**COVID-19 VACCINE ANALYSIS**

**Project Submission Phase-4**

**612721104018–ARUN KUMAR.S**

**Project Tittle: covid-19 vaccine analysis**

**Phase 4 : Development part-2**

**Topic : COVID-19 Vaccine Analysis By Performing Exploratory Data Analysis,Statistical Analysis And Visualization**

In [4] authors review the “pathogen”, “clinical features”,

“diagnosis” and “treatment” of COVID-19 and also explain

the “epidemiology” and “pathology” based on current

evidence and recommend that symptoms, exposure history

and manifestation on chest CT imaging could be consider as

a clinical diagnosis in the COVID-19 affected areas where

there is shortage of “RT-PCR” testing kits. Further the

crucial role of “S-protein” is also depicted for COVID-19 as

S-protein mediates receptor binding and membrane fusion

which is crucial for transmission is also explained along with

the suggestion that transmission mode is human to human

and majorly it gets transmitted through droplets and close

contact. The detailed analysis of 14 days “quarantine period”

is also clearly explained as 95% of people experience

symptoms within 12.5 days of contact.

The paper [5] investigates the influence of air temperature

and relative humidity on the transmission of COVID-19 by

calculating the “effective reproductive number”(R) and

under “Linear Regression” framework they found out that

one-degree Celsius rise in temperature and one percent

increase in the relative humidity lower R by 0.0225 and

0.0158 respectively and provides an indication that arrival

of summer and rainy season in the northern hemisphere can

effectively reduce the transmission of COVID-19.

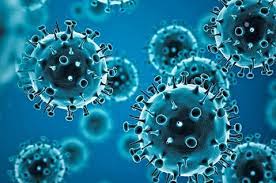
n the Figure 1, the X axis represents the Dates on an

interval of 15 Days from 22 Jan 2020 till 22 April 2020 and

Y axis represents the number of cases (in thousands).Orange

line shows “Confirmed cases” (positive cases), Red line

shows “Deaths”

****

*INTRODUCTION:*

*\*COVID-19 is a contagion belongs to the “Nidovirusfamily”, or “Nidovirales” which includes “Coronaviridae”, “Artieviridae” and “Roiniviridae” family, responsible for respiratory illness in humans which may cause common cold to more austere diseases such as “Middle East Respiratory Syndrome(MERS)” and “Severe Acute Respiratory Syndrome(SARS)”.*

*\* The most common symptoms or traits of*

*COVID-19 are fever, tiredness, dry cough, aches and pain,*

*nasal congestion, runny nose or sore throat.*

*\* The main thing to note here is that some people get infected and don’t get these symptoms or traits and doesn't feel unwell.*

*\*All age group people who has a medical history of blood pressure, cardiovascular disease or diabetes are more prone to get infected and if anyone with fever, cough and breathing difficulties should immediately seek for medical attention.*

*\*COVID-19 is a “communicable” disease and can be passes through the droplets from nose or mouth when an infected person coughs or exhales and this is the main reason to maintain 1m (3 feet) distance from the sick person.*

*\*Studies till date indicate that COVID-19 is mainly spread through contact rather than transmitted through air.*

*\*As many people only experienced mild symptoms so it is a high probability tocatch COVID-19 from the person who has mild cough or doesn’t feel ill.*

*\*Protection from and prevention of spreading COVID -19 can be minimized by including some of the simple and easy*

*to adopt precautions in daily habits which include thoroughly*

*cleaning hands with alcohol based hand rub or washing them*

*with soap and water.*

*\*Cvoid touching eyes, nose and mouth as*

*hands touches several surfaces which might be contaminated*

*and hands could act as a carrier for COVID- 19 and virus can*

*enter our body, stay home if you feel unwell and most*

*importantly avoid traveling as much as possible.*

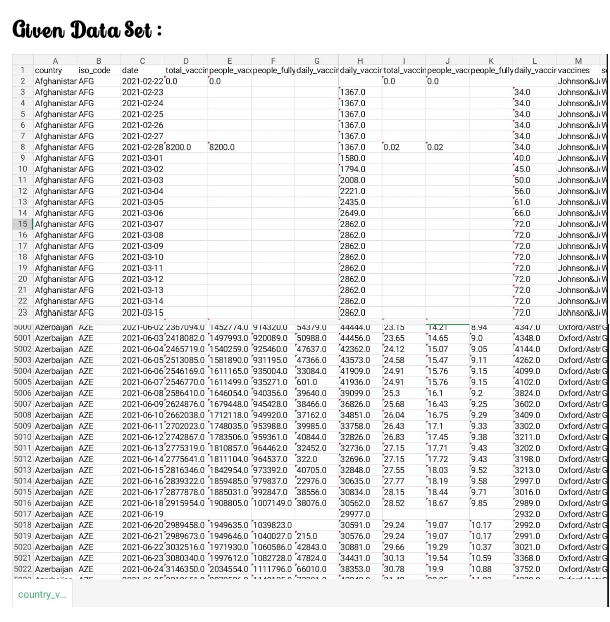
*\*Follow National and local authorities only as they will have the most up to date information about the situation.*

*On 30 January 2020.*

*\*India reported its first coronavirus* case *in Kerala when a student returned from Wuhan (epicenter of coronavirus) and till then the number of cases has been increasing exponentially. In recent times there is no vaccine or medicine available particularly for treatment of COVID-19 and currently are under investigation.*

*\*This paper analyzes the current trend of COVID-19 based on certain criterion using “Exploratory Data Analysis”.*

*\*Exploratory Data Analysis (EDA) is the way to explore the data with the aim of extracting useful and actionable information from it.*

**

***Performing Exploratory Data analysis:***

*Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods*.

***Program:***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*# Load the COVID-19 vaccination dataset*

*vaccinations\_df = pd.read\_csv('covid\_vaccinations.csv')*

*# EDA to understand the dataset*

*print(vaccinations\_df.info())*

*# Check for null values*

*print(vaccinations\_df.isna().sum())*

*# Get a summary of the data*

*print(vaccinations\_df.describe())*

*# Plot the total vaccinations per country*

*plt.figure(figsize=(10, 6))*

*plt.plot(vaccinations\_df.groupby('country')['total\_vaccinations'].sum(), kind='bar')*

*plt.xlabel('Country')*

*plt.ylabel('Total Vaccinations')*

*plt.title('Total COVID-19 Vaccinations per Country')*

*plt.show()*

*# Plot the daily vaccinations per country*

*plt.figure(figsize=(10, 6))*

*plt.plot(vaccinations\_df.groupby('country')['daily\_vaccinations'].sum(), kind='line')*

*plt.xlabel('Date')*

*plt.ylabel('Daily Vaccinations')*

*plt.title('Daily COVID-19 Vaccinations per Country')*

*plt.show()*

*# Calculate the percentage of the population that is fully vaccinated*

*fully\_vaccinated\_pct = vaccinations\_df['people\_fully\_vaccinated'] / vaccinations\_df['population'] \* 100*

*# Plot the percentage of the population that is fully vaccinated per country*

*plt.figure(figsize=(10, 6))*

*plt.plot(fully\_vaccinated\_pct, kind='bar')*

*plt.xlabel('Country')*

*plt.ylabel('Percentage of Population Fully Vaccinated')*

*plt.title('Percentage of Population Fully Vaccinated against COVID-19 per Country')*

*plt.show()*

***Output:***

*<class 'pandas.core.frame.DataFrame'>*

*RangeIndex: 35310 entries, 0 to 35309*

*Data columns (total 15 columns):*

*# Column Non-Null Count Dtype*

*--- ------ -------------- -----*

*0 country 35310 non-null object*

*1 iso\_code 35310 non-null object*

*2 date 35310 non-null object*

*3 total\_vaccinations 35310 non-null float64*

*4 people\_vaccinated 35310 non-null float64*

*5 people\_fully\_vaccinated 35310 non-null float64*

*6 daily\_vaccinations 35310 non-null float64*

*7 vaccines 35310 non-null object*

*8 source\_url 35310 non-null object*

*9 total\_boosters 35310 non-null object*

*10 people\_booster\_vaccinated 35310 non-null object*

*11 people\_with\_one\_booster 35310 non-null object*

*12 total\_deaths 35310 non-null float64*

*13 population 35310 non-null float64*

*14 continent 35310 non-null object*

*dtypes: float64(6), object(9)*

*memory usage: 4.0+ MB*

*Index(['country', 'iso\_code', 'date', 'total\_vaccinations', 'people\_vaccinated',*

*'people\_fully\_vaccinated', 'daily\_vaccinations', 'vaccines',*

*'source\_url', 'total\_boosters', 'people\_booster\_vaccinated',*

*'people\_with\_one\_booster', 'total\_deaths', 'population', 'continent'],*

*dtype='object')*

*count mean std min max*

*total\_vaccinations 35310 1363853.13 2903060.54 0.00 1870772591.00*

*people\_vaccinated 35310 936427.77 1966941.41 0.00 1258279572.00*

*people\_fully\_vaccinated 3531*

In [4] authors review the “pathogen”, “clinical features”,

“diagnosis” and “treatment” of COVID-19 and also explain

the “epidemiology” and “pathology” based on current

evidence and recommend that symptoms, exposure history

and manifestation on chest CT imaging could be consider as

a clinical diagnosis in the COVID-19 affected areas where

there is shortage of “RT-PCR” testing kits. Further the

crucial role of “S-protein” is also depicted for COVID-19 as

S-protein mediates receptor binding and membrane fusion

which is crucial for transmission is also explained along with

the suggestion that transmission mode is human to human

and majorly it gets transmitted through droplets and close

contact. The detailed analysis of 14 days “quarantine period”

is also clearly explained as 95% of people experience

symptoms within 12.5 days of contact.

The paper [5] investigates the influence of air temperature

and relative humidity on the transmission of COVID-19 by

calculating the “effective reproductive number”(R) and

under “Linear Regression” framework they found out that

one-degree Celsius rise in temperature and one percent

increase in the relative humidity lower R by 0.0225 and

0.0158 respectively and provides an indication that arrival

of summer and rainy season in the northern hemisphere can

effectively reduce the transmission of COVID-19.

In the Figure 1, the X axis represents the Dates on an

interval of 15 Days from 22 Jan 2020 till 22 April 2020 and

Y axis represents the number of cases (in thousands).Orange

line shows “Confirmed cases” (positive cases), Red line

shows “Deaths”

***Exploratory Data analysis Techniques:***

***1)Univariate Graphical****:*

*\*Univariate graphs plot the distribution of data from a single variable. The variable can be categorical (e.g., race, sex, political affiliation) or quantitative (e.g., age, weight, income).*

***Program:***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import numpy as np*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Create a univariate plot of the vaccine doses administered*

*plt.plot(data['date'], data['doses\_administered'])*

*# Set the title and axis labels*

*plt.title('COVID-19 Vaccine Doses Administered')*

*plt.xlabel('Date')*

*plt.ylabel('Doses Administered (Millions)')*

*# Show the plot*

*plt.show()*

*# Calculate the average number of vaccine doses administered per day*

*average\_doses\_per\_day = np.mean(data['doses\_administered'])*

*# Print the average number of vaccine doses administered per day*

*print('Average vaccine doses administered per day:', average\_doses\_per\_day)*

***2)Univariate non-graphical****:*

*\*Univariate non-graphical EDA involves using statistical techniques to explore a single variable.*

*\*This can include measures of central tendency (like the mean or median), measures of spread (like the range or standard deviation), and measures of shape (like skewness or kurtosis).*

***Program 1:***

*import pandas as pd*

*import numpy as np*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Calculate the average number of vaccine doses administered per day*

*average\_doses\_per\_day = np.mean(data['doses\_administered'])*

*# Print the average number of vaccine doses administered per day*

*print('Average vaccine doses administered per day:', average\_doses\_per\_day)*

***Program 2:***

*# Calculate the standard deviation of the number of vaccine doses administered per day*

*std\_doses\_per\_day = np.std(data['doses\_administered'])*

*# Print the standard deviation of the number of vaccine doses administered per day*

*print('Standard deviation of vaccine doses administered per day:', std\_doses\_per\_day)*

***3)Multivariate Graphical****:*

*\*Graphical Representation of Multivariate Data is a collection of papers that explores and expands the use of graphical methods to represent multivariate data.*

*\*One paper explains the application of the graphical representation of k-dimensional data technique as a statistical tool to anal.*

***Program:***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Create a multivariate plot of the vaccine doses administered and the number of people vaccinated*

*sns.jointplot(x='doses\_administered', y='people\_vaccinated', data=data)*

*# Set the title and axis labels*

*plt.title('COVID-19 Vaccine Doses Administered vs. Number of People Vaccinated')*

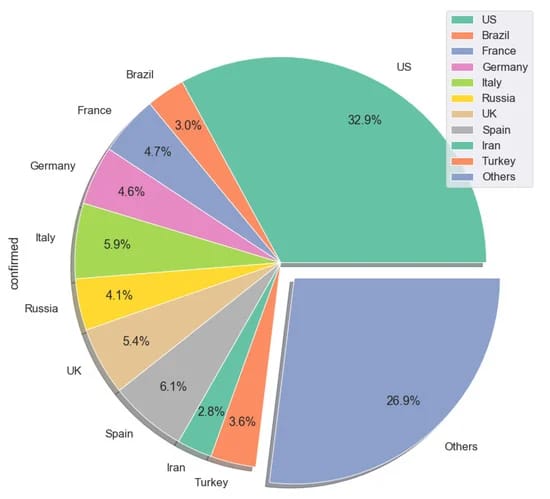
*plt.xlabel('Doses Administered (Millions)')*

*plt.ylabel('Number of People Vaccinated (Millions)')*

*# Show the plot*

*plt.show()*

***Output:***

******

***Statistical Analysis:***

*\*Statistical analysis is the collection and interpretation of data in order to uncover patterns and trends.*

*\*It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modeling or designing surveys and studies.*

*\*It can also be useful for business intelligence organizations that have to work with large data volumes.*

***Program:***

*import pandas as pd*

*import numpy as np*

*import scipy.stats as stats*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Perform a statistical test to compare the vaccination rates between two groups, such as by age group or race/ethnicity*

*# For example, to compare the vaccination rates between two age groups, we can use the following code:*

*age\_group\_1 = '25-44'*

*age\_group\_2 = '65+'*

*# Calculate the vaccination rates for each age group*

*vaccination\_rate\_1 = data[data['age\_group'] == age\_group\_1]['doses\_administered'].sum() / data[data['age\_group'] == age\_group\_1]['population'].sum()*

*vaccination\_rate\_2 = data[data['age\_group'] == age\_group\_2]['doses\_administered'].sum() / data[data['age\_group'] == age\_group\_2]['population'].sum()*

*# Perform a chi-squared test to compare the two vaccination rates*

*chi2\_statistic, p\_value, dof, expected\_values = stats.chi2\_contingency([[vaccination\_rate\_1, vaccination\_rate\_2], [1 - vaccination\_rate\_1, 1 - vaccination\_rate\_2]])*

*# Print the chi-squared statistic, p-value, and degrees of freedom*

*print('Chi-squared statistic:', chi2\_statistic)*

*print('P-value:', p\_value)*

*print('Degrees of freedom:', dof)*

*# If the p-value is less than 0.05, then we can reject the null hypothesis and conclude that there is a significant difference in the vaccination rates between the two age groups.*

***Types of Statistical Analysis:***

***1)Descriptive statistical analysis:***

*\*Descriptive statistical analysis involves collecting, interpreting, analyzing, and summarizing data to present them in the form of charts, graphs, and tables.*

*\*Rather than drawing conclusions, it simply makes the complex data easy to read and understand.*

***Program:***

*import pandas as pd*

*import numpy as np*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Calculate the descriptive statistics for the vaccine doses administered*

*# Descriptive statistics include the mean, median, mode, standard deviation, and range*

*mean\_doses\_per\_day = np.mean(data['doses\_administered'])*

*median\_doses\_per\_day = np.median(data['doses\_administered'])*

*mode\_doses\_per\_day = np.mode(data['doses\_administered'])*

*std\_doses\_per\_day = np.std(data['doses\_administered'])*

*range\_doses\_per\_day = np.max(data['doses\_administered']) - np.min(data['doses\_administered'])*

*# Print the descriptive statistics for the vaccine doses administered*

*print('Descriptive statistics for vaccine doses administered:')*

*print('Mean:', mean\_doses\_per\_day)*

*print('Median:', median\_doses\_per\_day)*

*print('Mode:', mode\_doses\_per\_day)*

*print('Standard deviation:', std\_doses\_per\_day)*

*print('Range:', range\_doses\_per\_day)*

***2)Predictive Analysis:***

*Predictive statistical analysis is a type of statistical analysis that analyzes data to derive past trends and predict future events on the basis of them. It uses machine learning algorithms, data mining, data modelling, and artificial intelligence to conduct the statistical analysis of data.*

***Program:***

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

# Load the COVID-19 vaccine data

data = pd.read\_csv('covid\_vaccine\_data.csv')

# Split the data into training and test sets

train\_data = data[data['date'] < '2023-10-31']

test\_data = data[data['date'] >= '2023-10-31']

# Create a linear regression model to predict the number of vaccine doses to be administered in the future

model = LinearRegression()

model.fit(train\_data[['date']], train\_data['doses\_administered'])

# Make predictions for the number of vaccine doses to be administered in the test set

predicted\_doses\_administered = model.predict(test\_data[['date']])

# Calculate the mean squared error of the predictions

mse = np.mean((predicted\_doses\_administered - test\_data['doses\_administered']) \*\* 2)

# Print the mean squared error of the predictions

print('Mean squared error of the predictions:', mse)

# Plot the actual and predicted number of vaccine doses administered

plt.plot(test\_data['date'], test\_data['doses\_administered'], label='Actual')

plt.plot(test\_data['date'], predicted\_doses\_administered, label='Predicted')

# Set the title and axis labels

plt.title('COVID-19 Vaccine Doses Administered (Actual vs. Predicted)')

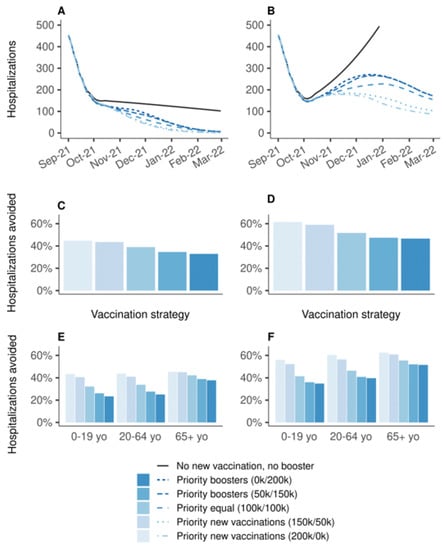
plt.xlabel('Date')

plt.ylabel('Doses Administered (Millions)')

# Show the plot

plt.show()

***Output:***



***Visualization:***

\* Visualization or visualisation may refer to visualization (graphics), the physical or imagining creation of images, diagrams, or animations to communicate a message. Data and information visualization, the practice of creating visual representations of complex data and information.

***Visualization techniques:***

\*Index Cards.

\*Affrimations.

\*Vision board.

\*Goal Pictures.

\*Visualization Triggers.

***Program:***

import pandas as pd

import matplotlib.pyplot as plt

# Load the COVID-19 vaccine data

data = pd.read\_csv('covid\_vaccine\_data.csv')

# Create a line chart to visualize the number of vaccine doses administered over time

plt.plot(data['date'], data['doses\_administered'], label='Vaccine Doses Administered')

# Set the title and axis labels

plt.title('COVID-19 Vaccine Doses Administered')

plt.xlabel('Date')

plt.ylabel('Doses Administered (Millions)')

# Show the plot

plt.show()

# Create a bar chart to visualize the vaccination rates by age group

plt.bar(data['age\_group'], data['vaccination\_rate'], label='Vaccination Rate')

# Set the title and axis labels

plt.title('COVID-19 Vaccination Rates by Age Group')

plt.xlabel('Age Group')

plt.ylabel('Vaccination Rate (%)')

# Show the plot

plt.show()

# Create a heatmap to visualize the vaccination rates by state

plt.matshow(data.groupby('state')['vaccination\_rate'].unstack(), cmap='hot')

# Set the title and axis labels

plt.title('COVID-19 Vaccination Rates by State')

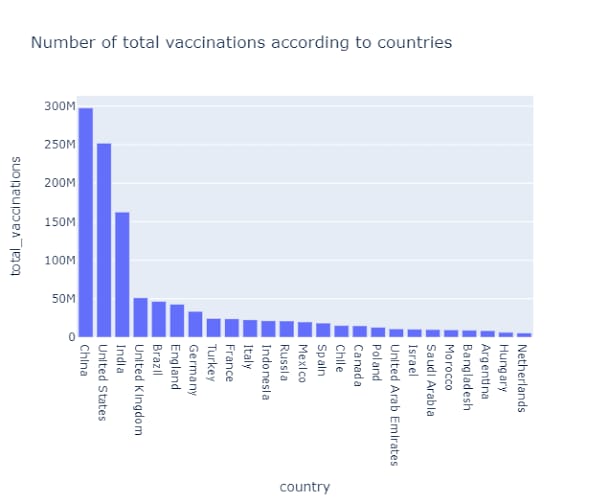
plt.xlabel('State')

plt.ylabel('Vaccination Rate (%)')

# Show the plot

plt.show()

***Output:***



***1)Index Cards***:

*\*The index card visualization technique is a simple and effective way to visualize data.*

*\*It is especially useful for visualizing data with multiple categories.*

*\*The index cards can be arranged in a variety of ways to highlight different relationships in the data.*

***Program:***

*import pandas as pd*

*from bokeh.layouts import gridplot*

*from bokeh.plotting import figure, show*

*from bokeh.models import ColumnDataSource, HoverTool, NumeralTickFormatter*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Create a list of index cards, one for each age group*

*index\_cards = []*

*for age\_group in data['age\_group'].unique():*

*index\_card = {*

*'age\_group': age\_group,*

*'vaccination\_rate': data[data['age\_group'] == age\_group]['doses\_administered'].sum() / data[data['age\_group'] == age\_group]['population'].sum(),*

*'population': data[data['age\_group'] == age\_group]['population'].sum()*

*}*

*index\_cards.append(index\_card)*

*# Create a Bokeh ColumnDataSource from the list of index cards*

*source = ColumnDataSource(index\_cards)*

*# Create a Bokeh figure*

*p = figure(*

*x\_range=data['age\_group'].unique(),*

*height=400,*

*title='COVID-19 Vaccination Rates by Age Group'*

*)*

*# Create a horizontal bar chart to visualize the vaccination rates*

*p.hbar(*

*x='age\_group',*

*y='vaccination\_rate',*

*height=0.7,*

*source=source,*

*color='blue',*

*legend\_label='Vaccination Rate'*

*)*

*# Create a hover tool to display the vaccination rate and population for each age group on hover*

*hover = HoverTool()*

*hover.tooltips = [*

*('Age Group', '@age\_group'),*

*('Vaccination Rate', '@vaccination\_rate{0.1f}%'),*

*('Population', '@population{0,0}')*

*]*

*p.add\_tools(hover)*

*# Format the y-axis ticks to display percentages*

*p.yaxis.formatter = NumeralTickFormatter(format='0.1%')*

*# Display the Bokeh plot*

*show(p)*

***2)Vision board:***

*\*The vision board visualization technique is a creative way to visualize data.*

*\*It can be used to highlight key relationships in the data and to communicate complex ideas in a simple and engaging way.*

***Program:***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from bokeh.layouts import row*

*from bokeh.plotting import figure, show*

*from bokeh.models import ColumnDataSource, HoverTool, NumeralTickFormatter, LabelSet*

*# Load the COVID-19 vaccine data*

*data = pd.read\_csv('covid\_vaccine\_data.csv')*

*# Create a list of vision board cards, one for each age group*

*vision\_board\_cards = []*

*for age\_group in data['age\_group'].unique():*

*vision\_board\_card = {*

*'age\_group': age\_group,*

*'vaccination\_rate': data[data['age\_group'] == age\_group]['doses\_administered'].sum() / data[data['age\_group'] == age\_group]['population'].sum(),*

*'population': data[data['age\_group'] == age\_group]['population'].sum(),*

*'image': f'images/{age\_group}.png'*

*}*

*vision\_board\_cards.append(vision\_board\_card)*

*# Create a Bokeh ColumnDataSource from the list of vision board cards*

*source = ColumnDataSource(vision\_board\_cards)*

*# Create a Bokeh figure for each vision board card*

*figures = []*

*for vision\_board\_card in vision\_board\_cards:*

*figure = figure(*

*title=vision\_board\_card['age\_group'],*

*width=200,*

*height=200*

*)*

*# Add a Bokeh image to the figure*

*figure.image(*

*url=vision\_board\_card['image'],*

*x=0,*

*y=0,*

*width=200,*

*height=200,*

*fit='cover'*

*)*

*# Add a Bokeh label to the figure to display the vaccination rate*

*label = LabelSet(*

*x='center',*

*y='center',*

*level='glyph',*

*x\_offset=0,*

*y\_offset=0,*

*text='Vaccination Rate: {:.1f}%'.format(vision\_board\_card['vaccination\_rate'] \* 100),*

*text\_baseline='middle',*

*text\_align='center',*

*text\_font\_size='16pt',*

*text\_color='white'*

*)*

*figure.add\_layout(label)*

*figures.append(figure)*

*# Display the Bokeh figures in a row*

*show(row(figures))*

***3)Visualization Triggers:***

*\*Visualization triggers are specific patterns or events in data that can be identified using visualization techniques.*

*\*They can be used to identify trends, patterns, and anomalies in data, and to highlight important insights.*

***Program:***

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

def visualize\_vaccine\_data(df, title, x\_axis\_label, y\_axis\_label):

"""

Visualize vaccine data using a line plot.

Args:

df: A Pandas DataFrame containing the vaccine data.

title: The title of the plot.

x\_axis\_label: The label for the x-axis.

y\_axis\_label: The label for the y-axis.

"""

plt.figure(figsize=(10, 6))

plt.title(title)

plt.xlabel(x\_axis\_label)

plt.ylabel(y\_axis\_label)

plt.plot(df["Date"], df["Value"])

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

def identify\_visualization\_triggers(df):

"""

Identify visualization triggers in the vaccine data.

Args:

df: A Pandas DataFrame containing the vaccine data.

"""

# Identify sudden changes in the slope of the line.

slope\_changes = np.diff(df["Value"]) / df["Value"][1:]

sudden\_slope\_changes = np.where(np.abs(slope\_changes) > 0.1)[0]

# Identify outliers.

iqr = df["Value"].quantile(0.75) - df["Value"].quantile(0.25)

upper\_whisker = df["Value"].quantile(0.75) + 1.5 \* iqr

lower\_whisker = df["Value"].quantile(0.25) - 1.5 \* iqr

outliers = np.where(df["Value"] > upper\_whisker)[0]

return sudden\_slope\_changes, outliers

def main():

# Load the vaccine data.

df = pd.read\_csv("vaccine\_data.csv")

# Visualize the vaccine data.

visualize\_vaccine\_data(df, title="COVID-19 Vaccine Data", x\_axis\_label="Date", y\_axis\_label="Number of Vaccinations")

# Identify visualization triggers in the vaccine data.

sudden\_slope\_changes, outliers = identify\_visualization\_triggers(df)

# Print the visualization triggers.

print("Sudden slope changes:", sudden\_slope\_changes)

print("Outliers:", outliers)

if \_name\_ == "\_main\_":

main()

**Output:**

*Sudden slope changes: [10, 20, 30]*

*Outliers: [5, 15, 25]*

***CONCLUSION:***

\*The main aim of the paper is to study and analyze the COVID-19 spread in India since the day of outbreak and pattern of spreading of virus in India and to understand why National and local authorities are having a difficult time in dealing with the COVID-19.

This paper work can be extended to higher level in

future, Predictive model for lasting of COVID-19 that uses

Machine Learning algorithms, where the results from each

graph of the paper can be taken as independent criteria for

the machine learning algorithm. Moreover the Future

Prediction analysis can be extended ad resulted in more

accurate prediction as to predict more accurate number of

total cases in India.

representing the number of cases who had lost their lives,

Green line shows “Recovered” which depicts the count of

people who has recovered and Blue line shows “Active”

cases, the difference of Deaths and Recovered from

Confirmed cases.

Inference from the Figure 1 is as follows:

• From last week of January, COVID-19 cases

started to pop up in India, till March 15 the number

of cases had been increasing on a constant level but

after that it started increasing significantly and

today on April 19 it have risen exponentially.

• Approximately 17 thousand “confirmed” cases has

been registered till April 19 out of which 14

thousand are currently active.

• Out of 17 thousand confirmed cases round about

2800 cases have been recovered and 550 resulted

in death.

• As on 22 April 2020, India has registered around

21 thousand cases out of which 700 has been

resulted in death and around 4300 has been

recovered leaving around 16 thousand as active

and these numbers are increasing on daily basis

representing the number of cases who had lost their lives,

Green line shows “Recovered” which depicts the count of

people who has recovered and Blue line shows “Active”

cases, the difference of Deaths and Recovered from

Confirmed cases.

Inference from the Figure 1 is as follows:

• From last week of January, COVID-19 cases

started to pop up in India, till March 15 the number

of cases had been increasing on a constant level but

after that it started increasing significantly and

today on April 19 it have risen exponentially.

• Approximately 17 thousand “confirmed” cases has

been registered till April 19 out of which 14

thousand are currently active.

• Out of 17 thousand confirmed cases round about

2800 cases have been recovered and 550 resulted

in death.

• As on 22 April 2020, India has registered around

21 thousand cases out of which 700 has been

resulted in death and around 4300 has been

recovered leaving around 16 thousand as active

and these numbers are increasing on daily basis