A
Mini Project Report
On
"Covid Vaccine Analytics"

NGSPM'S
BRAHMA VALLEY COLLEGE OF ENGINEERING AND RESEARCH INSTITUTE NASHIK

NAAC ACCREDITED INSTITUTE WITH "A" GRADE

Department of Computer Engineering

Submitted By

Mr. Akhilesh Jadhav

Under the guidance of

Prof. M. S. Salve

Submitted in partial fulfillment of Bachelor of Engineering – Computer Engineering, bySavitribai Phule Pune University

Savitribai Phule Pune University

A.Y 2024-25

NGSPM'S
BRAHMA VALLEY COLLEGE OF ENGINEERING AND RESEARCH INSTITUTE NASHIK, Nashik- 422013

NAAC ACCREDITED INSTITUTE WITH "A" GRADE

C E R T I F I C A T E

This is to certify that ***Mr. Akhilesh Keru Jadhav*** have successfully completed the Mini Project of Web Technology entitled **"Covid Vaccine Analytics"** in the partial fulfillment of Bachelor of Engineering – Computer Engineering of Savitribai Phule Pune University, Pune.

Date:

Place: Nashik

Prof. M. S. Salve	**Prof. P.P. Kakade**	**Dr. H. N. Kudal**
Guide	**Head Of Department**	**Principle**

ACKNOWLEDGEMENTS

I wish to express my sincere Gratitude to my Guide **Prof. M. S. Salve** of Computer Engineering Department. **Brahma Valley College Of Engineering, Nashik** for providing invaluable guidance and encouraging me to come up with such potential subject. I express my deep thanks to all the faculty members of Computer Engineering Department for providing the useful guidance and feel privileged to express my gratitude towards them.

I am also thankful to our respected **Prof. P. P. Kakade, Head of Department (Computer)** who constantly guided me throughout my thesis work.

I would like to thank our Respected **Prof. H. N. Kudal, Brahma Valley College Of Engineering, Nashik** for their encouragement at every step and those who helped me directly or indirectly this project.

Mr. Akhilesh Jadhav

INDEX

ABSTRACT

In India, a large country of about 1.3 billion people, the disease was first detected on January 30, 2020, in a student returning from Wuhan. The total number of confirmed infections in India as of May 3, 2020, is more than 37,000 and is currently growing fast. Most of the prior research and media coverage focused on the number of infections in the entire country. However, given the size and diversity of India, it is important to look at the spread of the disease in each state separately, where in the situations are quite different. In this report, we aim to analyse data on the number of infected people in each Indian state using csv dataset and predict the number of vaccinations for that state. We hope that such state wise predictions would help the state governments better channelize their limited health care resources.

Additionally, the report addresses challenges and obstacles encountered during the vaccination rollout, such as supply chain disruptions, vaccine hesitancy, and equity concerns. Through predictive modeling and machine learning algorithms, we anticipate future trends in vaccination uptake and identify strategies to overcome barriers to achieving herd immunity.

Overall, this project report provides valuable insights into the Covid-19 vaccination campaign's progress and effectiveness, offering recommendations for policymakers, healthcare providers, and public health officials to optimize vaccination efforts and combat the ongoing pandemic effectively.

Keywords:

Prediction, Analysis, Data Visualization, Data Cleaning, Vaccination, administered, Data Handling

SOFTWARE REQUIREMENT

- Operating System: Windows
- Jupyter Notebook
- Python

INTRODUCTION

The highly infectious coronavirus disease (COVID-19) was first detected in Wuhan, China in December 2019 and subsequently spread to 212 countries and territories around the world, infecting millions of people. In India, a large country of about 1.3 billion people, the disease was first detected on January 30, 2020, in a student returning from Wuhan. The total number of confirmed infections in India as of August 9, 2021, is more than 37,000 and is currently growing fast. An effective rollout of vaccinations against COVID-19 offers the most promising prospect of bringing the pandemic to an end. We present the Our World in Data COVID19 vaccination dataset, a global public dataset that tracks the scale and rate of the vaccine rollout across the Country. This dataset is updated regularly and includes data on the total number of vaccinations administered, first and second doses administered, Male (Doses Administered), Female (Doses Administered), Transgender (Doses Administered) for which data are available (28 state as of 9 August 2021). It will be maintained as the global vaccination campaign continues to progress. Our intention is to maintain the database for the foreseeable future and include additional State as they implement their vaccination campaigns. This dataset tracks the total number of COVID- 19 vaccinations administered, Number of persons state wise vaccinated for first dose in India Number of persons state wise vaccinated for second dose in India Number of persons state wise vaccinated for second dose in India, Number of Males vaccinated, Number of females vaccinated each State. In this project using python libraries doing various operation on state wise covid 19 vaccination dataset and in this project use csv file dataset.

The emergence of the Covid-19 pandemic in late 2019 presented an unprecedented global health crisis, challenging governments, healthcare systems, and communities worldwide. In response, the scientific community mobilized with remarkable speed to develop vaccines against the novel coronavirus, leading to the approval and distribution of multiple vaccines in record time. By leveraging advanced data analytics techniques, we seek to explore various dimensions of the vaccination campaign, including coverage rates, distribution strategies, effectiveness, and associated challenges.

In this project use some libraries for analyzing and predicting data for analyzing data, we need some libraries. In this section, we are importing all the required libraries like pandas, NumPy, matplotlib, pyplot, seaborn, and word cloud that are required for data analysis

Problem Statement:

Use the following covid_vaccine_statewise.csv dataset and perform following analytics on the given dataset

https://www.kaggle.com/sudalairajkumar/covid19-in-india?select=covid_vaccine_statewise.csv

Describe the dataset

Number of persons state wise vaccinated for first dose in India

Number of persons state wise vaccinated for second dose in India

Number of Males vaccinated.

Objective:

The main objective of the project on Covid19 Vaccination Analysis and Prediction is to manage the details of state wise vaccination. It manages all the information about the individual males and females, types of covid vaccine, total number of covid vaccine The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to analyse and predict the covid19 vaccination to reduce the manual work for managing the course, prediction, Result. It tracks all the details about the male, female and total vaccination.

Functionalities provided by Covid19 Vaccination Analysis and Prediction are as follows:

Covid19 Vaccination Analysis and Prediction also manages the details for state wise total males and female fully vaccinated.

It tracks all the information of question, covid 19 vaccination etc. Manage the information and description of the country vaccination

Outcome:

Covid-19 Vaccination Analytics is done on covid_vaccine_statewise.csv dataset.

THEORY

Importing Python Libraries:

import numpy as np-

NumPy (Numerical Python) is an open-source library for the Python programming language. It is used for scientific computing and working with arrays. Apart from its multidimensional array object, it also provides high-level functioning tools for working with arrays. In this tutorial, you will learn how to install NumPy.

import pandas as pd pandas-

It is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant.

import matplotlib as plt-

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Read Data and Basic Information:

Read the CSV file covid_vaccine_statewise.csv using pandas read_csv() function and show the output using head() function.

Data Cleaning:

Dataset has many null values as we have seen before. To get rid of it we need to clean the data first, after cleaning we will perform our further analysis. For cleaning the dataset, we will perform many steps. Some of these steps are shown below

Data Pre-processing:

In this section, we are going to draw some visuals to get insights from our dataset. describe() function in pandas used to get the statistics of each feature present in our dataset. Some of the information we get include count, max, min, standard deviation, median, etc.

Commands & Output

Import basic required libraries

#import the required libraries import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

[3]:
```
# import the required libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Reading the dataset

# for reading the csv file
df = pd.read_csv("covid_vaccine_statewise.csv")

returns first five rows of the table df.head()
returns last five rows of columns df.tail()

[4]:
```
# for reading the csv file
df = pd.read_csv("covid_vaccine_statewise.csv")
```

[5]:
```
# returns first five rows of the table
df.head()
```

[5]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)	45-60 Years (Doses Administered)	Ad
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN	...	NaN	NaN	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN	...	NaN	NaN	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN	...	NaN	NaN	
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN	...	NaN	NaN	
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN	...	NaN	NaN	

5 rows × 24 columns

[6]:
```
# returns last five rows of columns
df.tail()
```

[6]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)	45-60 Years (Doses Administered)	A
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	

Display no. of rows and columns

returns the shape of the dataset in the format of (rows, columns) df.shape (7845, 24)

returns the names of all columns names df.columns

[7]:
```
# returns the shape of the dataset in the format of (rows, columns)
df.shape
```

[7]:
```
(7845, 24)
```

[8]:
```
# returns the names of all columns names
df.columns
```

[8]:
```
Index(['Updated On', 'State', 'Total Doses Administered', 'Sessions',
       ' Sites ', 'First Dose Administered', 'Second Dose Administered',
       'Male (Doses Administered)', 'Female (Doses Administered)',
       'Transgender (Doses Administered)', ' Covaxin (Doses Administered)',
       'CoviShield (Doses Administered)', 'Sputnik V (Doses Administered)',
       'AEFI', '18-44 Years (Doses Administered)',
       '45-60 Years (Doses Administered)', '60+ Years (Doses Administered)',
       '18-44 Years(Individuals Vaccinated)',
       '45-60 Years(Individuals Vaccinated)',
       '60+ Years(Individuals Vaccinated)', 'Male(Individuals Vaccinated)',
       'Female(Individuals Vaccinated)', 'Transgender(Individuals Vaccinated)',
       'Total Individuals Vaccinated'],
      dtype='object')
```

Display dataset Information and finding null values
returns the help information for a function,class & module df.info()

[11]:
```
# returns the help information for a function,class & module
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7845 entries, 0 to 7844
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Updated On                            7845 non-null   object
 1   State                                 7845 non-null   object
 2   Total Doses Administered              7621 non-null   float64
 3   Sessions                              7621 non-null   float64
 4    Sites                                7621 non-null   float64
 5   First Dose Administered               7621 non-null   float64
 6   Second Dose Administered              7621 non-null   float64
 7   Male (Doses Administered)             7461 non-null   float64
 8   Female (Doses Administered)           7461 non-null   float64
 9   Transgender (Doses Administered)      7461 non-null   float64
 10   Covaxin (Doses Administered)         7621 non-null   float64
 11  CoviShield (Doses Administered)       7621 non-null   float64
 12  Sputnik V (Doses Administered)        2995 non-null   float64
 13  AEFI                                  5438 non-null   float64
 14  18-44 Years (Doses Administered)      1702 non-null   float64
 15  45-60 Years (Doses Administered)      1702 non-null   float64
 16  60+ Years (Doses Administered)        1702 non-null   float64
 17  18-44 Years(Individuals Vaccinated)   3733 non-null   float64
 18  45-60 Years(Individuals Vaccinated)   3734 non-null   float64
 19  60+ Years(Individuals Vaccinated)     3734 non-null   float64
 20  Male(Individuals Vaccinated)          160 non-null    float64
 21  Female(Individuals Vaccinated)        160 non-null    float64
 22  Transgender(Individuals Vaccinated)   160 non-null    float64
 23  Total Individuals Vaccinated          5919 non-null   float64
dtypes: float64(22), object(2)
memory usage: 1.4+ MB
```

#returns count of null values in columns df.isnull().sum()

[12]:
```
#returns count of null values in columns
df.isnull().sum()
```

[12]:
```
Updated On                            0
State                                 0
Total Doses Administered            224
Sessions                            224
 Sites                              224
First Dose Administered             224
Second Dose Administered            224
Male (Doses Administered)           384
Female (Doses Administered)         384
Transgender (Doses Administered)    384
 Covaxin (Doses Administered)       224
CoviShield (Doses Administered)     224
Sputnik V (Doses Administered)     4850
AEFI                               2407
18-44 Years (Doses Administered)   6143
45-60 Years (Doses Administered)   6143
60+ Years (Doses Administered)     6143
18-44 Years(Individuals Vaccinated) 4112
45-60 Years(Individuals Vaccinated) 4111
60+ Years(Individuals Vaccinated)  4111
Male(Individuals Vaccinated)       7685
Female(Individuals Vaccinated)     7685
Transgender(Individuals Vaccinated) 7685
Total Individuals Vaccinated       1926
dtype: int64
```

Describe the dataset: df.describe()

Describe the dataset

[9]:
```
#provides descriptive statistics (central tendency dispersion & shape)
df.describe()
```

[9]:

	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	Covaxin (Doses Administered)	CoviShield (Doses Administered)	...	18-44 (Adminis
count	7.621000e+03	7.621000e+03	7621.000000	7.621000e+03	7.621000e+03	7.461000e+03	7.461000e+03	7461.000000	7.621000e+03	7.621000e+03	...	1.70200
mean	9.188171e+06	4.792358e+05	2282.872064	7.414415e+06	1.773755e+06	3.620156e+06	3.168416e+06	1162.978019	1.044669e+06	8.126553e+06	...	8.77395
std	3.746180e+07	1.911511e+06	7275.973730	2.995209e+07	7.570382e+06	1.737938e+07	1.515310e+07	5931.353995	4.452259e+06	3.298414e+07	...	2.66082
min	7.000000e+00	0.000000e+00	0.000000	7.000000e+00	0.000000e+00	0.000000e+00	2.000000e+00	0.000000	0.000000e+00	7.000000e+00	...	2.66240
25%	1.356570e+05	6.004000e+03	69.000000	1.166320e+05	1.283100e+04	5.655500e+04	5.210700e+04	8.000000	0.000000e+00	1.331340e+05	...	4.34484
50%	8.182020e+05	4.547000e+04	597.000000	6.614590e+05	1.388180e+05	3.897850e+05	3.342380e+05	113.000000	1.185100e+04	7.567360e+05	...	3.09597
75%	6.625243e+06	3.428690e+05	1708.000000	5.387805e+06	1.166434e+06	2.735777e+06	2.561513e+06	800.000000	7.579300e+05	6.007817e+06	...	7.36624
max	5.132284e+08	3.501031e+07	73933.000000	4.001504e+08	1.130780e+08	2.701636e+08	2.395186e+08	98275.000000	6.236742e+07	4.468251e+08	...	2.24330

8 rows × 22 columns

[10]:
```
df.describe(include='object')
```

[10]:

	Updated On	State
count	7845	7845
unique	213	37
top	16/01/2021	Delhi

Number of persons state wise vaccinated for first dose in India:

```
# returns the average of First Dose Administered
avg_firstdose = df["First Dose Administered"].astype("float").mean()
print("Average of First Dose:", avg_firstdose)

# Replacing First Dose Administered
df["First Dose Administered"].fillna(value = avg_firstdose, inplace=True) df
```

Number of persons state wise vaccinated for first dose in India

```
[13]: # returns the average of First Dose Administered
avg_firstdose = df["First Dose Administered"].astype("float").mean()
print("Average of First Dose:", avg_firstdose)
```

```
Average of First Dose: 7414415.300354284
```

```
[14]: # Replacing First Dose Administered
df["First Dose Administered"].fillna(value = avg_firstdose, inplace=True)
df
```

```
C:\Users\anike\AppData\Local\Temp\ipykernel_4608\2535364140.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
hained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inste
ad, to perform the operation inplace on the original object.


  df["First Dose Administered"].fillna(value = avg_firstdose, inplace=True)
```

[14]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)	45-60 Years (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	4.827600e+04	0.0	NaN	NaN	NaN	...	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	5.860400e+04	0.0	NaN	NaN	NaN	...	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	9.944900e+04	0.0	NaN	NaN	NaN	...	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	1.955250e+05	0.0	NaN	NaN	NaN	...	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	2.512800e+05	0.0	NaN	NaN	NaN	...	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Number of persons state wise vaccinated for second dose in India:

returns the average of second dose administered

avg_seconddose = df["Second Dose Administered"].astype("float").mean(axis = 0) print("Average of Second Dose:", avg_seconddose)

Replacing Second Dose Administered

df["Second Dose Administered"].fillna(value = avg_seconddose, inplace = True) df

For Second Dose Administered

```
[15]: # returns the average of second dose administered
avg_seconddose = df["Second Dose Administered"].astype("float").mean(axis = 0)
print("Average of Second Dose:", avg_seconddose)
```

```
Average of Second Dose: 1773755.2436688098
```

```
[16]: # Replacing Second Dose Administered
df["Second Dose Administered"].fillna(value = avg_seconddose, inplace = True)
df
```

```
C:\Users\anike\AppData\Local\Temp\ipykernel_4608\1133825595.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
hained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inste
ad, to perform the operation inplace on the original object.


  df["Second Dose Administered"].fillna(value = avg_seconddose, inplace = True)
```

[16]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)	45-60 Years (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	4.827600e+04	0.000000e+00	NaN	NaN	NaN	...	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	5.860400e+04	0.000000e+00	NaN	NaN	NaN	...	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	9.944900e+04	0.000000e+00	NaN	NaN	NaN	...	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	1.955250e+05	0.000000e+00	NaN	NaN	NaN	...	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	2.512800e+05	0.000000e+00	NaN	NaN	NaN	...	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Number of persons state wise vaccinated for first dose in India:

first_dose = df.groupby('State')[['First Dose Administered']].sum()

first_dosecond_dose = df.groupby('State')[['‘Second Dose Administered']].sum()

Number of persons state wise vaccinated for first dose in India

```
[18]: first_dose = df.groupby('State')[['First Dose Administered']].sum()
      first_dose
```

[18]:

State	First Dose Administered
Andaman and Nicobar Islands	6.091235e+07
Andhra Pradesh	1.277347e+09
Arunachal Pradesh	9.349147e+07
Assam	6.300867e+08
Bihar	1.514989e+09
Chandigarh	8.918960e+07
Chhattisgarh	8.404894e+08
Dadra and Nagar Haveli and Daman and Diu	8.549597e+07
Delhi	6.762404e+08
Goa	1.204779e+08
Gujarat	2.176133e+09
Haryana	8.002848e+08
Himachal Pradesh	3.607805e+08
India	2.830663e+10
Jammu and Kashmir	4.545883e+08

Number of persons state wise vaccinated for second dose in India: second_dose = df.groupby('State')[['Second Dose Administered']].sum() second_dose

Number of persons state wise vaccinated for second dose in India

```
[22]: second_dose = df.groupby('State')[['Second Dose Administered']].sum()
      second_dose
```

[22]:

State	Second Dose Administered
Andaman and Nicobar Islands	1.476109e+07
Andhra Pradesh	3.694601e+08
Arunachal Pradesh	2.257485e+07
Assam	1.414313e+08
Bihar	2.814331e+08
Chandigarh	2.223627e+07
Chhattisgarh	1.827629e+08
Dadra and Nagar Haveli and Daman and Diu	1.701070e+07
Delhi	2.006352e+08
Goa	2.684071e+07
Gujarat	6.110609e+08
Haryana	1.692986e+08
Himachal Pradesh	8.448111e+07
India	6.770264e+09
Jammu and Kashmir	9.659418e+07
Jharkhand	1.327636e+08

Number of Males & Female Vaccinated:
male = df["Male(Individuals Vaccinated)"].sum()
print("The total number of male individuals vaccinated are", int(male))

female = df["Female(Individuals Vaccinated)"].sum()
print("The total number of female individuals vaccinated are", int(female))

Number of Males vaccinated

```
[20]: male = df["Male(Individuals Vaccinated)"].sum()
      print("The total number of male individuals vaccinated are", int(male))
```

```
The total number of male individuals vaccinated are 7138698858
```

Number of females vaccinated

```
[21]: female = df["Female(Individuals Vaccinated)"].sum()
      print("The total number of female individuals vaccinated are", int(female))
```

```
The total number of female individuals vaccinated are 6321628736
```

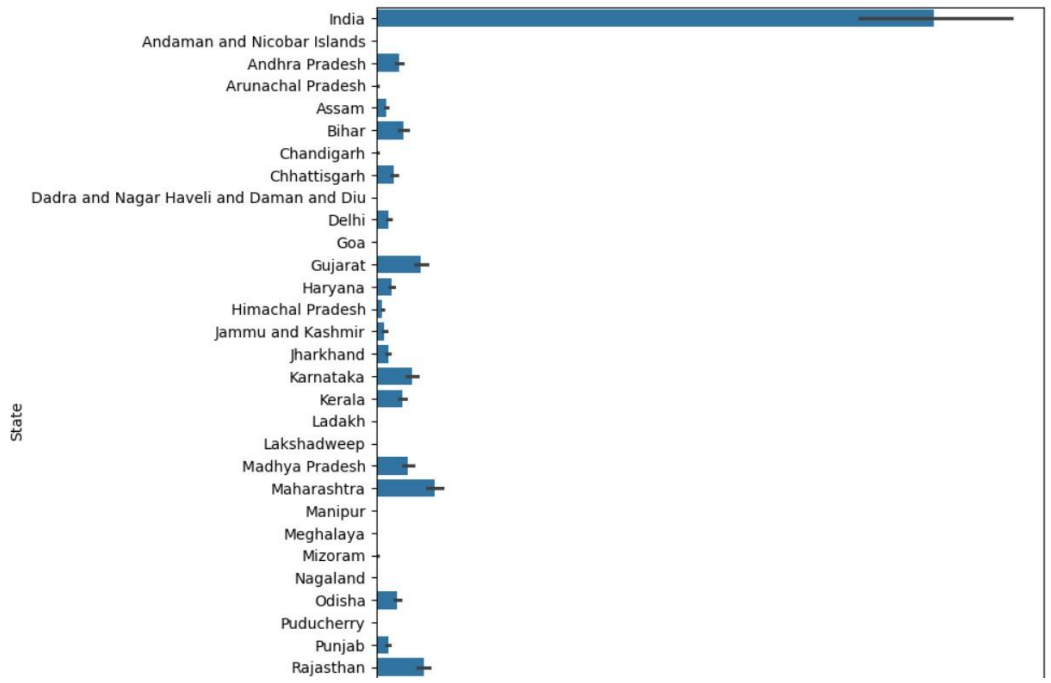
Data Visualization:

#display bar plot plt.figure(figsize=(8, 10))
sns.barplot(data = df, y="State", x="Total Individuals Vaccinated")

Data Visualization

```
[45]: plt.figure(figsize=(8, 10))
      sns.barplot(data = df, y="State", x="Total Individuals Vaccinated")
```

```
[45]: <Axes: xlabel='Total Individuals Vaccinated', ylabel='State'>
```



males_vaccinated=df['Male (Doses Administered)'].sum() females_vaccinated=df['Female (Doses Administered)'].sum() print("Numbers of Males Vaccinated:", males_vaccinated) print("Numbers of Females Vaccinated:", females_vaccinated)

labels=['Male','Female']

sizes=[males_vaccinated, females_vaccinated] colors=['cyan','salmon']

plt.figure(figsize=(4,4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140) plt.title('Total Numbers of Males and Females Vaccinated')
plt.axis('equal')
#show the plot plt.tight_layout() plt.show()

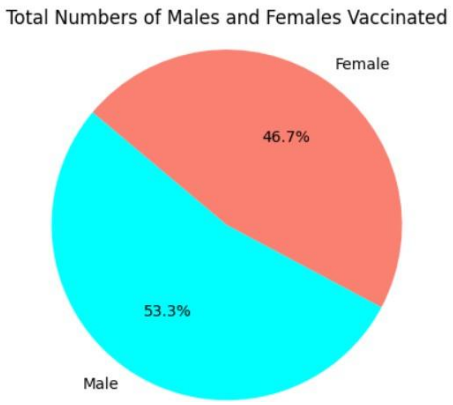
```
[46]: males_vaccinated=df['Male (Doses Administered)'].sum()
females_vaccinated=df['Female (Doses Administered)'].sum()
print("Numbers of Males Vaccinated:", males_vaccinated)
print("Numbers of Females Vaccinated:", females_vaccinated)

labels=['Male','Female']
sizes=[males_vaccinated, females_vaccinated]
colors=['cyan','salmon']

plt.figure(figsize=(4,4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title('Total Numbers of Males and Females Vaccinated')
plt.axis('equal')

#show the plot
plt.tight_layout()
plt.show()
```

```
Numbers of Males Vaccinated: 27009983996.0
Numbers of Females Vaccinated: 23639554465.0
```



first=df['First Dose Administered'].sum() second=df['Second Dose Administered'].sum() print("Numbers of First Dose Administered:", first) print("Numbers of Second Dose Administered:", second)
labels=['First Dose Administered','Second Dose Administered'] sizes=[first, second]
colors=['green','red']
plt.figure(figsize=(4,4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140) plt.title('Total Numbers First and Second Dose Administered')
plt.axis('equal')

#show the plot

plt.tight_layout() plt.show()

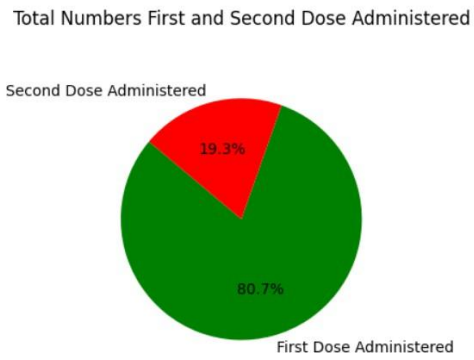
[57]:
```
first=df['First Dose Administered'].sum()
second=df['Second Dose Administered'].sum()
print("Numbers of First Dose Administered:", first)
print("Numbers of Second Dose Administered:", second)

labels=['First Dose Administered','Second Dose Administered']
sizes=[first, second]
colors=['green','red']

plt.figure(figsize=(4,4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title('Total Numbers First and Second Dose Administered')
plt.axis('equal')

#show the plot
plt.tight_layout()
plt.show()
```

```
Numbers of First Dose Administered: 58166088031.27935
Numbers of Second Dose Administered: 13915109886.581814
```



CONCLUSION

The COVID vaccination dataset provides vaccination status across India, detailing male and female vaccination by state. Through visualization, we can analyze insights such as minimum, maximum, count, standard deviation, and mean vaccination rates. To declare the end of the pandemic, we look for a consistent decrease in the Daily Infection Rate (DIR) over 14 days until it reaches zero or turns negative.

REFERENCES

https://www.w3schools.com/python/

https://www.kaggle.com/sudalairajkumar/covid19-in india?select=covid_vaccine_statewise.csv

https://www.kaggle.com/sudalairajkumar/covid19-in-india?select=covid_vaccine_statewise.csv

https://www.tableau.com/learn/articles/data-visualization#:~:text=Data%20visualization%20is%20the%20graphical,outliers%2C%20a nd%20patterns%20in%20data.

https://www.python.org/