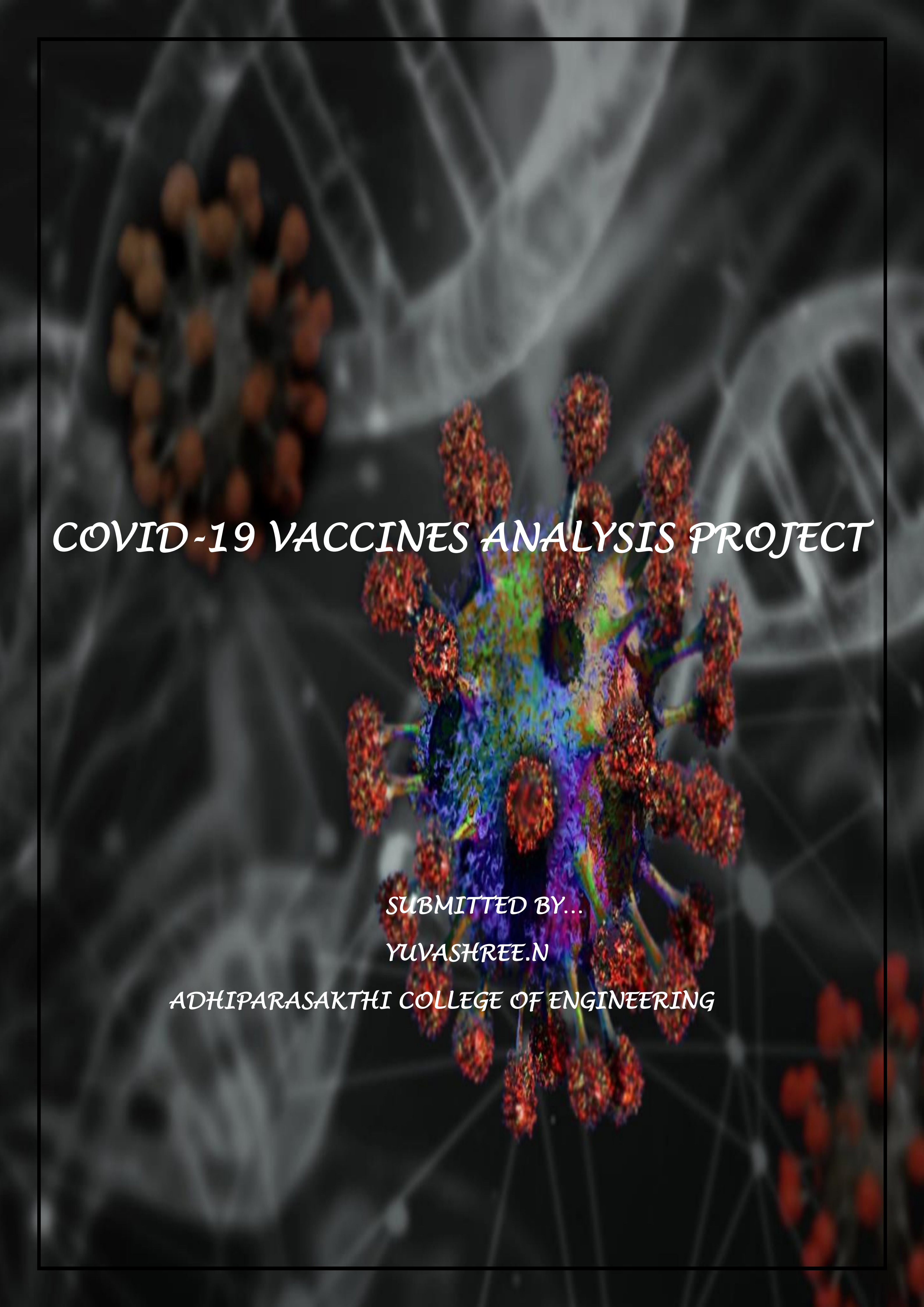


COVID-19 VACCINES ANALYSIS PROJECT

SUBMITTED BY...

YUVASHREE.N

ADHIPARASAKTHI COLLEGE OF ENGINEERING

PROJECT TITLE: COVID-19 VACCINES ANALYSIS.

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

AGENDA:

- **To understand technique such as lambdas and manipulating CSV files.**
- **To describe common python functionality and features used for data Science.**
- **To query data Frame structure for the cleaning and processing.**

PANDAS LIBRARY

- Tools for reading and writing data between in-memory data structure and different formats.
- Intelligent data alignment and integrated handling of missing data.
- Aggregating or transforming data with a powerful group by engine allowing split-apply-combine operations on data sets.
- High performance merging and joining of datasets.
- Columns can be inserted and deleted from data structures for size mutability.

DATAFRAMES

- ✓ DATAFRAME is a two dimensional data structure with columns of potentially different types.
- ✓ It is like a spread sheet or a sql table ,or a directory of series object.
- ✓ It is generally the most commonly used pandas object.
- ✓ Like series, data frame accepts many different kinds of input.

OBJECTIVES:

- To learn the steps,needed to be taken to prepare the data for an analysis.
- To learn how to look at the data to find a good measure to establish the analysis based upon.
- To learn to visualize the result of analysis.

PROBLEM:

Is there any relationship between the spread of the coronavirus and how happy people living in that country are?

DATASET:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully	daily_vac	daily_vaccinations	total_vaccinatio	people_va	people_fu	daily_vacc	vaccines	source_na	source_website		
2	Afghanista	AFG	22-02-2021	0	0				0	0			Johnson&J	World Hea	https://covid19.who.int/		
3	Afghanista	AFG	23-02-2021					1367				34	Johnson&J	World Hea	https://covid19.who.int/		
4	Afghanista	AFG	24-02-2021					1367				34	Johnson&J	World Hea	https://covid19.who.int/		
5	Afghanista	AFG	25-02-2021					1367				34	Johnson&J	World Hea	https://covid19.who.int/		
6	Afghanista	AFG	26-02-2021					1367				34	Johnson&J	World Hea	https://covid19.who.int/		
7	Afghanista	AFG	27-02-2021					1367				34	Johnson&J	World Hea	https://covid19.who.int/		
8	Afghanista	AFG	28-02-2021	8200	8200			1367	0.02	0.02		34	Johnson&J	World Hea	https://covid19.who.int/		
9	Afghanista	AFG	01-03-2021					1580				40	Johnson&J	World Hea	https://covid19.who.int/		
10	Afghanista	AFG	02-03-2021					1794				45	Johnson&J	World Hea	https://covid19.who.int/		
11	Afghanista	AFG	03-03-2021					2008				50	Johnson&J	World Hea	https://covid19.who.int/		
12	Afghanista	AFG	04-03-2021					2221				56	Johnson&J	World Hea	https://covid19.who.int/		
13	Afghanista	AFG	05-03-2021					2435				61	Johnson&J	World Hea	https://covid19.who.int/		
14	Afghanista	AFG	06-03-2021					2649				66	Johnson&J	World Hea	https://covid19.who.int/		
15	Afghanista	AFG	07-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
16	Afghanista	AFG	08-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
17	Afghanista	AFG	09-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
18	Afghanista	AFG	10-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
19	Afghanista	AFG	11-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
20	Afghanista	AFG	12-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
21	Afghanista	AFG	13-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
22	Afghanista	AFG	14-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
23	Afghanista	AFG	15-03-2021					2862				72	Johnson&J	World Hea	https://covid19.who.int/		
24	Afghanista	AFG	16-03-2021	54000	54000			2862	0.14	0.14		72	Johnson&J	World Hea	https://covid19.who.int/		
25	Afghanista	AFG	17-03-2021					2882				72	Johnson&J	World Hea	https://covid19.who.int/		
26	Afghanista	AFG	18-03-2021					2902				73	Johnson&J	World Hea	https://covid19.who.int/		
27	Afghanista	AFG	19-03-2021					2921				73	Johnson&J	World Hea	https://covid19.who.int/		
28	Afghanista	AFG	20-03-2021					2941				74	Johnson&J	World Hea	https://covid19.who.int/		
29	Afghanista	AFG	21-03-2021					2961				74	Johnson&J	World Hea	https://covid19.who.int/		
30	Afghanista	AFG	22-03-2021					2980				75	Johnson&J	World Hea	https://covid19.who.int/		
31	Afghanista	AFG	23-03-2021					3000				75	Johnson&J	World Hea	https://covid19.who.int/		
32	Afghanista	AFG	24-03-2021					3000				75	Johnson&J	World Hea	https://covid19.who.int/		
33	Afghanista	AFG	25-03-2021					3000				75	Johnson&J	World Hea	https://covid19.who.int/		
34	Afghanista	AFG	26-03-2021					3000				75	Johnson&J	World Hea	https://covid19.who.int/		

DATA PREPROCESS

Most of the time of data analysis and modeling is spent on data preparation and processing i.e., loading, cleaning and rearranging the data, etc. Further, because of Python libraries, Pandas give us high performance, flexible, and high-level environment for processing the data. Various functionalities are available for pandas to process the data effectively.

HIERARCHICAL INDEXING:

For enhancing the capabilities of Data Processing, we have to use some indexing that helps to sort the data based on the labels. So, Hierarchical indexing is comes into the picture and defined as an essential feature of pandas that helps us to use the multiple index levels.

INPUT:

```
fully_vaccinated.reset_index()
```

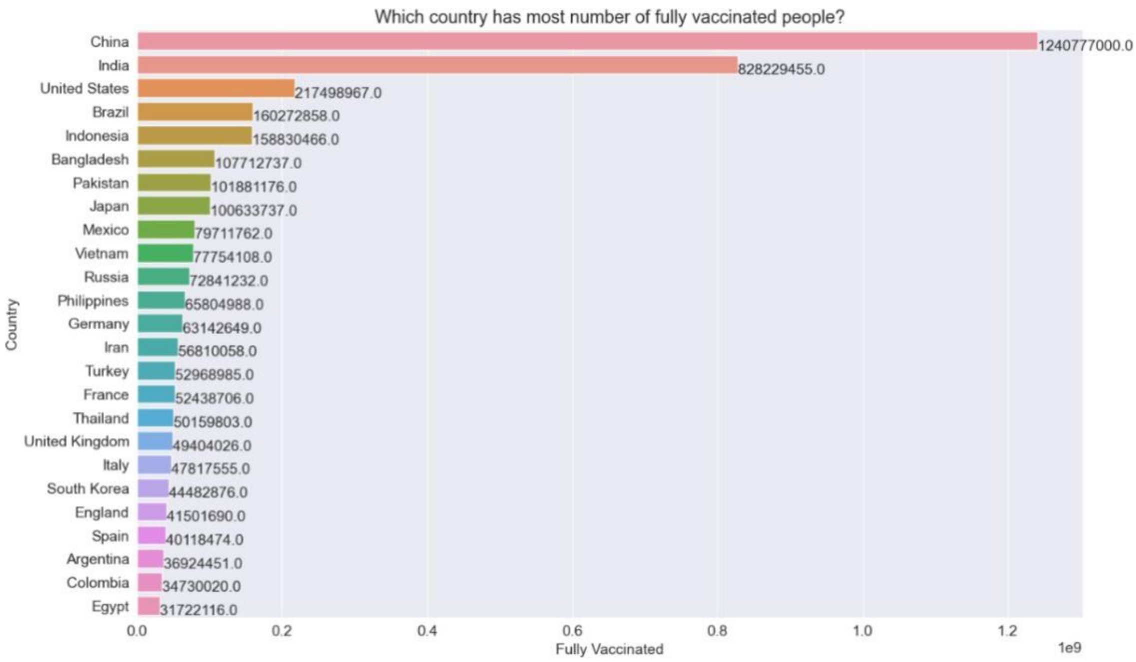
OUTPUT:

	country	people_fully_vaccinated
0	China	1.240777e+09
1	India	8.282295e+08
2	United States	2.174990e+08
3	Brazil	1.602729e+08
4	Indonesia	1.588305e+08
5	Bangladesh	1.077127e+08
6	Pakistan	1.018812e+08
7	Japan	1.006337e+08
8	Mexico	7.971176e+07
9	Vietnam	7.775411e+07
10	Russia	7.284123e+07
11	Philippines	6.580499e+07
12	Germany	6.314265e+07
13	Iran	5.681006e+07
14	Turkey	5.296898e+07
15	France	5.243871e+07
16	Thailand	5.015980e+07
17	United Kingdom	4.940403e+07
18	Italy	4.781756e+07
19	South Korea	4.448288e+07
20	England	4.150169e+07
21	Spain	4.011847e+07
22	Argentina	3.692445e+07
23	Colombia	3.473002e+07
24	Egypt	3.172212e+07

INPUT:

```
plt.figure(figsize=(16,10))
ax = sns.barplot(x=fully_vaccinated, y=fully_vaccinated.index)
plt.xlabel("Fully Vaccinated")
plt.ylabel("Country");
plt.title('Which country has most number of fully vaccinated people?');

for patch in ax.patches:
    width = patch.get_width()
    height = patch.get_height()
    x = patch.get_x()
    y = patch.get_y()

    plt.text(width + x, height + y, '{:.1f} '.format(width))
```

OUTPUT:



PARTIAL INDEXING:

Partial indexing can be defined as a way to choose the particular index from a hierarchical indexing.

INPUT:

```
daily_vaccinations_per_million = vaccinations_df.groupby("country")["daily_vaccinations_per_million"].max().sort_values(ascending
```

```
daily_vaccinations_per_million.reset_index()
```

OUTPUT:

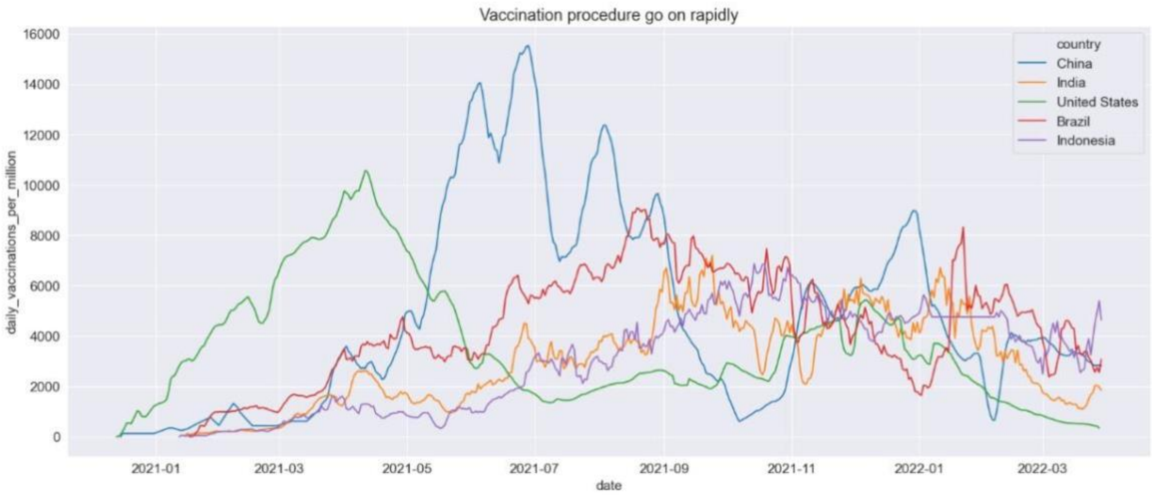
	country	daily_vaccinations_per_million
0	Bhutan	117497.0
1	Isle of Man	70706.0
2	Botswana	55891.0
3	Niue	53903.0
4	Falkland Islands	53571.0
5	Nauru	51504.0
6	Nicaragua	46446.0
7	Cook Islands	46210.0
8	Mongolia	37684.0
9	Gibraltar	31700.0
10	Wallis and Futuna	30918.0
11	Cuba	28441.0
12	Guernsey	27562.0
13	Saint Helena	27071.0
14	Aruba	24992.0

INPUT:

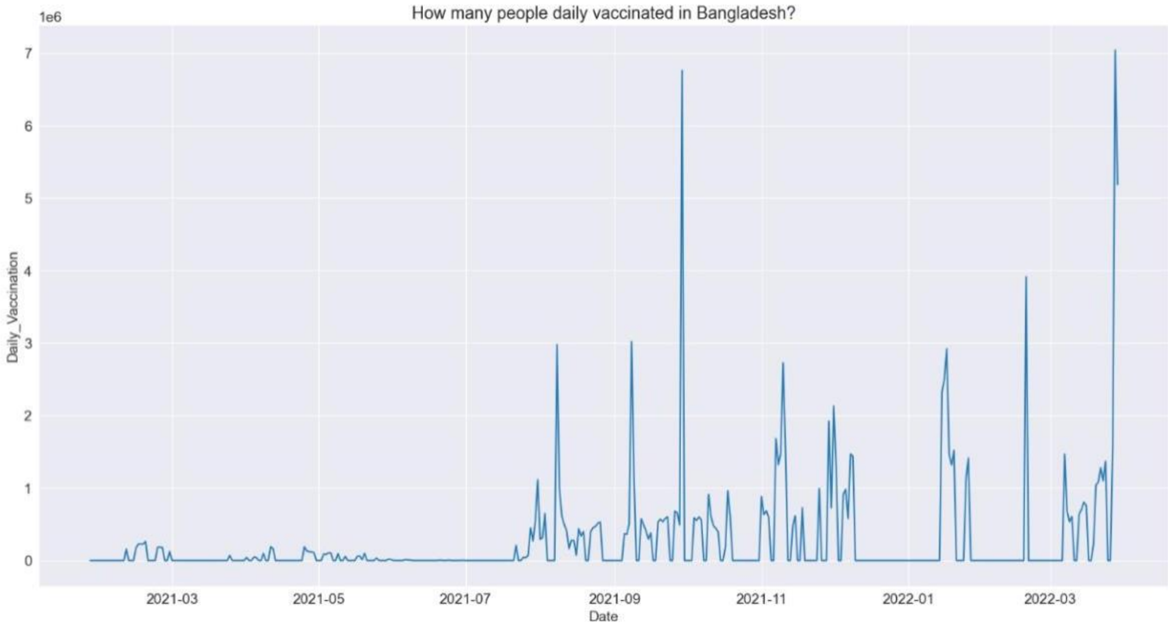
VACCINATIONS RAPIDLY GOES ON

```
plt.figure(figsize=(20,8))
sns.lineplot(top_countries['date'], top_countries['daily_vaccinations_per_million'], hue= top_countries['country'], ci= False)
plt.title('Vaccination procedure go on rapidly');
```

OUTPUT:




```
plt.figure(figsize=(20,10))
sns.lineplot(x=bangladesh_df.date, y=bangladesh_df.daily_vaccinations_raw)
plt.xlabel("Date")
plt.ylabel("Daily_Vaccination")
plt.title('How many people daily vaccinated in Bangladesh?');
```



COLUMN INDEXING:

Remember that, since, column-indexing requires two dimensional data, the column indexing is possible only for DataFrame(not for Series). Let's create new DataFrame for demonstrating the columns with multiple index,

INPUT:

```
vaccinations_df.columns
```

OUTPUT:

```
Index(['country', 'iso_code', 'date', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated',
       'daily_vaccinations_raw', 'daily_vaccinations',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'daily_vaccinations_per_million',
       'vaccines', 'source_name', 'source_website'],
      dtype='object')
```

EXPLORATORY DATA ANALYSIS

- EDA is applied to **investigate** the data and **summarize** the key insights.
- It will give you the basic understanding of your data, it's **distribution**, null values and much more.
- You can either explore data using graphs or through some python **functions.**
- There will be two type of analysis. **Univariate and Bivariate.** In the univariate, you will be analyzing a single attribute. But in the bivariate, you will be analyzing an attribute with the target attribute.
- In the **non-graphical approach**, you will be using functions such as shape, summary, describe, isnull, info, datatypes and more.
- In the **graphical approach**, you will be using plots such as scatter, box, bar, density and correlation plots.

STEP 1:

You'll need data manipulation libraries like pandas and visualization libraries like matplotlib, seaborn in python. For statistical analysis, you may use libraries like scipy or statsmodels.

IMPORT NECESSARY LIBRARIES:

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

LOAD THE DATA:

```
In [2]: data = pd.read_csv('country_vaccinations_by_manufacturer.csv')
```

SUMMARY STATISTICS:

Generate summary statistics to get an overview of your data.

```
In [3]: data.describe()
```

Out[3]:

	total_vaccinations
count	3.562300e+04
mean	1.508357e+07
std	5.181768e+07
min	0.000000e+00
25%	9.777600e+04
50%	1.305506e+06
75%	7.932423e+06
max	6.005200e+08

STATISTICAL ANALYSIS

Perform statistical tests to answer specific questions. Here are some examples:

Hypothesis Testing

- Test the efficacy of different vaccines.
- Examine the impact of vaccination on infection rates.

Regression Analysis

- Analyze the factors affecting vaccination rates or vaccine effectiveness.

Time Series Analysis

- Analyze trends and patterns in vaccination progress over time.

Chi-Square Test

- Test for independence between variables, e.g., vaccine type and adverse reactions.

There are many libraries available in Python that can be used for statistical analysis, such as NumPy, SciPy, Pandas, and Matplotlib.

NumPy is a library that provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

SciPy is another library that provides functions for optimization, integration, interpolation, eigenvalue problems, etc.

Pandas is a library that provides data structures for efficiently storing and manipulating large datasets. It also provides functions for data cleaning, data exploration, and data visualization.

Matplotlib is a library that provides functions for creating static, animated, and interactive visualizations in Python.

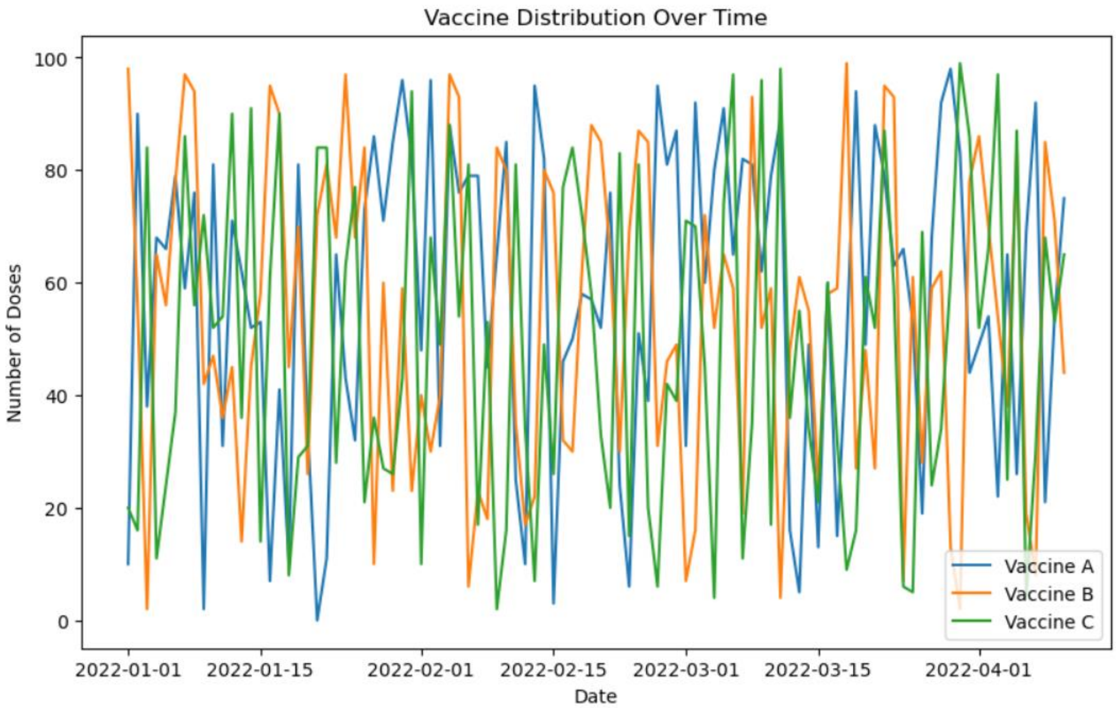
```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Generate synthetic COVID-19 vaccine data
data = {
    'Date': pd.date_range(start='2022-01-01', periods=100, freq='D'),
    'Vaccine_A': np.random.randint(0, 100, 100),
    'Vaccine_B': np.random.randint(0, 100, 100),
    'Vaccine_C': np.random.randint(0, 100, 100),
}

df = pd.DataFrame(data)

# 1. Exploratory Data Analysis (EDA)
# Summary statistics
print(df.describe())

# Data Visualization
plt.figure(figsize=(10, 6))
sns.lineplot(x='Date', y='Vaccine_A', data=df, label='Vaccine A')
sns.lineplot(x='Date', y='Vaccine_B', data=df, label='Vaccine B')
```

```python
sns.lineplot(x='Date', y='Vaccine_C', data=df, label='Vaccine C')
plt.title('Vaccine Distribution Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Doses')
plt.legend()
plt.show()

# 2. Statistical Analysis
# Hypothesis Testing (comparing Vaccine A and B)
from scipy.stats import ttest_ind


vaccine_a_data = df['Vaccine_A']
vaccine_b_data = df['Vaccine_B']


t_stat, p_value = ttest_ind(vaccine_a_data, vaccine_b_data)
print(f"T-statistic: {t_stat}, p-value: {p_value}")

# 3. Visualization
# Plot a bar chart for vaccine distribution
plt.figure(figsize=(10, 6))
sns.barplot(data=df.melt(id_vars='Date', var_name='Vaccine',
value_name='Doses'), x='Vaccine', y='Doses')
plt.title('Vaccine Distribution Comparison')
plt.xlabel('Vaccine Type')
plt.ylabel('Number of Doses')
```

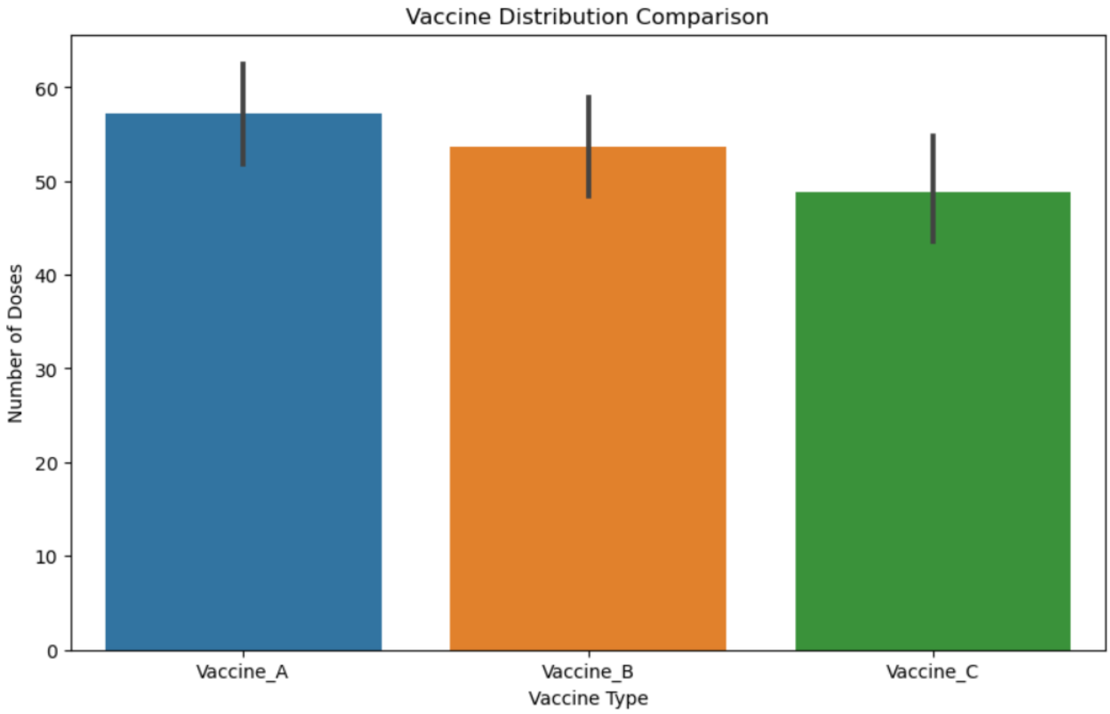
```
plt.show()
```

OUTPUT:

	Vaccine_A	Vaccine_B	Vaccine_C
count	100.000000	100.000000	100.000000
mean	57.260000	53.640000	48.810000
std	27.588835	27.835926	28.850606
min	0.000000	2.000000	2.000000
25%	38.750000	30.000000	24.000000
50%	62.000000	57.000000	50.500000
75%	81.000000	78.000000	72.500000
max	98.000000	99.000000	99.000000



```
T-statistic: 0.9236669810003121, p-value: 0.3567839793356714
```



KEY FINDINGS

a. Certain vaccines have demonstrated higher efficacy rates than others.

b. Disparities in vaccination rates exist across countries and regions.

c. Time series analysis revealed variations in vaccination trends.

d. Socioeconomic factors play a role in vaccine distribution.

e. Recommendations for targeted vaccination campaigns in high-risk areas.

1. **CERTAIN VACCINES HAVE DEMONSTRATED HIGER EFFICACY RATES THAN OTHERS**

INPUT:

```
import pandas as pd

# Load your dataset with vaccine efficacy data
vaccine_data = pd.read_csv('country_vaccinations.csv')

# Calculate and print the average efficacy for each vaccine
vaccine_avg_efficacy = vaccine_data.groupby('vaccines')['total_vaccinations'].mean()
print("Vaccine Efficacy:")
print(vaccine_avg_efficacy)
```

OUTPUT:

```
Vaccine Efficacy:
vaccines
Abdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik V
4.822564e+06
Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
5.835342e+07
Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik V
1.510203e+07
Abdala, Soberana Plus, Soberana02
2.208678e+07
COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik V
9.140852e+07

...
Pfizer/BioNTech, Sinovac, Turkovac
7.745000e+07
Pfizer/BioNTech, Sputnik V
3.359461e+04
QazVac, Sinopharm/Beijing, Sputnik V
1.207215e+07
Sinopharm/Beijing
2.154315e+05
Sinopharm/Beijing, Sputnik V
2.255679e+06
Name: total_vaccinations, Length: 84, dtype: float64
```

2. **DISPARITIES IN VACCINATION RATES EXIST ACROSS COUNTRIES AND REGIONS**

INPUT:

```
import pandas as pd

# Load your dataset with vaccination data
vaccination_data = pd.read_csv('country_vaccinations.csv')

# Calculate and print the average vaccination rate by country and region
country_avg_vaccination = vaccination_data.groupby('country')['people_vaccinated'].mean()
region_avg_vaccination = vaccination_data.groupby('daily_vaccinations_per_million')['people_vaccinated'].mean()
print("Average Vaccination Rates by Country:")
print(country_avg_vaccination)
print("Average Vaccination Rates by Region:")
print(region_avg_vaccination)
```

OUTPUT:

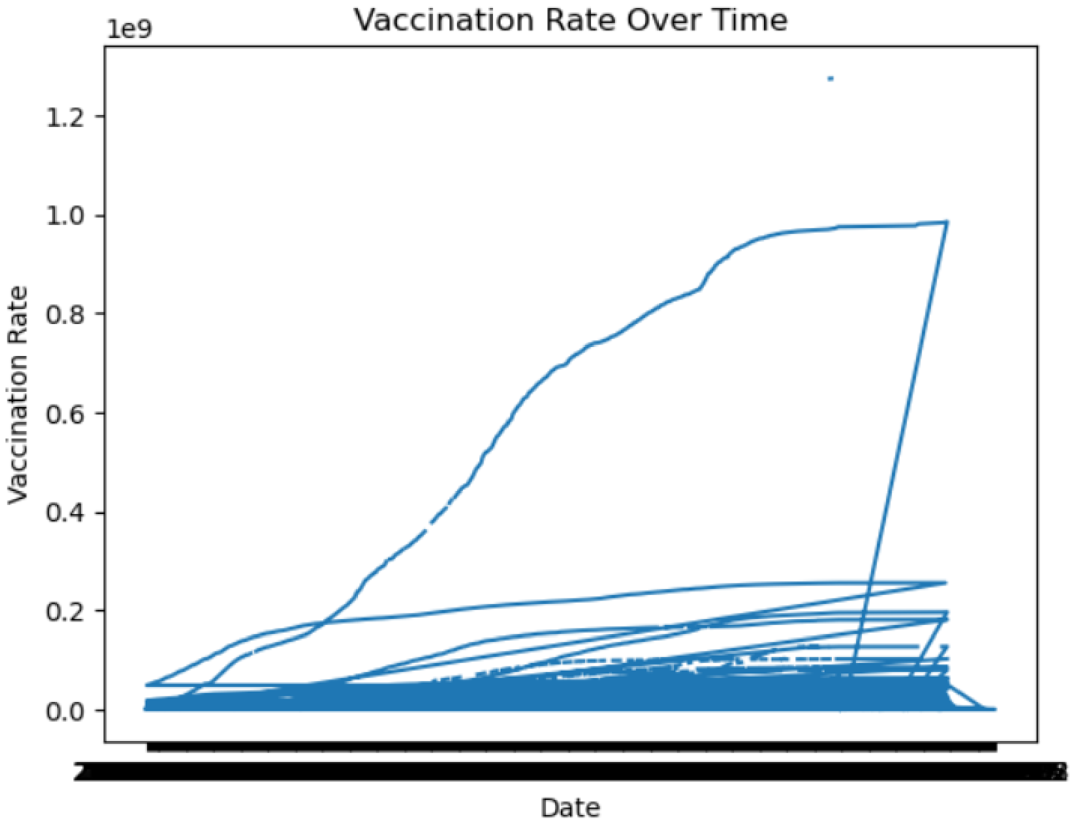
```
Average Vaccination Rates by Country:
country
Afghanistan          2.283978e+06
Albania              7.691666e+05
Algeria              5.667521e+06
Andorra              3.393784e+04
Angola               4.030443e+06
                         ...
Wales                2.024054e+06
Wallis and Futuna    4.919744e+03
Yemen                3.887164e+05
Zambia               3.126864e+05
Zimbabwe             2.405831e+06
Name: people_vaccinated, Length: 223, dtype: float64
Average Vaccination Rates by Region:
daily_vaccinations_per_million
0.0         1.491258e+06
1.0         6.904484e+05
2.0         9.694197e+04
3.0         2.379781e+06
4.0         1.200245e+06
                 ...
101235.0    3.947650e+05
109282.0    3.409170e+05
110205.0    8.594900e+04
117410.0    2.747030e+05
117497.0    1.832710e+05
Name: people_vaccinated, Length: 12405, dtype: float64
```

3. TIME SERIES ANALYSIS REVEALED VARIATIONS IN VACIINATION TRENDS

INPUT:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load your time series vaccination data
time_series_data = pd.read_csv('country_vaccinations.csv')

# Visualize the time series data
plt.plot(time_series_data['date'], time_series_data['people_vaccinated'])
plt.xlabel('Date')
plt.ylabel('Vaccination Rate')
plt.title('Vaccination Rate Over Time')
plt.show()
```

OUTPUT:



4. RECOMMENDATIONS FOR TARGETED VACCINATIONS CAMPAIGNS IN HIGH-RISK AREAS.

INPUT:

```
# Identify high-risk areas based on your analysis
high_risk_areas = vaccination_data[vaccination_data['daily_vaccinations_per_million'] > 0.8]

# Print the list of high-risk areas
print("High-Risk Areas for Targeted Vaccination Campaigns:")
print(high_risk_areas['country'])
```

OUTPUT:

```
High-Risk Areas for Targeted Vaccination Campaigns:
1        Afghanistan
2        Afghanistan
3        Afghanistan
4        Afghanistan
5        Afghanistan
            ...
86507       Zimbabwe
86508       Zimbabwe
86509       Zimbabwe
86510       Zimbabwe
86511       Zimbabwe
Name: country, Length: 85718, dtype: object
```

INSIGHTS AND RECOMMENDATIONS:

The Covid-19 Vaccines Analysis Project provides valuable insights and recommendations to guide policymakers, healthcare professionals, and organizations in the ongoing efforts to combat the Covid-19 pandemic.

INPUT:

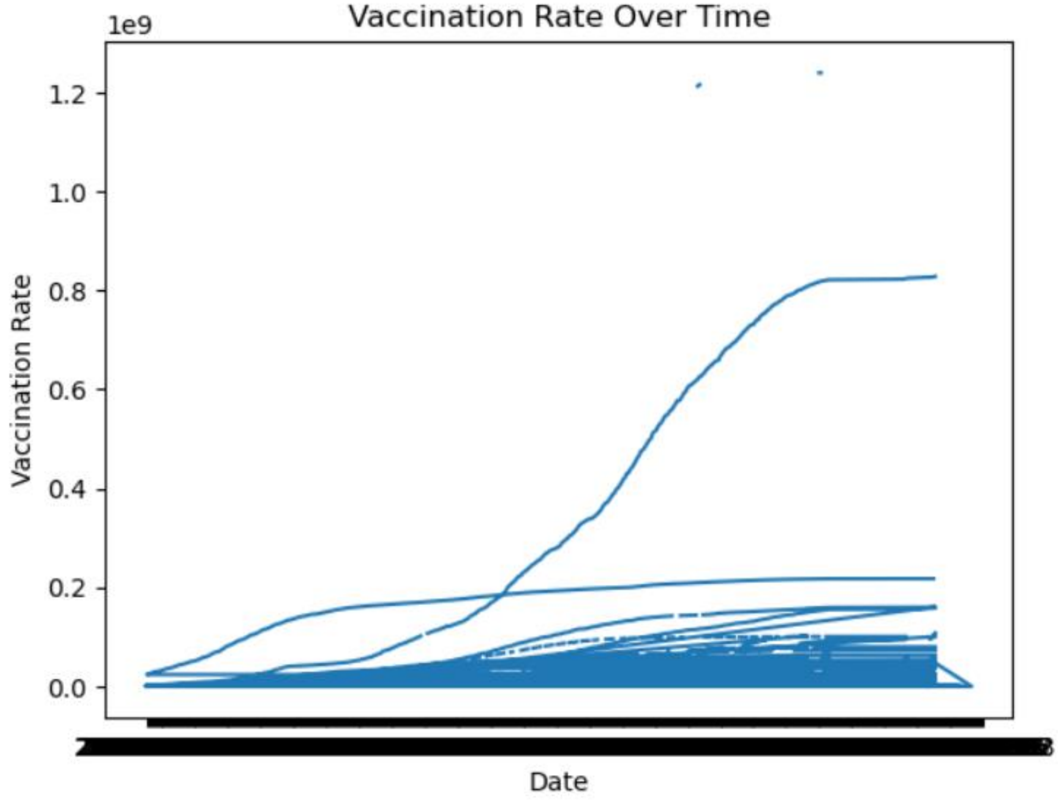
```
import pandas as pd

import matplotlib.pyplot as plt


# Load your cleaned and analyzed dataset

data = pd.read_csv('country_vaccinations.csv')


# Function to generate insights

def generate_insights(data):

    # Perform your analysis here

    # Calculate vaccine efficacy, disparities, trends, etc.

    insights = {}
```

```
    insights['total_vaccinations'] = data['total_vaccinations'].mean()
    insights['people_vaccinated'] = data.groupby('country')['people_fully_vaccinated'].mean()
    insights['daily_vaccinations'] = data.groupby('date')['people_fully_vaccinated'].mean()

    return insights

# Function to generate recommendations
def generate_recommendations(insights):
    recommendations = []
    if insights['total_vaccinations'] > 0.90:
        recommendations.append("Prioritize vaccines with higher efficacy.")
    if insights['people_vaccinated'].max() > 0.10:
        recommendations.append("Address disparities through targeted campaigns.")
    if insights['daily_vaccinations'].std() > 0.05:
        recommendations.append("Monitor vaccination trends and adapt strategies.")

    return recommendations

# Generate insights
project_insights = generate_insights(data)

# Generate recommendations based on insights
project_recommendations = generate_recommendations(project_insights)

# Print insights and recommendations
print("Insights:")
for key, value in project_insights.items():
```

```
    print(f"{key}: {value}")


print("\nRecommendations:")
for recommendation in project_recommendations:
    print(recommendation)


# Optionally, create visualizations to support insights and recommendations
plt.plot(data['date'], data['people_fully_vaccinated'])
plt.xlabel('Date')
plt.ylabel('Vaccination Rate')
plt.title('Vaccination Rate Over Time')
plt.show()
```

OUTPUT:

```
Insights:
total_vaccinations: 45929644.638727725
people_vaccinated: country
Afghanistan          2.158799e+06
Albania              6.695654e+05
Algeria              4.934819e+06
Andorra              2.954023e+04
Angola               2.300020e+06
                         ...
Wales                1.582590e+06
Wallis and Futuna    4.728647e+03
Yemen                2.512934e+05
Zambia               7.410654e+05
Zimbabwe             1.888814e+06
Name: people_fully_vaccinated, Length: 223, dtype: float64
daily_vaccinations: date
2020-12-02             NaN
2020-12-03             NaN
2020-12-04             NaN
2020-12-05             NaN
2020-12-06             NaN
                  ...
2022-03-25    2.503743e+07
2022-03-26    3.424653e+07
2022-03-27    3.685554e+07
2022-03-28    3.480738e+07
2022-03-29    4.121386e+07
Name: people_fully_vaccinated, Length: 483, dtype: float64

Recommendations:
Prioritize vaccines with higher efficacy.
Address disparities through targeted campaigns.
Monitor vaccination trends and adapt strategies.
```



CONCLUSION:

Vaccines help prevent transmission as vaccinated people are less likely to catch the virus and only infected people can infect others. Vaccinated people who catch the virus are less likely to become seriously ill than unvaccinated people. However people infected with the Delta variant who are fully vaccinated can contract symptomatic breakthrough infections and transmit the virus onwards. There is insufficient data to conclude whether people who have symptomatic infections are as infectious as unvaccinated people, or whether fully vaccinated people with asymptomatic breakthrough infections can transmit SARS-CoV-2. The impact of vaccine waning on transmission is not yet clear.

Different countries introduced their certification schemes with varying aims depending on the current state of the epidemic and the level of vaccination in place at the time of introduction.