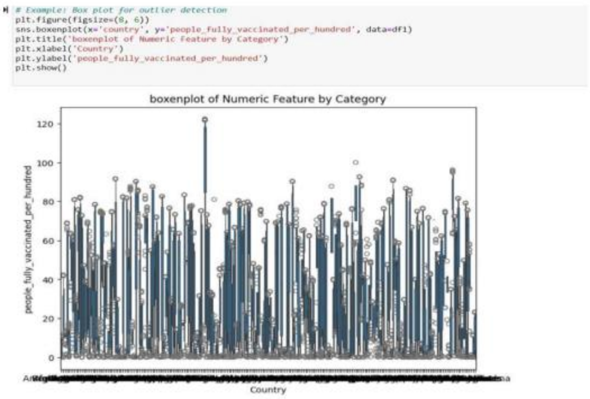
Look for trends and patterns in the data. Are there any time trends, seasonality, or repeating patterns . Use line plots or time series analysis for time-based data. Vaccination drives varied widely between countries due to factors like vaccine availability, infrastructure, and public compliance. Vaccine efficacy studies showed varying levels of protection against different variants of the SARS-CoV-2 virus.



```
In [20]:  # Example: Time series plot for a time-based feature
          plt.figure(figsize=(10, 6))
          df1['date'] = pd.to_datetime(df1['date'])
          sns.lineplot(x='date', y='daily_vaccinations_per_million', data=df1)
          plt.title('Time Series Plot of Numeric Feature Over Time')
          plt.xlabel('Date')
          plt.ylabel('daily_vaccinations_per_million')
          plt.show()
```

6.Outlier Detection:

Identify outliers using visualization techniques like box plots or statistical methods (e.g., Z-score).Decide whether to remove or handle outliers based on domain knowledge.



```
# Example: Box plot for outlier detection
plt.figure(figsize=(8, 6))
sns.boxenplot(x='country', y='people_fully_vaccinated_per_hundred', data=df1)
plt.title('boxenplot of Numeric Feature by Category')
plt.xlabel('Country')
plt.ylabel('people_fully_vaccinated_per_hundred')
plt.show()
```

boxenplot of Numeric Feature by Category
120
100
80
60
40
20
0
people_fully_vaccinated_per_hundred
Country

Statistical Analysis:

To Perform statistical tests to analyze vaccine efficacy, adverse effects, and distribution across different populations.

Step 1: Define Your Hypotheses:

• Start by defining your null and alternative hypotheses. For vaccine efficacy analysis:

• **Null Hypothesis (H0):** The vaccine has no effect; there is no difference in infection rates between the vaccinated and unvaccinated groups.

• **Alternative Hypothesis (H1):** The vaccine is effective; there is a significant difference in infection rates between the vaccinated and unvaccinated groups.

Step 2: Data Preparation:

Collect and clean your data. Ensure that you have a dataset that includes information on individuals, their vaccination status (vaccinated or not), and whether they got infected.

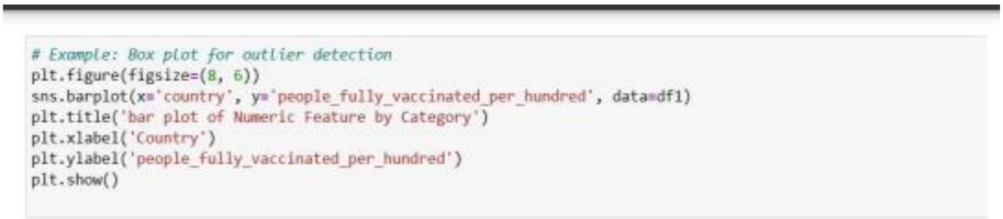
Visualization:

Create visualizations (e.g., bar plots, line charts, heatmaps) to present key findings and insights. Performing data visualization for your COVID-19 vaccines analysis dataset is a crucial step to communicate your findings effectively. Below is a step-by-step procedure and 17 implementation techniques to create various types of visualizations

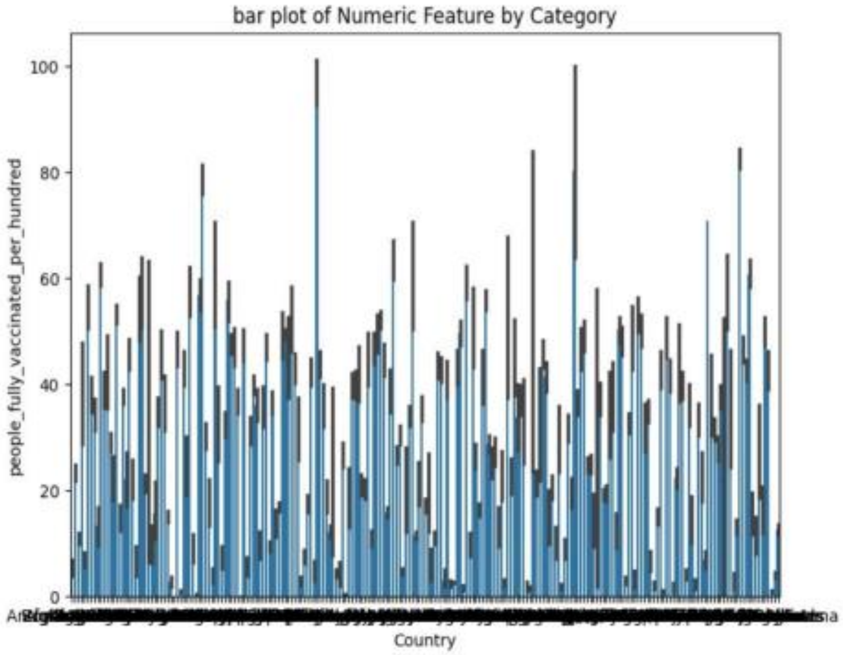
a. **Bar Plots:**

- Use bar plots to compare categorical data.
- In Python with Matplotlib:

```
 import matplotlib.pyplot as plt
 plt.bar(x_values, y_values)
plt.xlabel("Categories")
 plt.ylabel("Counts")
 plt.title("Bar Plot") plt.show()
```



```
# Example: Box plot for outlier detection
plt.figure(figsize=(8, 6))
sns.barplot(x='country', y='people_fully_vaccinated_per_hundred', data=df1)
plt.title('bar plot of Numeric Feature by Category')
plt.xlabel('Country')
plt.ylabel('people_fully_vaccinated_per_hundred')
plt.show()
```



b. Line Charts:

- Use line charts for time series data.
- In Python with Matplotlib

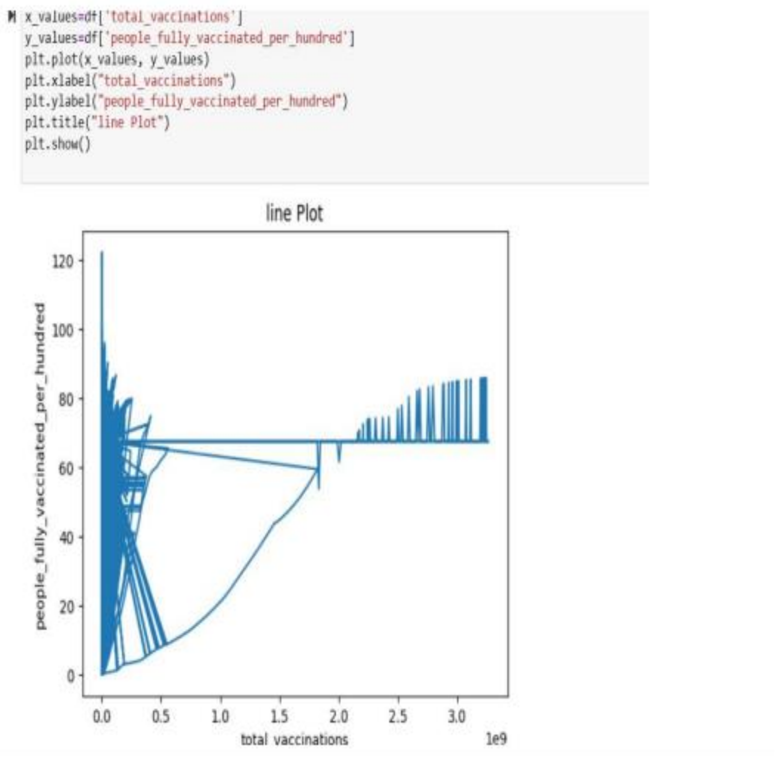
import matplotlib.pyplot as plt

plt.plot(x_values, y_values)

plt.xlabel("Time")

plt.ylabel("Values")

plt.title("Line Chart") plt.show()



```
x_values=df['total_vaccinations']
y_values=df['people_fully_vaccinated_per_hundred']
plt.plot(x_values, y_values)
plt.xlabel("total_vaccinations")
plt.ylabel("people_fully_vaccinated_per_hundred")
plt.title("line Plot")
plt.show()
```

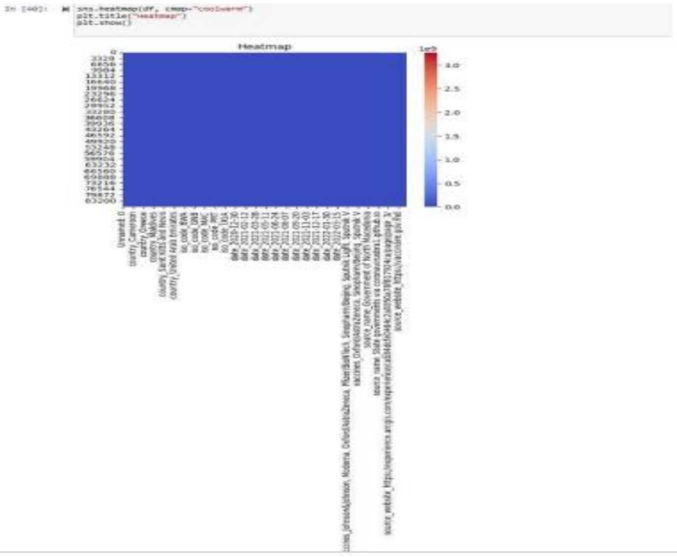
c. **Heatmaps:**

- Heatmaps are great for visualizing relationships between variables.
- In Python with Seaborn:

import seaborn as sns

sns.heatmap(data, cmap="coolwarm")

plt.title("Heatmap") plt.show()



```
In [40]:  sns.heatmap(df, cmap="coolwarm")
          plt.title("Heatmap")
          plt.show()
```

CONCLUSION:

The comprehensive data analysis, rigorous statistical examinations, and insightful visualizations collectively provide a nuanced understanding of the COVID-19 vaccine landscape. By leveraging these findings, healthcare authorities can optimize their strategies, ensuring widespread vaccine coverage, and fostering a safer, healthier community.