COMPREHENSIVE ANALYSIS OF COVID-19 VACCINATION DATA

1. Dinesh K
2. Ezhil Oviya C
3. Geetika TK
4. Gogilarasan S
5. Nijantha Nathan M

In partial fulfillment for the award of the degree
Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY, CHENNAI 600025

NOVEMBER 2023

COVID-19 VACCINE ANALYSIS

Phase 5

PROJECT TITLE:

COMPREHENSIVE ANALYSIS OF COVID-19 VACCINATION DATA:

Enhancing deployment strategies for optimal public health impact

TEAM MEMBERS:

Dinesh K
- Email: dineshreddykonda@gmail.com

Ezhil Oviya C
- Email: oviyagarcia141@gmail.com

Geetika TK
- Email: geetikatk@gmail.com

Gogilarasan S
- Email: gogilarasan@gmail.com

Nijantha Nathan M
- Email: nijanthanathan72@gmail.com

Vaccination Progress in the World

What is the COVID-19/Coronavirus?

- According to the CDC, "coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness."
- Continuing, "the best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face."
- Stay informed about COVID-19. Practice social distancing and healthy practices (washing hands). Educate your friends and family on the dangers of COVID-19.
- Yes, there's currently a vaccine out, but the vaccine will take quite a while to distribute to everyone. Thus, I encourage everyone reading to be patient about COVID so we can unite together and minimize the catastrophic damage the coronavirus has inflicted.

So what role do vaccines play?

- How well it works: 95% efficacy in preventing COVID-19 in those without prior infection. In clinical trials, the vaccine was 100% effective at preventing severe disease.
- In early May, the Pfizer-BioNTech vaccine was found to be more than 95% effective against severe disease or death from the Alpha variant (first detected in the United Kingdom) and the Beta variant (first identified in South Africa) in two studies based on real-world use of the vaccine. As far as the Delta variant (first seen in India), two studies reported by Public Health England that have not yet been peer reviewed showed that full vaccination (after two doses) is 88% effective against symptomatic disease and 96% effective against hospitalization.

Bottom line: Vaccines are **very** effective, so if you haven't already and have the ability to, please get vaccinated!

Abstract:

The primary objective of this project is to conduct an extensive analysis of COVID-19 vaccine data, prioritizing the examination of vaccine efficacy, distribution patterns, and adverse effects. The ultimate aim is to furnish valuable insights to policymakers and healthcare organizations, facilitating the refinement and optimization of vaccine deployment strategies. This multifaceted project encompasses key stages including data collection, meticulous data preprocessing, in-depth exploratory data analysis, rigorous statistical examination, and effective data visualization.

Problem Statement:

The challenge at hand is to conduct an exhaustive analysis of COVID-19 vaccine data, focusing on vaccine efficacy, distribution, and adverse effects. This endeavor is imperative in order to provide tailored insights to policymakers and healthcare organizations. These insights are crucial for enhancing the precision and effectiveness of vaccine deployment strategies. The project methodology is designed to address these critical aspects comprehensively. (IBM Cognos)

Project Design:

1. Data Collection:

- Source COVID-19 vaccine data from reputable institutions including health organizations, government databases, and peer-reviewed research publications. Ensuring the highest data quality is paramount.

2. Data Preprocessing:

- Execute rigorous data cleaning and preprocessing protocols. This involves managing missing values and standardizing formats, ensuring data integrity and accuracy.

3. Exploratory Data Analysis (EDA):

- Employ sophisticated techniques for EDA to reveal underlying trends, patterns, and potential outliers in the data. This phase is pivotal for understanding the dataset's nuances.

4. Statistical Analysis:

- Implement advanced statistical tests to assess vaccine efficacy, adverse effects, and distribution trends across diverse populations. This forms the backbone of evidence-based decision-making.

5. Visualization:

- Leverage a diverse array of visualization techniques, including bar plots, line charts, and heatmaps, to effectively communicate key findings and insights.

Insights and Recommendations:

The culmination of this project will yield actionable insights and recommendations based on the analysis. These insights will serve as a strategic guide for policymakers and healthcare organizations in their efforts to optimize vaccine deployment strategies. By tailoring deployment approaches to specific demographics and regions, the goal is to maximize the impact of vaccination efforts.

Conclusion:

The Comprehensive Analysis of COVID-19 Vaccination Data project represents a crucial step towards refining vaccine deployment strategies. By rigorously examining vaccine efficacy, distribution, and adverse effects, this project provides a robust foundation for informed decision-making. The insights garnered will play a pivotal role in ensuring that vaccines reach those who need them most, ultimately contributing to a safer and healthier global community.

Tendative Timeline:

Week 1-2: Problem Definition and Design Thinking

- Understand the project scope and requirements.
- Create a detailed project proposal, including problem statement and design approach.
- Compile a list of reputable sources for COVID-19 vaccine data.
- Begin data collection process from identified sources.

Week 3-4: Data Collection and Preprocessing

- Continue data collection, ensuring data quality and reliability.
- Preliminary data cleaning and initial preprocessing.

Week 5: Exploratory Data Analysis (EDA)

- Begin EDA, identifying trends, patterns, and potential outliers.

Week 6: Statistical Analysis

- Initiate statistical tests to analyze vaccine efficacy, distribution, and adverse effects.

Week 7: Visualization

- Create initial visualizations to represent key findings and insights from the analysis.

Week 8: Insights and Recommendations (Part 1)

- Summarize initial insights and recommendations based on the analysis.

Week 9: Insights and Recommendations (Part 2)

- Conduct deeper analyses and refine recommendations.

Week 10: Documentation and Presentation (Part 1)

- Document the project methodology, results, and code.
- Begin preparing the project presentation.

Week 11: Documentation and Presentation (Part 2)

- Finalize documentation and presentation materials.
- Conduct a preliminary review to ensure completeness.

Week 12: Finalization and Submission

- Conduct a final review of the entire project for accuracy and completeness.
- Submit the project along with all documentation and presentation materials.

Overview:

Innovation in the Comprehensive Analysis of COVID-19 Vaccination Data project involves a transformative approach that harnesses advanced technologies, interdisciplinary collaboration, and ethical considerations to address the challenges of vaccine deployment. The goal is to provide actionable insights and recommendations to policymakers and healthcare organizations, ensuring precision, equity, and effectiveness in vaccine distribution strategies. Through cutting-edge data collection, preprocessing, exploratory data analysis, statistical analysis, immersive visualization, and global collaboration, the project aims to revolutionize the way COVID-19 vaccine data is analyzed and utilized for informed decision-making.

Innovation Steps:

1. Advanced Data Collection:

- Utilize web scraping tools and APIs for real-time data from diverse, global sources.
- Foster collaborations with research institutions for enriched, comprehensive datasets.

2. Enhanced Data Preprocessing:

- Implement AI-driven algorithms for automated handling of missing values, outliers, and inconsistencies.
- Employ dynamic standardization techniques adaptable to evolving data formats.

3. Innovative Exploratory Data Analysis (EDA):

- Develop interactive visualization dashboards using tools like Tableau or Power BI.
- Integrate predictive modeling techniques for proactive vaccine demand forecasting.

4. Cutting-edge Statistical Analysis:

- Implement machine learning algorithms (Random Forest, Neural Networks) for predictive modeling.
- Integrate genomic data analysis to identify vaccine-resistant strains.

5. Interactive and Immersive Visualization:

- Develop Augmented Reality (AR) applications for real-world vaccine distribution simulations.
- Conduct Virtual Reality (VR) workshops for immersive understanding of complex data insights.

6. Actionable Insights and Recommendations:

- Apply Natural Language Processing (NLP) algorithms to extract insights from scientific literature and policy documents.
- Establish a continuous monitoring system and feedback loop for adaptive strategies.

7. Ethical and Inclusive Approach:

- Develop an ethical AI framework to ensure responsible and unbiased use of AI algorithms.

- Consider socio-economic and cultural factors for inclusive and accessible vaccine deployment.

8. Capacity Building and Training:

- Organize training programs for healthcare professionals and policymakers on data-driven decision-making.
- Facilitate knowledge transfer workshops for exchange of expertise and best practices.

9. Global Collaboration and Knowledge Sharing:

- Participate in international forums and conferences to share findings and learn from global initiatives.
- Foster a collaborative approach, leveraging global expertise to address challenges in COVID-19 vaccination efforts.

This innovative approach not only refines vaccine deployment strategies but also contributes significantly to creating a safer and healthier global community, leveraging the power of advanced technology, collaboration, and ethical considerations.

Technologies Utilized:

Machine Learning Framework: Python

Description: Python, a versatile and widely adopted programming language, serves as the foundation for our machine learning endeavors. Its extensive library ecosystem provides robust tools for data analysis, modeling, and visualization.

Collaborative Workspace: Google Colab

Description: Google Colab provides a cloud-based environment conducive to collaborative work on data-driven projects. With its seamless integration with Google

Drive and access to GPU resources, it enhances our capacity for advanced computations and modeling.

Data Source and Management: Kaggle

Description: Kaggle, a renowned platform for data science enthusiasts, furnishes us with a diverse array of high-quality datasets. Its user-friendly interface and comprehensive dataset repository facilitate efficient data acquisition and management for our analyses.

Covid Vaccine Analysis

```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px


plt.rc('font', size=10)
%matplotlib inline


# import dataset from CSV
vac = '../country_vaccinations.csv'
manu = '../country_vaccinations_by_manufacturer.csv'
df_vac = pd.read_csv(vac, parse_dates= ['date'])
df_manu = pd.read_csv(manu, parse_dates = [])
df_manu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35623 entries, 0 to 35622
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   location            35623 non-null  object
 1   date                35623 non-null  object
 2   vaccine             35623 non-null  object
 3   total_vaccinations  35623 non-null  int64
dtypes: int64(1), object(3)
memory usage: 1.1+ MB
```

```
df_vac.tail(5)
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_va
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	

```
'total_vaccinations',
```

```
df_manu.head()
```

	location	date	vaccine	total_vaccinations

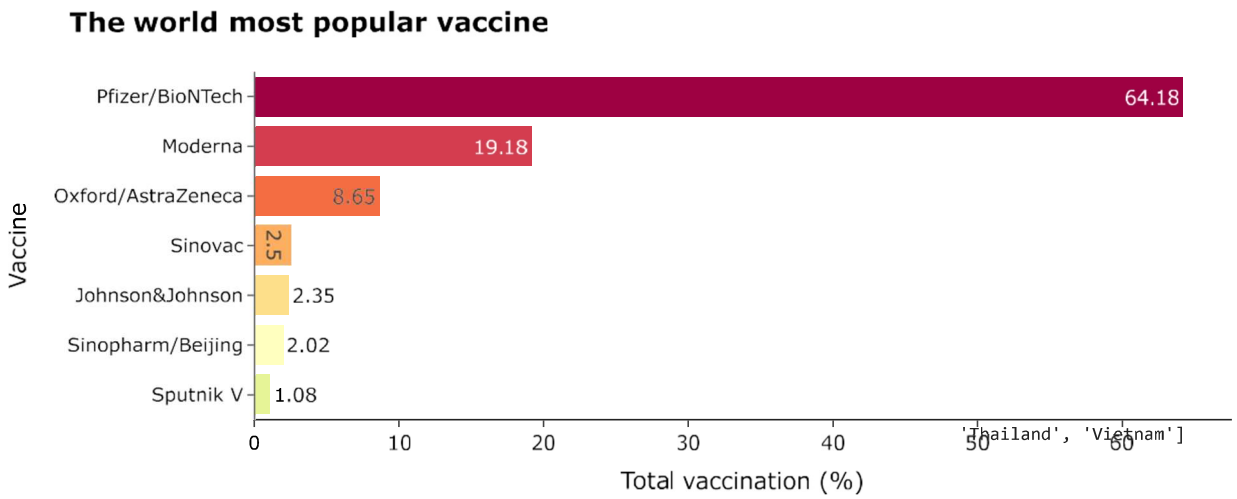
0	Argentina	2020-12-29	Moderna	2

0

```
<ipython-input-5-ba5b9c3bd4be>:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future ve
  most_vac = df_manu.groupby(['vaccine'])[['location','date', 'total_vaccinations']].sum().sort_values(by = 'total_vaccinations', ascend
```

	vaccine	total_vaccinations	Total_vac_per_million	Percent_of_total_vac
	Pfizer/BioNTech	344835955037	344835.96	64.18
1	Moderna	103072147621	103072.15	19.18
2	Oxford/AstraZeneca	46451509497	46451.51	8.65
3	Sinovac	13407163275	13407.16	2.50
4	Johnson&Johnson	12611375881	12611.38	2.35
5	Sinopharm/Beijing	10877006517	10877.01	2.02
6	Sputnik V	5787343199	5787.34	1.08
7	CanSino	271397675	271.40	0.05
8	Novavax	8268113	8.27	0.00
9	Covaxin	3572	0.00	0.00

```
# Let's plot this for easy visualization
fig = px.bar(most_vac[:7], x="Percent_of_total_vac", y="vaccine", template = 'simple_white',
             width=1000, height=400 , orientation = 'h', color = "vaccine",
            color_discrete_sequence=px.colors.diverging.Spectral, text_auto=True,
            labels=dict(Percent_of_total_vac ="Total vaccination (%)", vaccine="Vaccine")).update_xaxes(categoryorder = "total descending")
fig.update_layout(
    title="<b>The world most popular vaccine</b>",
    font=dict(
        size=14,
        color="black"), showlegend = False
)
fig.show()
```



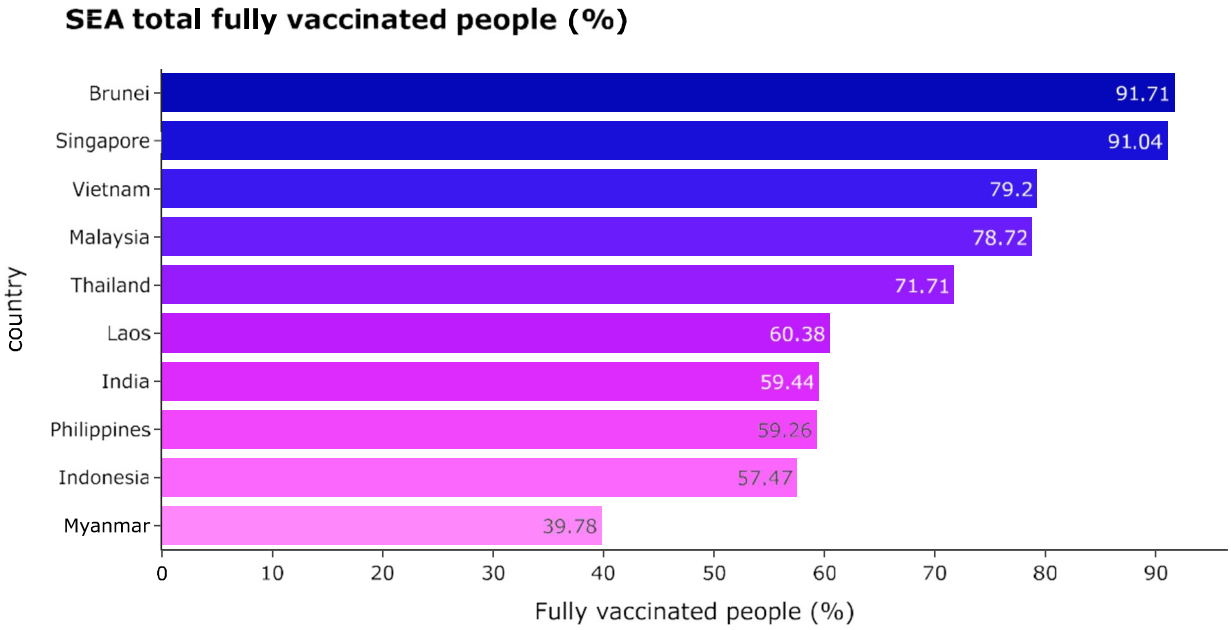

```
# The list of SEA countries which have the highest percentage of fully vaccinated people
sea = ['Brunei', 'India', 'Indonesia', 'Laos', 'Malaysia', 'Myanmar', 'Philippines', 'Singapore',
df_vac_sea = df_vac[df_vac['country'].isin(sea)]
df_vac_sea_group = df_vac_sea.groupby(['country'])[['date','people_fully_vaccinated_per_hundred']].max().sort_values(by = 'people_fully_vacci
df_vac_sea_group
```

country	date	people_fully_vaccinated_per_hundred
Brunei	2022-03-18	91.71

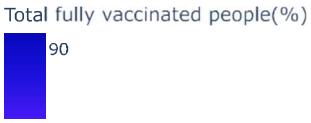
```
df_vac_sea_group['iso_alpha'] = ["BRN","SGP","VNM", "MYS","THA","LAO","IND", "PHL", "IDN","MMR"]
```

Vietnam	2022-03-22	79.20

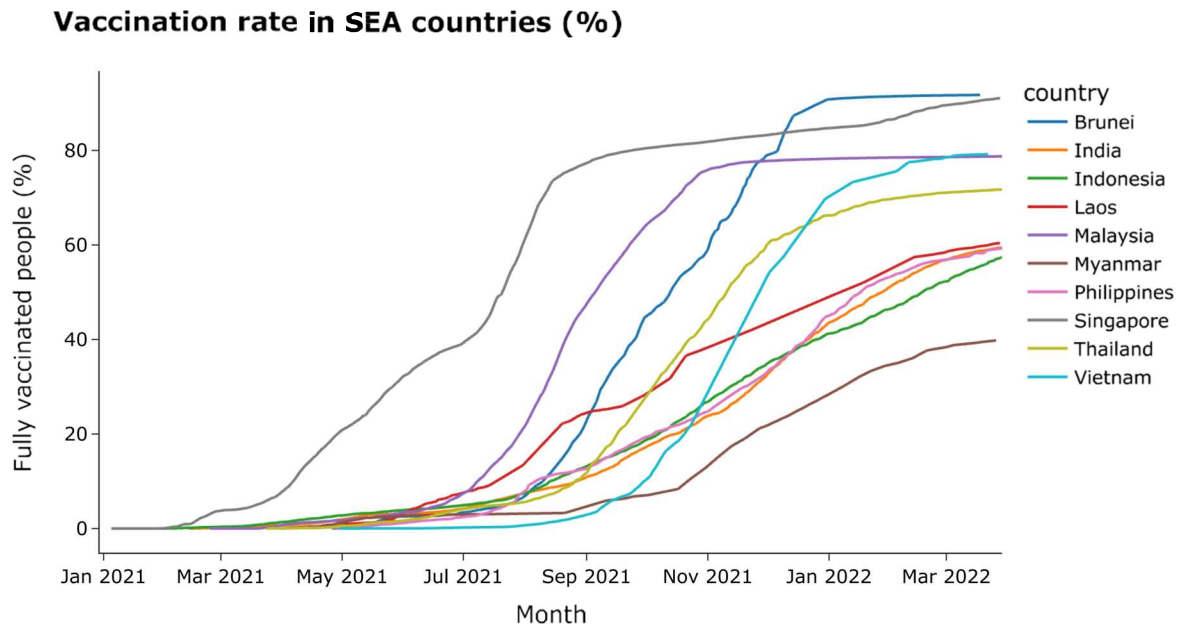
```
fig = px.bar(df_vac_sea_group, x= "people_fully_vaccinated_per_hundred", y= df_vac_sea_group.index, template = 'simple_white',
             width=1000, height=500 , orientation = 'h', color = df_vac_sea_group.index,
            color_discrete_sequence=px.colors.sequential.Plotly3, text_auto=True,
            labels=dict(people_fully_vaccinated_per_hundred ="Fully vaccinated people (%)")).update_xaxes(categoryorder = "total descending"
fig.update_layout(
    title="<b>SEA total fully vaccinated people (%)</b>",
    font=dict(
        size=14,
        color="black"),
    showlegend = False
)
fig.show()
```




```
fig = px.choropleth(df_vac_sea_group, locations="iso_alpha",
                    color="people_fully_vaccinated_per_hundred",
                    width=900, height=600,
                    hover_name=df_vac_sea_group.index, # column to add to hover information
                    color_continuous_scale=px.colors.sequential.Plotly3[::-1],
                    labels=dict(people_fully_vaccinated_per_hundred ="Total fully vaccinated people(%)"))

fig.update_geos(fitbounds="locations", visible=True)
fig.update_layout(height=400,margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```




```
fig = px.line(df_vac_sea, x = 'date', y='people_fully_vaccinated_per_hundred', color = 'country', template="simple_white",
              width = 900, height = 500)

fig.update_layout(
    title="<b>Vaccination rate in SEA countries (%)</b>",
    xaxis_title="Month",
    yaxis_title="Fully vaccinated people (%)",
    font=dict(
        size=14,
        color="black")
)
fig.update_traces(connectgaps=True)
fig.show()
```




```
# check NULL values in daily_vaccination data
df_vac_daily = df_vac_sea[['country','date', 'daily_vaccinations_per_million']]
top = ['Brunei', 'Singapore', 'Vietnam']
df_vac_daily = df_vac_daily[df_vac_daily['country'].isin(top)]
df_vac_daily.daily_vaccinations_per_million.isna().sum()
df_vac_daily[df_vac_daily['daily_vaccinations_per_million'].isna()]
```

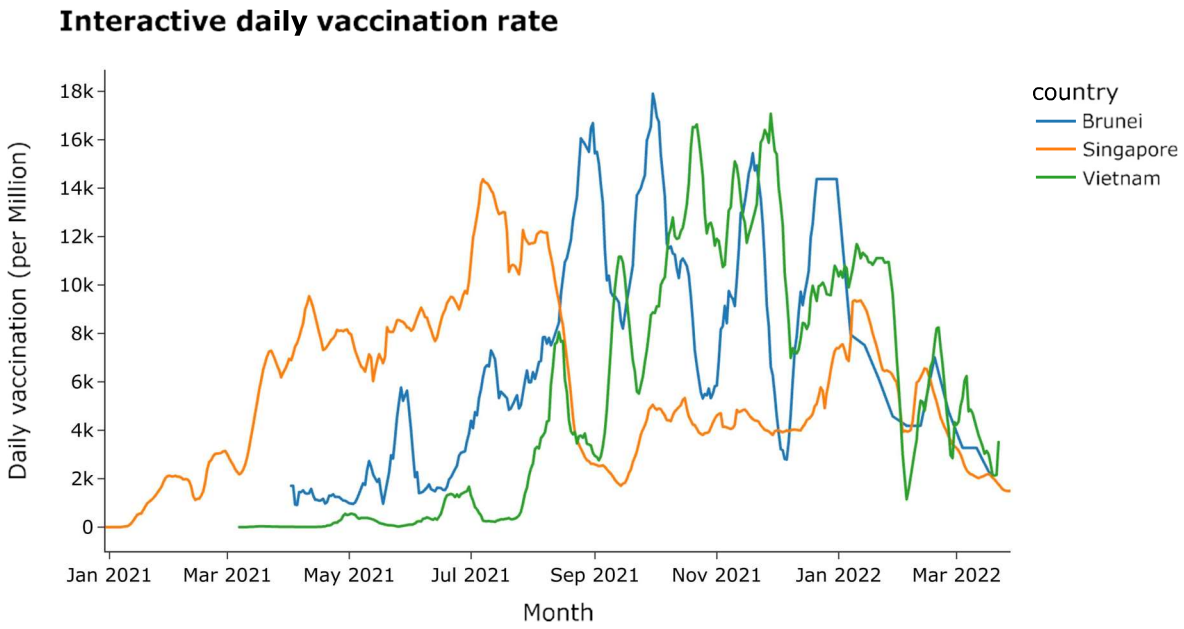
	country	date	daily_vaccinations_per_million
11395	Brunei	2021-04-02	NaN
69775	Singapore	2020-12-30	NaN
84250	Vietnam	2021-03-07	NaN

```
# Fill NULL values with back values close to that NULL
df_vac_daily['daily_vaccinations_per_million'].fillna(method = 'bfill', inplace =True)
df_vac_daily.daily_vaccinations_per_million.isna().sum()
```

```
0
```

```
# Creating the Figure instance
fig = px.line(df_vac_daily, x= 'date' , y= 'daily_vaccinations_per_million', color = 'country', template="simple_white",
              width = 900, height = 500)

fig.update_layout(
    title="<b>Interactive daily vaccination  rate</b>",
```

```
    xaxis_title="Month",
    yaxis_title="Daily vaccination (per Million)",
    font=dict(
        size=14,
        color="black")
)
fig.show()
```




```
!pip install --upgrade pip
```

```
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (23.1.2)
Collecting pip
  Downloading pip-23.3-py3-none-any.whl (2.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 12.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.1.2
    Uninstalling pip-23.1.2:
      Successfully uninstalled pip-23.1.2
Successfully installed pip-23.3
```

```
!pip install pystan~=2.14
```

```
Requirement already satisfied: pystan~=2.14 in /usr/local/lib/python3.10/dist-packages (2.19.1.1)
Requirement already satisfied: Cython!=0.25.1,>=0.22 in /usr/local/lib/python3.10/dist-packages (from pystan~=2.14) (3.0.3)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.10/dist-packages (from pystan~=2.14) (1.23.5)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It i
```

```
# Fill NULL using interpolate
df_in = df_vac_sea[df_vac_sea['country'] == 'Indonesia'][['date','people_fully_vaccinated_per_hundred']]
df_in['people_fully_vaccinated_per_hundred']= df_in['people_fully_vaccinated_per_hundred'].interpolate()
df_in = df_in.rename(columns={'people_fully_vaccinated_per_hundred': 'y', 'date':'ds'})
```

New Section

Phase 4

INTRODUCTION

In this phase, we will continue advancing our project by performing the following essential tasks:

- *Exploratory Data Analysis (EDA)*
- *Statistical Analysis*
- *Data Visualization*

These critical processes are carried out on the dataset we have already collected and pre-processed. During this stage, we will delve deeper into our project by conducting various analyses on the prepared dataset and using visualization techniques to present the findings for a clearer and more insightful understanding. This step marks a significant milestone in our project, helping us gain valuable insights from the data at our disposal.

DATA AQUISTION:

Collecting data for COVID-19 vaccine analysis is a crucial aspect of comprehending the effectiveness and impact of vaccination campaigns, enabling us to conduct a thorough analysis. This process entails gathering a wide range of information related to vaccination efforts, including:

- Country-wise total vaccinations administered
- Number of people vaccinated
- Adverse events reported
- Distribution logistics and so on..

The primary source of data for our project is the Kaggle dataset, which provides comprehensive information related to global COVID-19 vaccination progress. You can access the dataset through the following link:

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

DATA PREPARATION:

Once we have collected reliable data, the next step is to clean and prepare the data for analysis, a crucial process known as Data Pre-processing. It's important to note that these pre-processing steps have already been performed and well-documented in the previous phase. Now, we will build upon this foundation and proceed with the subsequent stages of our analysis.

EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) is an essential initial step in data analysis. It is the method of studying and exploring data set to recognizetheir traits, discover patterns, locate outliers, and identify relationships between variables.

EDA is essential for getting a clear picture of the data which is useful in subsequent decision-making and can be performed using various statistical and graphical techniques. It involves multiple iterations and proves especially beneficial in prepping data for machine learning or statistical modeling. It is performed in the project as follows,

Initially, we take a look at the different types of data we have in our dataset.

```
df_vaccination.dtypes # take a look of the data types that we dealing with (precisely the date column)
```

The output is,

```
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
```

Note that the “date” field is of object datatype and so for better analysis, it is converted to datetime format by,

```
In [4]: df_vaccination['date'] = pd.to_datetime(df_vaccination['date'], format="%Y-%m-%d")
        # to_datetime is a pandas method which helps to convert datetime string into pandas datetime object

In [5]: df_vaccination.dtypes # check our new data types after converting date(column) into datetime64[ns] by using pd.to_datetime()

Out[5]: country                                object
        iso_code                               object
        date                           datetime64[ns]
        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
```

Now it can be seen that the datatype has been changed which makes it easier to work with it.

After which, various other fields are being examined to make sure we have the perfect set of data to analyze.

```
df_vaccination.isnull().sum()

#There are no empty rows for country, iso_code or date columns.
```

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                 42905
people_vaccinated                  45218
people_fully_vaccinated            47710
daily_vaccinations_raw             51150
daily_vaccinations                   299
total_vaccinations_per_hundred     42905
people_vaccinated_per_hundred      45218
people_fully_vaccinated_per_hundred 47710
daily_vaccinations_per_million       299
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

```
# General Overview of the calculations in data

df_vaccination.describe()
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccina
count	43,607.00	41,294.00	38,802.00	35,362.00	86,213.00	43,607.00	
mean	45,929,644.64	17,705,077.79	14,138,299.85	270,599.58	131,305.49	80.19	
std	224,600,360.18	70,787,311.50	57,139,201.72	1,212,426.60	768,238.77	67.91	
min	0.00	0.00	1.00	0.00	0.00	0.00	
25%	526,410.00	349,464.25	243,962.25	4,668.00	900.00	16.05	
50%	3,590,096.00	2,187,310.50	1,722,140.50	25,309.00	7,343.00	67.52	
75%	17,012,303.50	9,152,519.75	7,559,869.50	123,492.50	44,098.00	132.74	
max	3,263,129,000.00	1,275,541,000.00	1,240,777,000.00	24,741,000.00	22,424,286.00	345.37	

Followed by,

It is not always necessary that all the fields/attributes in the collected dataset is/are useful for our analysis.

Therefore, the fields “source_name”,”source_website” and “vaccine_columns” are not required and hence are dropped for more efficient analysis.

```
#drop the source_name,source_website and vaccine columns

df_vaccine_country = df_vaccination.drop(['source_name','source_website','vaccines'],axis=1)
df_vaccine_country.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_
0	Afghanistan	AFG	2021-02-22	0.00	0.00	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1,367.00	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1,367.00	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1,367.00	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1,367.00	

All the Nan values are then replaced by 0 to make calculations easier. From the screenshot below, it can be seen that the sum of all null values in every column is 0.

```
# convert Date column to date type and fill na values with 0 for calculation

df_vaccine_country["date"] = pd.to_datetime(df_vaccine_country["date"], format = '%Y-%m-%d')

df_vaccine_country = df_vaccine_country.replace([np.inf, -np.inf], np.nan)
df_vaccine_country = df_vaccine_country.fillna(0)
df_vaccine_country.isnull().sum()
```

```
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
dtype: int64
```

Once the dataset is prepared and ready for analysis, statistical analysis is performed on it.

STATISTICAL ANALYSIS

In statistical analysis, the total, average, maximum and minimum of different vaccinations status by country is calculated.

```
#########STATISTICAL ANALYSIS##########
#Function to find total, average, maximum  and minimum of different vaccinations status by country

def vaccination_country(col_name,func_name):

    '''
    Function that requires vaccination column name, and sum/mean/max/min function name as string arguments.
    '''
    if func_name == 'sum':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .sum()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                            )

    elif func_name == 'mean':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .mean()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                               )
    elif func_name == 'max':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .max()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                               )
    elif func_name == 'min':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                .min()
                                .sort_values(by=col_name,ascending= False)
                                .reset_index()
                               )
```

The code snippet of function for finding country with maximum and minimum daily vaccinations is,

```
#Function for Country with maximum and minimum daily vaccinations
def daily_vaccination_country(col_name,func_name):

    '''
    A function that requires daily_vaccination column and max/min function name as string arguments.
    '''

    daily_vaccination = (df_vaccine_country
                                .pivot_table(index='country',columns='date',values=col_name)
                                  )
    if func_name == 'max':

        daily_vaccination['Highest Daily Vaccination'] = daily_vaccination.max(axis=1)
        daily_vaccination['Date - Highest Daily Vaccination'] = daily_vaccination.idxmax(axis=1)
        daily_vaccination.sort_values(by='Highest Daily Vaccination',ascending=False,inplace=True)
        daily_vaccination.rename_axis('',axis=1,inplace=True)

        return daily_vaccination[['Highest Daily Vaccination','Date - Highest Daily Vaccination']].reset_index()

    elif func_name == 'min':

        daily_vaccination.replace(0.00,np.nan,inplace=True)
        daily_vaccination['Lowest Daily Vaccination'] = daily_vaccination.min(axis=1)
        daily_vaccination['Date - Lowest Daily Vaccination'] = daily_vaccination.idxmin(axis=1)
        daily_vaccination.sort_values(by='Lowest Daily Vaccination',ascending=False,inplace=True)
        daily_vaccination.rename_axis('',axis=1,inplace=True)

        return daily_vaccination[['Lowest Daily Vaccination','Date - Lowest Daily Vaccination']].reset_index()
```

Finally, calculating the highest and lowest daily vaccination and the respective dates.

```
#Calculating highest and lowest daily vaccination and the respective dates.
highest_daily_vaccination = daily_vaccination_country('daily_vaccinations','max')
lowest_daily_vaccination = daily_vaccination_country('daily_vaccinations','min')
```

Once all necessary aspects are calculated, it now time for visualization i.e., representing the analyzed records graphically for better understanding of complex data patterns and relations.

VISUALIZATION

Data visualization is the use of graphical elements such as charts, graphs, and maps to represent data and information visually. The use of visualization tools provides an accessible way to see and understand trends, outliers, and patterns in data.

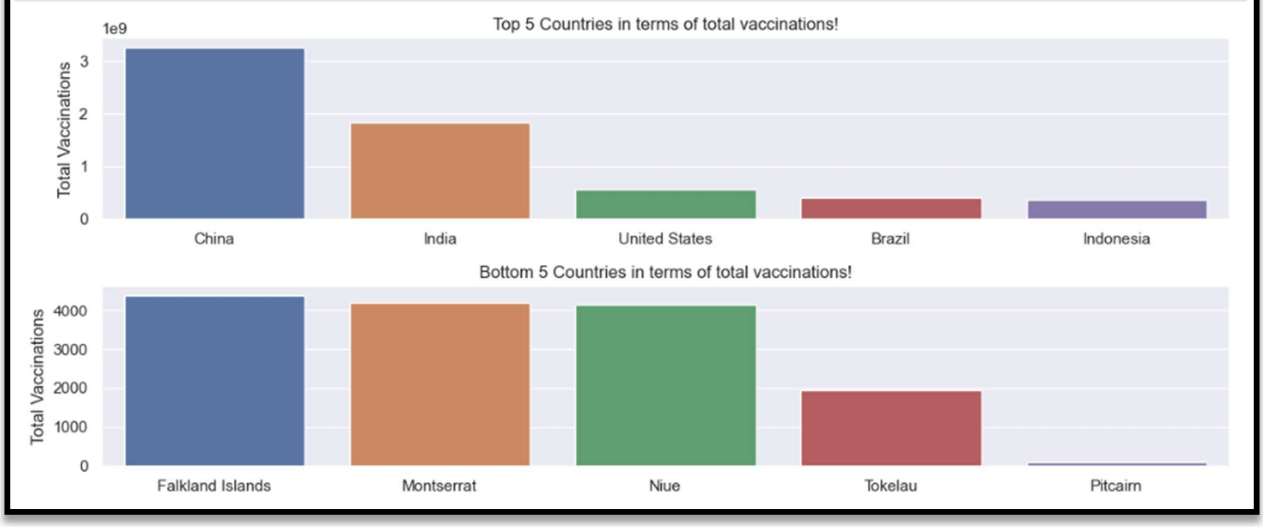
There are various techniques in data visualization. Few of them are described below,

- ***Histograms***: Plot the frequency distribution of numerical variables to identify patterns and distributions.
- ***Box Plots***: Display the distribution, central tendency, and outliers in numerical data.
- ***Scatter Plots***: Visualize relationships between two numerical variables to identify correlations or patterns.
- ***Bar Charts***: Used for categorical data to show the frequency of different categories.
- ***Heatmaps***: Display the correlation between variables using color gradients.
- ***Pair Plots***: When dealing with multiple numerical variables, pair plots help visualize relationships between them.

```
##########VISUALIZATION##########
# Calculating different vaccinations for visualizations
max_total_vaccinations = vaccination_country('total_vaccinations','max')

sum_people_vaccinated = vaccination_country('people_vaccinated','sum')
sum_people_fully_vaccinated = vaccination_country('people_fully_vaccinated','sum')

avg_total_vaccinations = vaccination_country('total_vaccinations_per_hundred','mean')
avg_people_vaccinated = vaccination_country('people_vaccinated_per_hundred','mean')
avg_people_fully_vaccinated = vaccination_country('people_fully_vaccinated_per_hundred','mean')
avg_daily_vaccinations = vaccination_country('daily_vaccinations_per_million','mean')
```

First, all required parameters are calculated using the previously created functions.

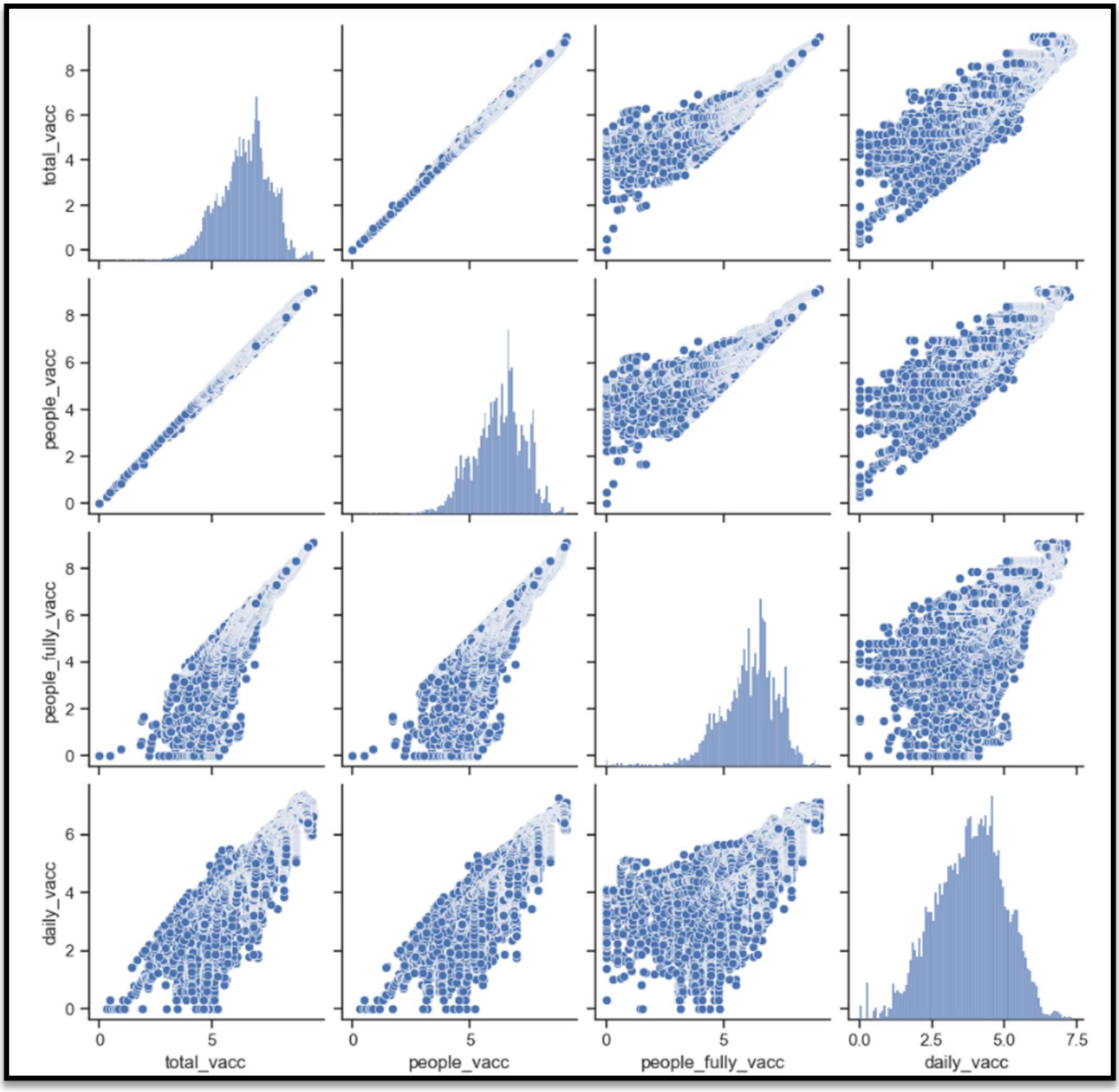
```
#Set sns theme and default figsize for all the sns visualizations.

sns.set_theme(style='whitegrid')
sns.set(rc={'figure.figsize' : (12,5)})

fig, axes  = plt.subplots(2,1)

sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.head(),ax=axes[0])
axes[0].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Top 5 Countries in terms of total vaccinations!')

sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.tail(),ax=axes[1])
axes[1].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Bottom 5 Countries in terms of total vaccinations!')


fig.tight_layout()
plt.show()
```



Then, a bar graph is used to represent the Top 5 and Bottom 5 countries in terms of total vaccinations.

```
#Plotting scatterplot matrix using Seaborn
#create dataframe with important features.
df_vaccination['total_vacc'] = np.log10(df_vaccination['total_vaccinations'])
df_vaccination['people_vacc'] = np.log10(df_vaccination['people_vaccinated'])
df_vaccination['people_fully_vacc'] = np.log10(df_vaccination['people_fully_vaccinated'])
df_vaccination['daily_vacc'] = np.log10(df_vaccination['daily_vaccinations'])

#drop the original nontransformed columns
df_vaccination = df_vaccination.drop(columns = ['total_vaccinations','people_vaccinated','people_fully_vaccinated', 'daily_vaccir


covid_features = df_vaccination[['date', 'total_vacc', 'people_vacc' , 'people_fully_vacc' , 'daily_vacc']]
sns.set_theme(style="ticks")
sns.pairplot(covid_features)
```

Here scatter plot is used for which the output is,



CONCLUSION:

At the end of this phase, the collected and prepared data has been gone through Exploratory data analysis (EDA) and Statistical analysis.And finally, visualization tools have been used to graphically represent the analyzed data which helps in deeper insights on the same.