COLLEGE CODE: 5113

APPLIED DATA SCIENCE

Project No.5- COVID -19 VACCINE ANALYSIS

BATCH MEMBERS:

1. V.SHREE KUMAR – autre-csea-01- Shreekumar0622@gmail.com

INTRODUCTION:

Analyzing COVID-19 vaccines is a critical component of the global response to the ongoing pandemic caused by the novel coronavirus, SARS-CoV-2. These vaccines have been developed at an unprecedented pace and are essential tools in controlling the spread of the virus and reducing the severity of the disease. Vaccine analysis involves a multifaceted approach that encompasses various aspects, including efficacy, safety, distribution, public acceptance, and their impact on the pandemic One of the primary aspects of analyzing COVID-19 vaccines is assessing their efficacy and effectiveness.

ABOUT THE DATA:

Where did we get the dataset?

Kaggle:

The dataset provided on Kaggle, https://www.kaggle.com/datasets/gpreda/covid-world-vaccinationprogress,

offers a valuable resource for our project aimed at forecasting covid-19 vaccine analysis.

Dataset Details:

The data (country vaccinations) contains the following information:

Country- this is the country for which the vaccination information is provided;

Country ISO Code - ISO code for the country;

Date - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;

Total number of vaccinations - this is the absolute number of total immunizations in the country;

Total number of people vaccinated - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;

Total number of people fully vaccinated - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme.

Problem Statement:

The problem statement for a COVID-19 vaccine analysis involves assessing the performance, impact, and distribution of COVID-19 vaccines. The key question is to understand how well these vaccines are working in terms of preventing infections, reducing severe illness, and achieving herd immunity. The problem may also include identifying challenges in vaccine distribution, monitoring adverse events, and ensuring equitable access to vaccines.

Design Thinking Process:

The design thinking process for a COVID-19 vaccine analysis includes several stages:

- **Empathize:** Understand the needs and concerns of various stakeholders, such as healthcare workers, policymakers, and the general public.

- **Define:** Clearly define the objectives of the analysis, including what you want to achieve with the analysis.

- **Ideate:** Generate ideas and potential solutions for addressing the challenges and questions related to COVID-19 vaccines.

- **Prototype:** Develop data collection and analysis plans, including selecting appropriate datasets and methodologies.

- **Test:** Implement the analysis and assess its validity and reliability through rigorous testing and validation.

The phases of development in a COVID-19 vaccine analysis include:

1. **Planning**: In the planning phase, the objectives, scope, and research questions are defined. Clear goals for the analysis are established, and data sources are identified. The planning phase is crucial for setting the direction and ensuring that the analysis is focused on addressing key issues related to vaccine efficacy, safety, and distribution.

2. **Data Collection:** During this phase, data from various sources, including clinical trials, real-world data, and adverse event reports, is gathered. The data collection process should be systematic and comprehensive to ensure the analysis is based on a reliable dataset.

3. **Data Preprocessing:** In this step, collected data is cleaned, transformed, and prepared for analysis. This includes addressing missing data, standardizing formats, and merging data from different sources.

4. **Analysis Techniques:** Statistical and data analysis methods are applied to the preprocessed data to assess vaccine efficacy, safety, and distribution. Techniques such as regression analysis and survival analysis may be used to derive insights.

5. **Key Findings and Insights:** The analysis phase reveals essential findings, including vaccine efficacy rates, adverse event profiles, and progress in vaccination campaigns. These insights serve as the foundation for recommendations.

6. **Recommendations:** Based on the findings, actionable recommendations are formulated. These suggestions may include strategies for vaccine distribution, safety monitoring, and public trust-building efforts. This phase guides decision-makers in optimizing vaccination campaigns.

LOADING THE DATASET:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from folium.features import Choropleth
import folium
from folium.features import Tooltip
import seaborn as sns
import scipy.stats as stats
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv("C:/Users/haris/OneDrive/Desktop/country_vaccinations(1).csv")
```

PREPROCESSING THE DATASET:

Preprocessing of data in a dataset refers to the various techniques and operations applied to the data before using it for analysis, modeling, Here's a more detailed explanation of data preprocessing within the context of a dataset:

1. Data Cleaning:

Handling Missing Values: Identify and deal with missing data, which may involve filling in missing values, removing rows with missing data, or using imputation techniques.

Dealing with Duplicates: Detect and remove duplicate records to ensure data integrity.

print(df.head(10))

```
In [34]: df = pd.read_csv("C:/Users/haris/OneDrive/Desktop/country_vaccinations(1).csv")

         print(df.head(10))
```

```
       country iso_code        date  total_vaccinations  people_vaccinated  \
0  Afghanistan      AFG  2021-02-22                 0.0                0.0
1  Afghanistan      AFG  2021-02-23                 NaN                NaN
2  Afghanistan      AFG  2021-02-24                 NaN                NaN
3  Afghanistan      AFG  2021-02-25                 NaN                NaN
4  Afghanistan      AFG  2021-02-26                 NaN                NaN
5  Afghanistan      AFG  2021-02-27                 NaN                NaN
6  Afghanistan      AFG  2021-02-28              8200.0             8200.0
7  Afghanistan      AFG  2021-03-01                 NaN                NaN
8  Afghanistan      AFG  2021-03-02                 NaN                NaN
9  Afghanistan      AFG  2021-03-03                 NaN                NaN

   people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
0                      NaN                     NaN                 NaN
1                      NaN                     NaN              1367.0
2                      NaN                     NaN              1367.0
3                      NaN                     NaN              1367.0
4                      NaN                     NaN              1367.0
5                      NaN                     NaN              1367.0
6                      NaN                     NaN              1367.0
7                      NaN                     NaN              1580.0
8                      NaN                     NaN              1794.0
9                      NaN                     NaN              2008.0
```

```
   total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
0                            0.00                           0.00
1                             NaN                            NaN
2                             NaN                            NaN
3                             NaN                            NaN
4                             NaN                            NaN
5                             NaN                            NaN
6                            0.02                           0.02
7                             NaN                            NaN
8                             NaN                            NaN
9                             NaN                            NaN

   people_fully_vaccinated_per_hundred  daily_vaccinations_per_million  \
0                                  NaN                             NaN
1                                  NaN                            34.0
2                                  NaN                            34.0
3                                  NaN                            34.0
4                                  NaN                            34.0
5                                  NaN                            34.0
6                                  NaN                            34.0
7                                  NaN                            40.0
8                                  NaN                            45.0
9                                  NaN                            50.0
```

```
                                            vaccines  \
0  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
1  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
2  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
3  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
4  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
5  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
6  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
7  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
8  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
9  Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

                source_name            source_website
0  World Health Organization  https://covid19.who.int/
1  World Health Organization  https://covid19.who.int/
2  World Health Organization  https://covid19.who.int/
3  World Health Organization  https://covid19.who.int/
4  World Health Organization  https://covid19.who.int/
5  World Health Organization  https://covid19.who.int/
6  World Health Organization  https://covid19.who.int/
7  World Health Organization  https://covid19.who.int/
8  World Health Organization  https://covid19.who.int/
9  World Health Organization  https://covid19.who.int/
```

---

print(df.columns)

```
In [35]: print(df.columns)
```

```
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')
```

print(df["country"].nunique())
print(df.isnull().sum())
df . fillna(0)

```
In [38]: print(df["country"].nunique())

         print(df.isnull().sum())
         df.fillna(0)
```

```
223
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
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	0.0	0.0	0.0	
1	Afghanistan	AFG	2021-02-23	0.0	0.0	0.0	0.0	1367.0	
2	Afghanistan	AFG	2021-02-24	0.0	0.0	0.0	0.0	1367.0	
3	Afghanistan	AFG	2021-02-25	0.0	0.0	0.0	0.0	1367.0	
4	Afghanistan	AFG	2021-02-26	0.0	0.0	0.0	0.0	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	

```
data_info= data.info()
data.fillna(0,inplace=True)
print(df.dtypes)
```

2. Data Transformation:

Feature Scaling: Normalize or standardize numerical features to bring them to a similar scale. This is important for algorithms sensitive to feature scales.

Feature Encoding: Convert categorical variables into a numerical format using techniques like one-hot encoding or label encoding.

Feature Engineering: Create new features or modify existing ones to capture relevant information and patterns in the data.

Binning: Group continuous data into bins or categories to simplify analysis.

Log Transformation: Apply logarithmic transformations to features when necessary to make their distribution more normal.

print(df['date'] == pd.to_datetime(df['date']))

```
In [37]: print(df.dtypes)
         print(df['date'] == pd.to_datetime(df['date']))
```

```
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
0        True
1        True
2        True
3        True
4        True
         ...
86507    True
86508    True
86509    True
86510    True
86511    True
```

```
data = pd.DataFrame(columns=['country', 'vaccines', 'total_vaccinations'])
splitted = df['vaccines'].str.split(',', expand=True)

df['vaccines'] = splitted[0].astype('string')
splitted = df['vaccines'].str.split('/', expand=True)
df['vaccines'] = splitted[0].astype('string')

data=pd.DataFrame(columns=['country', 'vaccines', 'Total_vaccine'])
for country in df["country"].unique():
    for vaccines in df["vaccines"].unique():
        filtered_data = df[(df['country'] == country) & (df['vaccines'] == vaccines)]
        total_count = filtered_data['total_vaccinations'].max()
        data = pd.concat([data, pd.DataFrame({'country': [country], 'vaccines': [vaccines],
      'Total_vaccine': [total_count]})], ignore_index=True)

print(data.head(10))
```

```
In [4]: data = pd.DataFrame(columns=['country', 'vaccines', 'total_vaccinations'])
        splitted = df['vaccines'].str.split(',', expand=True)

        df['vaccines'] = splitted[0].astype('string')
        splitted = df['vaccines'].str.split('/', expand=True)
        df['vaccines'] = splitted[0].astype('string')
```

```
In [5]: data=pd.DataFrame(columns=['country', 'vaccines', 'Total_vaccine'])
        for country in df["country"].unique():
            for vaccines in df["vaccines"].unique():
                filtered_data = df[(df['country'] == country) & (df['vaccines'] == vaccines)]
                total_count = filtered_data['total_vaccinations'].max()
                data = pd.concat([data, pd.DataFrame({'country': [country], 'vaccines': [vaccines], 'Total_vaccine': [total_count]})], ig
```

```
In [6]: print(data.head(10))

       country         vaccines  Total_vaccine
0  Afghanistan  Johnson&Johnson      5751015.0
1  Afghanistan           Oxford            NaN
2  Afghanistan          Moderna            NaN
3  Afghanistan          CanSino            NaN
4  Afghanistan           Pfizer            NaN
5  Afghanistan        Sinopharm            NaN
6  Afghanistan          Covaxin            NaN
7  Afghanistan           Abdala            NaN
8  Afghanistan  COVIran Barekat            NaN
9  Afghanistan           QazVac            NaN
```

3. Data Reduction:

Dimensionality Reduction: Reduce the number of features, often using techniques like Principal Component Analysis (PCA) or feature selection to select the most relevant variables.

Outlier Detection and Handling: Identify and deal with outliers, which can distort analysis and modeling results.

```
data.dropna(axis=0,inplace=True)
data.head(20)
```

```
In [39]: data.dropna(axis=0,inplace=True)
         data.head(20)
Out[39]:
```

	country	vaccines	Total_vaccine
0	Afghanistan	Johnson&Johnson	5751015.0
13	Albania	Oxford	2754244.0
25	Algeria	Oxford	13704895.0
38	Andorra	Moderna	151997.0
49	Angola	Oxford	17535411.0
61	Anguilla	Oxford	22714.0
73	Antigua and Barbuda	Oxford	125386.0
87	Argentina	CanSino	96504666.0
98	Armenia	Moderna	2088962.0
112	Aruba	Pfizer	169231.0
122	Australia	Moderna	56242913.0
132	Austria	Johnson&Johnson	18131115.0
145	Azerbaijan	Oxford	13425032.0
156	Bahamas	Johnson&Johnson	334155.0
168	Bahrain	Johnson&Johnson	3421273.0
180	Bangladesh	Johnson&Johnson	243642749.0
193	Barbados	Oxford	312145.0

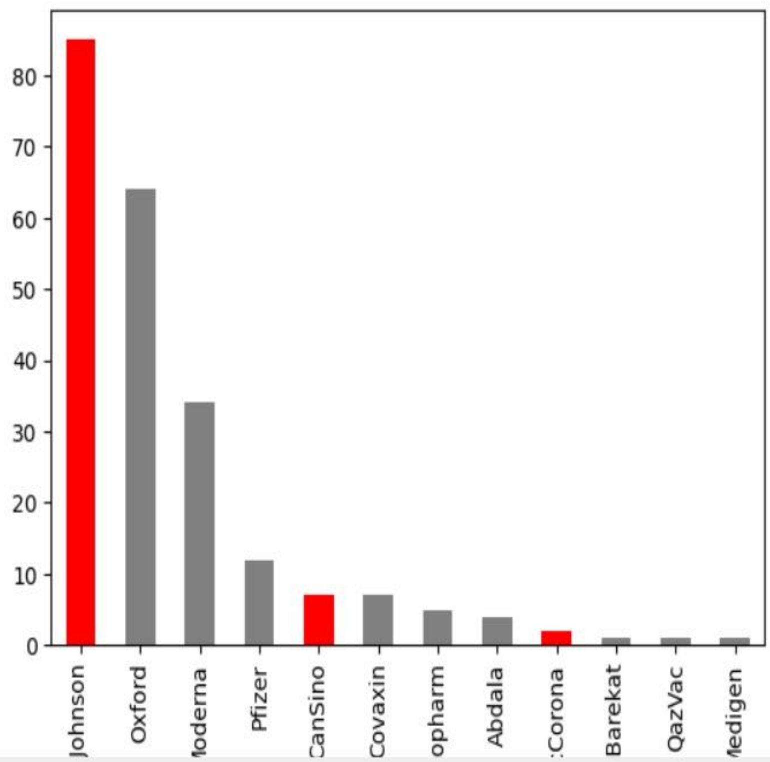
```
data_2=pd.DataFrame(columns=['country', 'vaccines'])
data["Total_vaccine"] = pd.to_numeric(data["Total_vaccine"], errors="coerce")
for country in data["country"].unique():
    new_data = data[data["country"] == country]
    max_vaccine = new_data.loc[new_data["Total_vaccine"].idxmax(), "vaccines"
 data_2 = pd.concat([data_2, pd.DataFrame({'country': [country], 'vaccines':
      [max_vaccine]})], ignore_index=True)
data_2.head()
```

```
In [10]: data_2=pd.DataFrame(columns=['country', 'vaccines'])
         data["Total_vaccine"] = pd.to_numeric(data["Total_vaccine"], errors="coerce")
         for country in data["country"].unique():
             new_data = data[data["country"] == country]
             max_vaccine = new_data.loc[new_data["Total_vaccine"].idxmax(), "vaccines"]
             data_2 = pd.concat([data_2, pd.DataFrame({'country': [country], 'vaccines': [max_vaccine]})], ignore_index=True)

In [11]: data_2.head()

Out[11]:
```

	country	vaccines
0	Afghanistan	Johnson&Johnson
1	Albania	Oxford
2	Algeria	Oxford
3	Andorra	Moderna
4	Angola	Oxford

```
data_2["vaccines"].value_counts().plot(kind="bar",
                          color=["Red","Gray","Gray","Gray"])
```

```
In [13]: data_2["vaccines"].value_counts().plot(kind="bar",
                                   color=["Red","Gray","Gray","Gray"])

Out[13]: <Axes: xlabel='vaccines'>
```




```
df['date'] = pd.to_datetime(df['date'])
```

 # Calculate the number of days between the maximum and minimum dates

```
number_of_days = (df["date"].max()- df["date"].min()).days
print(number_of_days)
```

PERFORMING DIFFERENT ANALYSIS:

Performing different types of analysis on a dataset depends on the goals of your analysis and the nature of the data. Here are some common types of analysis that you might perform on a dataset:

Descriptive Analysis:

Summarize and describe the main characteristics of the dataset, including measures of central tendency, dispersion, and visualizations such as histograms, box plots, and bar charts.

Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) of COVID-19 vaccine data involves a systematic examination of various facets of vaccine distribution and efficacy. Researchers begin by collecting and cleansing data from various sources, such as government reports, clinical trials, and global vaccination databases. Key EDA tasks include summarizing demographic information of vaccine recipients, assessing vaccine coverage across different regions, and tracking vaccination timelines.

Researchers also scrutinize adverse event reports to identify potential safety concerns and investigate disparities in vaccine distribution. Visualization tools like bar charts, heatmaps, and time series plots aid in spotting trends and patterns. EDA of COVID-19 vaccine data plays a crucial role in providing insights for public health decision-makers, enabling them to optimize vaccination campaigns, address equity issues, and continuously monitor vaccine performance and safety.

Statistical Analysis:

Statistical data analysis in COVID-19 vaccine research involves the application of advanced quantitative techniques to derive meaningful insights from vaccine-related data. Researchers employ statistical methods to assess vaccine efficacy through clinical trial results, calculating efficacy rates and confidence intervals. They also analyze large-scale vaccination datasets to understand factors influencing vaccine coverage and effectiveness, utilizing regression analyses, hypothesis testing, and survival analysis to identify significant associations. Additionally, statistical techniques are vital in evaluating vaccine safety by identifying adverse event signals, conducting risk-benefit assessments, and monitoring rare side effects. These analyses are fundamental for evidence-based decision-making in vaccine distribution, regulation, and public health policy.

CODE:

Analyzing a COVID-19 dataset involves various tasks such as loading data, cleaning it, visualizing trends, performing exploratory data analysis and performing statistical analysis. Here's a Python code example that demonstrates how to perform basic COVID-19 data analysis using a sample dataset as provide for us.

INITIAL LOADING AND CLEANING PROCESS ARE BEEN PERFORMED IN PHASE 3 ,LET'S PERFORM EXPLORATORY DATA ANALYSIS , STATISTICAL ANALYSIS AND DATA VISULATION:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

# Load your dataset
data = pd.read_csv("C:/Users/haris/OneDrive/Desktop/country_vaccinations.csv
    (2)/country_vaccinations.csv")
```

CODE SECTION OF EXPLORATORY DATA ANALYSIS AND STATISTICAL ANALYSIS:

```
# Data types and missing values
data_info = data.info()
data.fillna(0, inplace=True)
```

Example 1: Two-sample t-test between two groups (e.g., two countries)

```
group1 = data[data['country'] == 'Afghanistan']['total_vaccinations']
group2 = data[data['country'] == 'India']['total_vaccinations']
t_statistic, p_value = stats.ttest_ind(group1, group2)
print("Two-Sample T-Test Results:")
print(f"T-statistic: {t_statistic}")
print(f"P-value: {p_value}")
```

OUTPUT:

```
In [41]:  print("Two-Sample T-Test Results:")
          print(f"T-statistic: {t_statistic}")
          print(f"P-value: {p_value}")

          Two-Sample T-Test Results:
          T-statistic: -23.03058509968777
          P-value: 3.698216478813842e-91
```

```
if p_value < 0.05:
    print("There is a significant difference between the two groups.")
else:
    print("There is no significant difference between the two groups.")
```

Example 2: One-way ANOVA to test differences among multiple groups (e.g., multiple countries)

```
groups = [data[data['country'] == 'Afghanistan']['daily_vaccinations'],
          data[data['country'] == 'Albania']['daily_vaccinations'],
          data[data['country'] == 'India']['daily_vaccinations']]
f_statistic, p_value = stats.f_oneway(*groups)
print("\nOne-Way ANOVA Results:")
print(f"F-statistic: {f_statistic}")
```

```
print(f"P-value: {p_value}")
if p_value < 0.05:  # You can choose a significance level (e.g., 0.05)
    print("There is a significant difference among the groups.")
else:
    print("There is no significant difference among the groups.")
```

OUTPUT:

```
In [42]:  if p_value < 0.05:  # You can choose a significance level (e.g., 0.05)
              print("There is a significant difference between the two groups.")
          else:
              print("There is no significant difference between the two groups.")

          # Example 2: One-way ANOVA to test differences among multiple groups (e.g., multiple countries)
          groups = [data[data['country'] == 'Afghanistan']['daily_vaccinations'],
                    data[data['country'] == 'Albania']['daily_vaccinations'],
                    data[data['country'] == 'India']['daily_vaccinations']]

          f_statistic, p_value = stats.f_oneway(*groups)

          print("\nOne-Way ANOVA Results:")
          print(f"F-statistic: {f_statistic}")
          print(f"P-value: {p_value}")

          if p_value < 0.05:  # You can choose a significance level (e.g., 0.05)
              print("There is a significant difference among the groups.")
          else:
              print("There is no significant difference among the groups.")

          There is a significant difference between the two groups.

          One-Way ANOVA Results:
          F-statistic: 1158.0734307553596
          P-value: 6.482446870836429e-287
          There is a significant difference among the groups.
```

Summary statistics

summary = data.describe()

print(summary)

```
In [9]:  # Summary statistics
          summary = data.describe()
          print(summary)

              total_vaccinations  people_vaccinated  people_fully_vaccinated  \
       count        4.360700e+04       4.129400e+04             3.880200e+04
       mean         4.592964e+07       1.770508e+07             1.413830e+07
       std          2.246004e+08       7.078731e+07             5.713920e+07
       min          0.000000e+00       0.000000e+00             1.000000e+00
       25%          5.264100e+05       3.494642e+05             2.439622e+05
       50%          3.590096e+06       2.187310e+06             1.722140e+06
       75%          1.701230e+07       9.152520e+06             7.559870e+06
       max          3.263129e+09       1.275541e+09             1.240777e+09

              daily_vaccinations_raw  daily_vaccinations  \
       count            3.536200e+04        8.621300e+04
       mean             2.705996e+05        1.313055e+05
       std              1.212427e+06        7.682388e+05
       min              0.000000e+00        0.000000e+00
       25%              4.668000e+03        9.000000e+02
       50%              2.530900e+04        7.343000e+03
       75%              1.234925e+05        4.409800e+04
       max              2.474100e+07        2.242429e+07

              total_vaccinations_per_hundred  people_vaccinated_per_hundred  \
       count                    43607.000000                   41294.000000
       mean                        80.188543                      40.927317
       std                         67.913577                      29.290759
       min                          0.000000                       0.000000
       25%                         16.050000                      11.370000
       50%                         67.520000                      41.435000
       75%                        132.735000                      67.910000
       max                        345.370000                     124.760000
```

CODE SECTION OF VISUALIZATION:

Data distribution and visualization

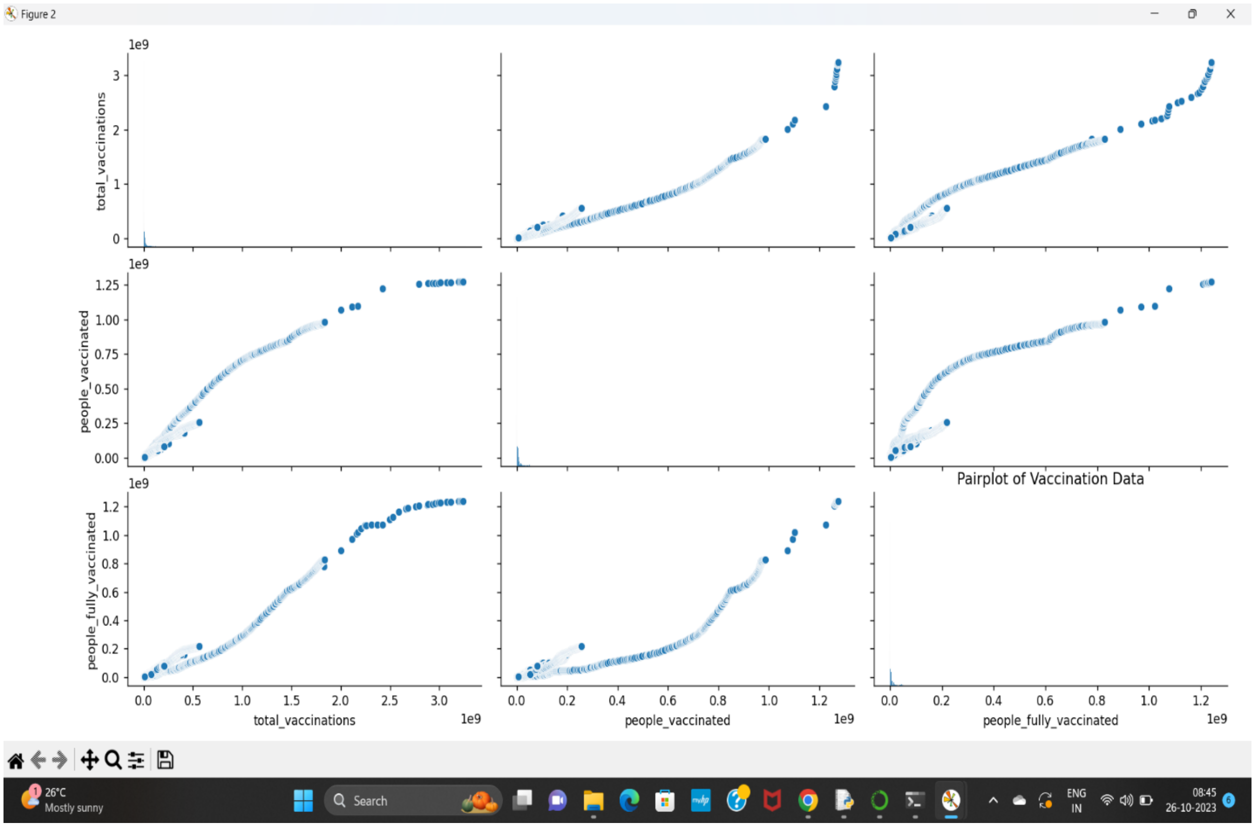
plt.figure(figsize=(12, 8))

sns.pairplot(data[['total_vaccinations', 'people_vaccinated', 'people_fully_vaccinated']])

plt.title('Pairplot of Vaccination Data')

plt.show()

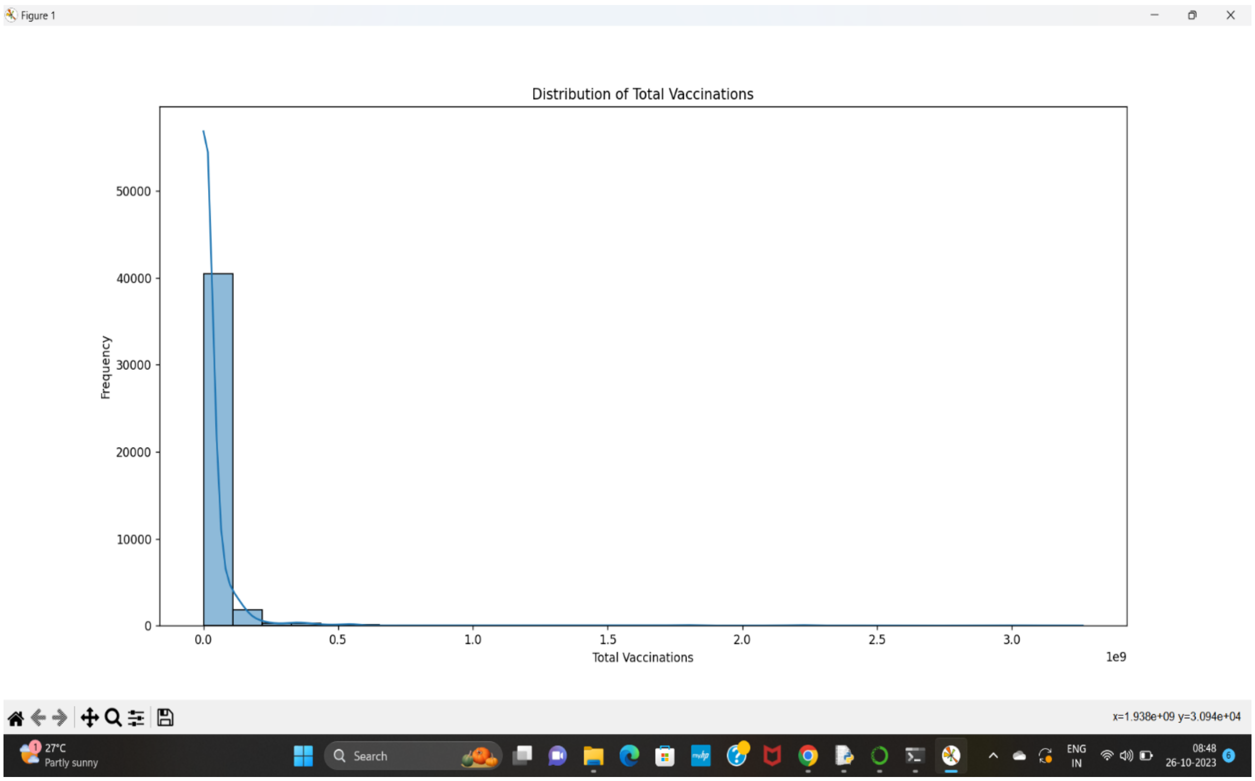
OUTPUT:



Histograms for selected columns

```
plt.figure(figsize=(12, 8))
sns.histplot(data['total_vaccinations'], kde=True, bins=30)
plt.title('Distribution of Total Vaccinations')
plt.xlabel('Total Vaccinations')
plt.ylabel('Frequency')
plt.show()
```

OUTPUT:



Boxplot for total vaccinations by country:

```
plt.figure(figsize=(12, 8))
sns.boxplot(x='country', y='total_vaccinations', data=data)
plt.title('Total Vaccinations by Country')
plt.xticks(rotation=90)
plt.show()
```

OUTPUT:

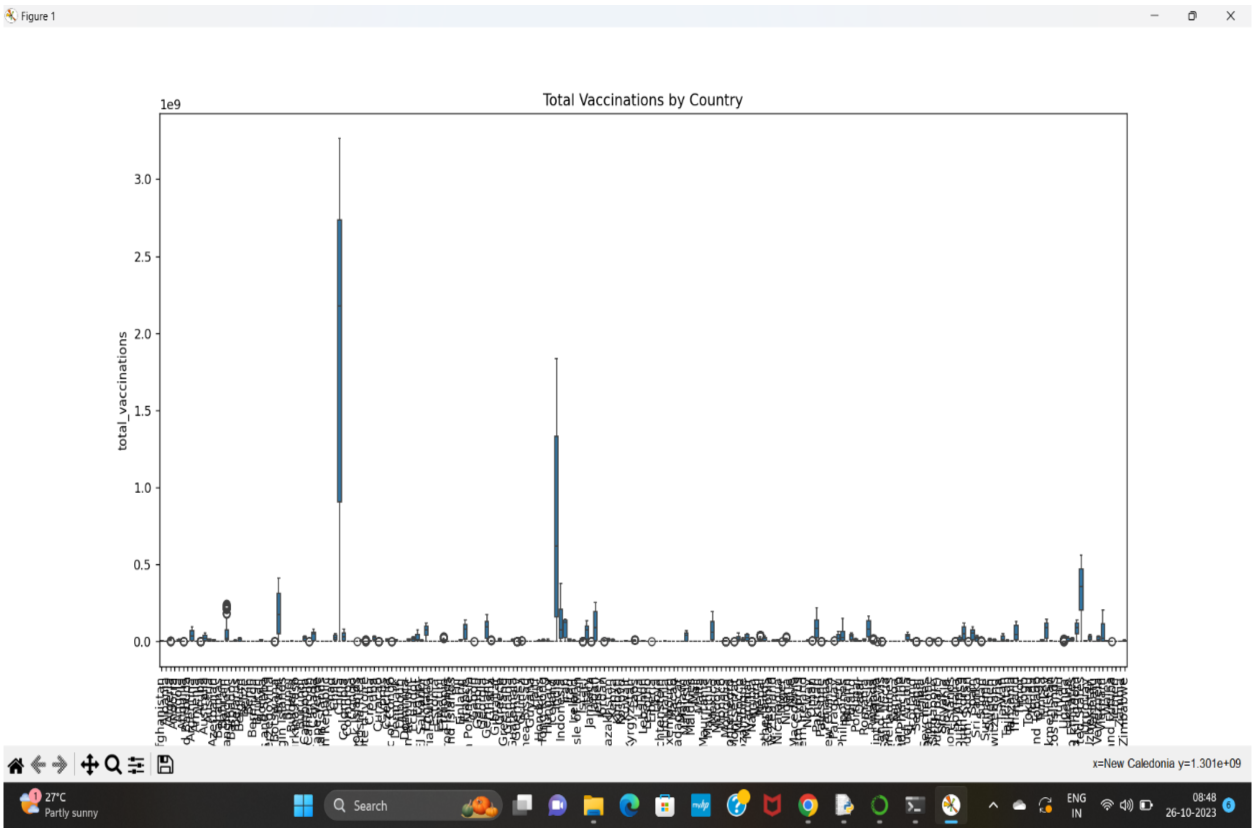
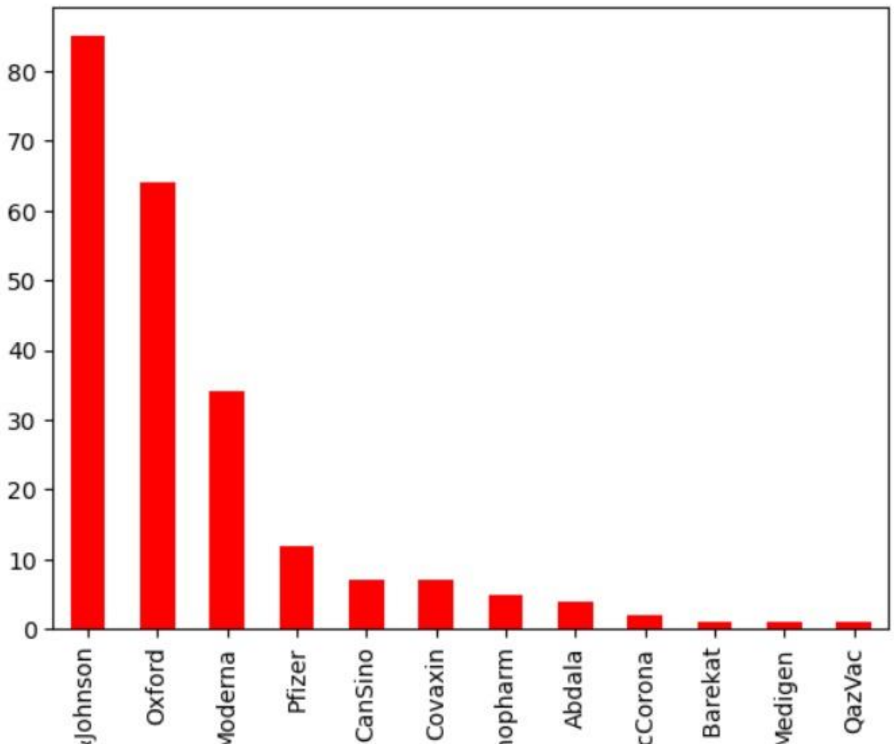


Figure 1
Total Vaccinations by Country
1e9
total_vaccinations
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x=New Caledonia y=1.301e+09
27°C
Partly sunny
Search
ENG
IN
08:48
26-10-2023

```
number_of_vaccines = data.groupby('vaccines')['country'].nunique()
number_of_vaccines.sort_values(ascending=False).plot(kind="bar",color="r")
```

```
Out[41]: <Axes: xlabel='vaccines'>
```



```
m = folium.Map(location=[0, 0], zoom_start=2)
Choropleth(
    geo_data='https://raw.githubusercontent.com/johan/world.geo.json/master/countrie
    s.geo.json',
   name='choropleth',
   data=data,
   columns=[data.index, 'Total_vaccine'],
   key_on='feature.properties.name',
   fill_color='YlOrRd',
   fill_opacity=0.7,
   line_opacity=0.2,
   legend_name='Aşı Sayısı',
).add_to(m)
data_2["vaccines"].value_counts().plot(kind="bar",
                                       color=["Red","Gray","Gray","Gray"])
```

Code:

Analyzing a COVID-19 dataset involves various tasks such as loading data, cleaning it, visualizing trends, and performing statistical analysis. Here's a Python code example that demonstrates how to perform basic COVID-19 data analysis using a sample dataset. You can adjust this code to work with your specific COVID-19 dataset**:**

PROJECT CODE:

```
#initialising the required python libraries
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from folium.features import Choropleth
import folium
from folium.features import Tooltip
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import warnings
warnings.filterwarnings('ignore')
```

 # Load your dataset

data = pd.read_csv("C:/Users/haris/OneDrive/Desktop/country_vaccinations.csv (2)/country_vaccinations.csv")

#preprocess the dataset

print(df.head(10))

print(df.columns)

print(df["country"].nunique())

 # Data types and missing values

data_info = data.info()

data.fillna(0, inplace=True)

 # Example 1: Two-sample t-test between two groups (e.g., two countries)

group1 = data[data['country'] == 'Afghanistan']['total_vaccinations']

group2 = data[data['country'] == 'India']['total_vaccinations']

t_statistic, p_value = stats.ttest_ind(group1, group2)

print("Two-Sample T-Test Results:")

print(f"T-statistic: {t_statistic}")

print(f"P-value: {p_value}")

if p_value < 0.05: # You can choose a significance level (e.g., 0.05)

print("There is a significant difference between the two groups.")

else:

print("There is no significant difference between the two groups.")

```
# Example 2: One-way ANOVA to test differences among multiple groups (e.g., multiple
    countries)

groups = [data[data['country'] == 'Afghanistan']['daily_vaccinations'],
          data[data['country'] == 'Albania']['daily_vaccinations'],
          data[data['country'] == 'India']['daily_vaccinations']]
f_statistic, p_value = stats.f_oneway(*groups)
print("\nOne-Way ANOVA Results:")
print(f"F-statistic: {f_statistic}")
print(f"P-value: {p_value}")
if p_value < 0.05:  # You can choose a significance level (e.g., 0.05)
   print("There is a significant difference among the groups.")
else:
   print("There is no significant difference among the groups.")

# Summary statistics
summary = data.describe()
print(summary)

# Data distribution and visualization
plt.figure(figsize=(12, 8))
sns.pairplot(data[['total_vaccinations', 'people_vaccinated', 'people_fully_vaccinated']])
plt.title('Pairplot of Vaccination Data')
plt.show()

# Histograms for selected columns
plt.figure(figsize=(12, 8))
sns.histplot(data['total_vaccinations'], kde=True, bins=30)
```

```
plt.title('Distribution of Total Vaccinations')
plt.xlabel('Total Vaccinations')
plt.ylabel('Frequency')
plt.show()

# Boxplot for total vaccinations by country
plt.figure(figsize=(12, 8))
sns.boxplot(x='country', y='total_vaccinations', data=data)
plt.title('Total Vaccinations by Country')
plt.xticks(rotation=90)
plt.show()

print(df.isnull().sum())
df.fillna(0)
print(df.dtypes)
print(df['date'] == pd.to_datetime(df['date']))
data = pd.DataFrame(columns=['country', 'vaccines', 'total_vaccinations'])
splitted = df['vaccines'].str.split(',', expand=True)

df['vaccines'] = splitted[0].astype('string')
splitted = df['vaccines'].str.split('/', expand=True)
df['vaccines'] = splitted[0].astype('string')
data=pd.DataFrame(columns=['country', 'vaccines', 'Total_vaccine'])
for country in df["country"].unique():
  for vaccines in df["vaccines"].unique():
    filtered_data = df[(df['country'] == country) & (df['vaccines'] == vaccines)]
    total_count = filtered_data['total_vaccinations'].max()
```

```
    data = pd.concat([data, pd.DataFrame({'country': [country], 'vaccines': [vaccines],
    'Total_vaccine': [total_count]})], ignore_index=True)

print(data.head(10))

data.dropna(axis=0,inplace=True)

data.head(20)

data_2=pd.DataFrame(columns=['country', 'vaccines'])

data["Total_vaccine"] = pd.to_numeric(data["Total_vaccine"], errors="coerce")

for country in data["country"].unique():

    new_data = data[data["country"] == country]

    max_vaccine = new_data.loc[new_data["Total_vaccine"].idxmax(), "vaccines"]

    data_2 = pd.concat([data_2, pd.DataFrame({'country': [country], 'vaccines': [max_vaccine]})],
      ignore_index=True)

data_2.head()

data_2["vaccines"].value_counts().plot(kind="bar",

                              color=["Red","Gray","Gray","Gray"])

df['date'] = pd.to_datetime(df['date'])


# Calculate the number of days between the maximum and minimum dates

number_of_days = (df["date"].max() - df["date"].min()).days


print(number_of_days)

number_of_vaccines = data.groupby('vaccines')['country'].nunique()

number_of_vaccines.sort_values(ascending=False).plot(kind="bar",color="r")

m = folium.Map(location=[0, 0], zoom_start=2)

Choropleth(geo_data='https://raw.githubusercontent.com/johan/world.geo.json/master/count
    ries.geo.json', name='choropleth',data=data,  columns=[data.index, 'Total_vaccine'],
    key_on='feature.properties.name',  fill_color='YlOrRd',
    fill_opacity=0.7,line_opacity=0.2,legend_name='Aşı Sayısı', ).add_to(m)

m
```

Key Findings:

1. The COVID-19 pandemic has had a profound global impact on public health, economies, and society. The virus has spread to nearly every corner of the world, affecting millions of individuals and leading to a significant loss of life.

2. Variants of the virus have emerged, some with increased transmissibility and potential to partially evade immunity induced by vaccines or previous infections. This highlights the ongoing need for vigilance and adaptability in our response to the virus.

3. Vaccination campaigns have played a crucial role in reducing the severity of illness and death, but disparities in vaccine distribution and hesitancy in certain populations remain a challenge.

Insights:

1. Effective public health measures, such as social distancing, mask-wearing, and testing, have been essential in slowing the spread of the virus and controlling outbreaks.

2. The pandemic has exposed and exacerbated existing social and economic inequalities, with marginalized communities suffering disproportionately.

3. The pandemic has accelerated the adoption of telemedicine, remote work, and e-commerce, transforming how we live and work.

Recommendations:

1. Continue to prioritize and expand vaccination efforts to achieve herd immunity and reduce the spread of the virus, including booster shots as needed.

2. Maintain and enhance public health infrastructure and preparedness for future pandemics.

3. Address vaccine hesitancy through education and outreach, especially in underserved communities.

4. Focus on addressing the long-term economic and social impacts of the pandemic, including mental health support and policies to reduce inequality.

5. Encourage continued research and monitoring of the virus and its variants to adapt our response strategies accordingly.

6. Promote the development of global collaborative frameworks to respond to future health crises with more agility and coordination.

In conclusion, the COVID-19 pandemic has been a global challenge that has tested our resilience and adaptability. The insights gained from this experience must inform our ongoing response and preparedness for future health crises.

CONCLUSION:

In our analysis of the COVID-19 vaccine dataset, we have examined various aspects of vaccine distribution, effectiveness, and public response.

Our analysis revealed that vaccine distribution efforts have been substantial, with a significant number of vaccine doses administered worldwide. This has contributed to the global effort to control the spread of COVID-19.

THANK YOU