Pune Institute of Computer Technology, Pune

DSBDA Mini Project
(AY 2022-23)

Class: T.E. 2
Batch: K2

Mini-Project

On

Analysis of Covid Pandemic Response in India

Group Members:

31302: Amrit Singh
31318: Jaydeep Datre

Guided by

Prof. Kimaya Urane

Title: Analysis of Covid Pandemic Response in India

Problem Definition:

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

a. Describe the dataset
b. Number of persons state-wise vaccinated for the first dose in India
c. Number of persons state-wise vaccinated for the second dose
d. Number of Males vaccinated
e. Number of females vaccinated

Objectives:

1. To understand how to load and pre-process a given dataset.
2. To understand how to remove useless, wrong, or null values.
3. To create visualizations to find relationships among the data.
4. To perform analysis using various techniques to find the relation between dependent and independent variables.

Outcomes:

After completion of the project, students will be able to:

1. Load and pre-process data, and remove unwanted and null values.
2. Categorize and rename columns, and set up the dataset.
3. Use data visualization and find relationships in the data.
4. Find the effectiveness of the Anti-Covid drive implemented in India.

Requirements:

- Computer System with:
- I5 processor, 256 GB SSD, 8GB RAM. ● Jupyter Notebook
- Python with numpy, pandas, matplotlib

Abstract:
This project aimed to analyze and visualize data from India's Covid19 vaccination drive to identify trends and inefficiencies in the program. We used various data visualization and analysis techniques, including linear regression, to explore the relationships between different variables and the number of successfully vaccinated individuals.

Introduction:

The Covid pandemic has brought about unprecedented changes in the lives of people around the world, including India. With the sudden
emergence of the virus, the Indian government has had to take a number of measures to control the spread of the virus, including a vaccination drive to ensure that every citizen of India has access to safe and free vaccines. However, there are concerns about the efficiency of the immunization program, with questions about the proportion of people receiving vaccines properly and the people who are being treated unfairly or not given the opportunity for immunization.

The vaccination drive is a crucial aspect of India's fight against the Covid pandemic, but it is important to understand how well the program is being implemented to ensure that the maximum number of people are being vaccinated in an efficient and fair manner. The objective of our data analytics project is to analyze and visualize the available data to find out the efficiency of the immunization drive and identify state-wise trends in vaccination.

To achieve this objective, we will be exploring trends in the number of successfully vaccinated individuals based on various factors such as age, gender, state of residence, the vaccine manufacturer, and other important variables. By aggregating and properly visualizing this data, we can gain insights into whether some people have been deprived of proper healthcare due to poor infrastructure, prejudices, or other reasons.

Through our data analysis, we hope to provide valuable insights to policymakers and health professionals, enabling them to identify areas of improvement and take appropriate measures to ensure that the vaccination drive is implemented in the most efficient and equitable manner possible. Ultimately, our project aims to contribute to India's efforts to combat the Covid pandemic and protect its citizens.

Methodology:

Softwares Used:

[1] NumPy:

NumPy is a widely used package in Python for scientific computing and data analysis. Its core feature is the N-dimensional array object, which is the basis for many numerical computing applications in Python. NumPy provides a variety of tools for performing operations on arrays, such as mathematical, logical, and shape manipulation operations. It also includes useful tools for linear algebra, Fourier analysis, and random number generation.

[2] Pandas:

Pandas is a powerful Python package for data manipulation and analysis. It provides two primary data structures: Series and DataFrame. A Series is a onedimensional array-like object that can hold any data type, while a data frame is a two-dimensional table-like data structure with labeled axes.

Pandas are particularly useful for handling tabular data, such as data from spreadsheets and SQL tables. It provides a wide range of tools for data cleaning, merging, reshaping, slicing, indexing, and aggregation. Pandas is also capable of handling time-series data and providing tools for working with dates and times.

[3] Matplotlib.pyplot:

Matplotlib.pyplot is a popular Python package for creating visualizations and plots. It provides a range of functions for creating various types of plots, including line plots, scatter plots, bar plots, histograms, and many more.

Matplotlib. pyplot is highly customizable, and users can modify almost every aspect of a plot, such as colors, labels, axis limits, and annotations. It is widely used in the

scientific and engineering communities for creating high-quality visualizations of data.

[4] Seaborn:

Seaborn is a Python data visualization library built on top of matplotlib. It provides a high-level interface for creating statistical graphics. Seaborn is particularly useful for visualizing complex statistical relationships, such as regression models, categorical data, and time-series data. It includes many built-in themes and color palettes that make it easy to create aesthetically pleasing plots. Seaborn is widely used in data science and machine learning applications.

[5] Scikit-learn:

Scikit-learn is a popular open-source Python library that provides a range of tools for data analysis, machine learning, and artificial intelligence. It is built on top of other scientific computing libraries in Python, such as NumPy, SciPy, and Matplotlib.

Scikit-learn offers a wide range of machine learning algorithms, including supervised and unsupervised learning algorithms, clustering, regression, classification, and dimensionality reduction, among others. It also provides tools for data preprocessing, feature extraction, and data visualization.

Implementation:

The actual implementation of the project involved various steps. According to the data analysis life cycle, different steps were implemented

1. **Data Collection**:

   A dataset called covid_19_india.csv and another dataset called covid_vaccine_statewise.csv was retrieved from Kaggle and used for the project.

2. **Data Processing**:

   Many pre-processing tasks were done such as renaming the columns to increase the ease of coding tasks, checking for null values, removing columns having the same value for each record, removing null values, and replacing them with appropriate values (either 0 or mean of the column). We also checked for outliers or errors in data and accordingly removed those outliers by using Z square method.

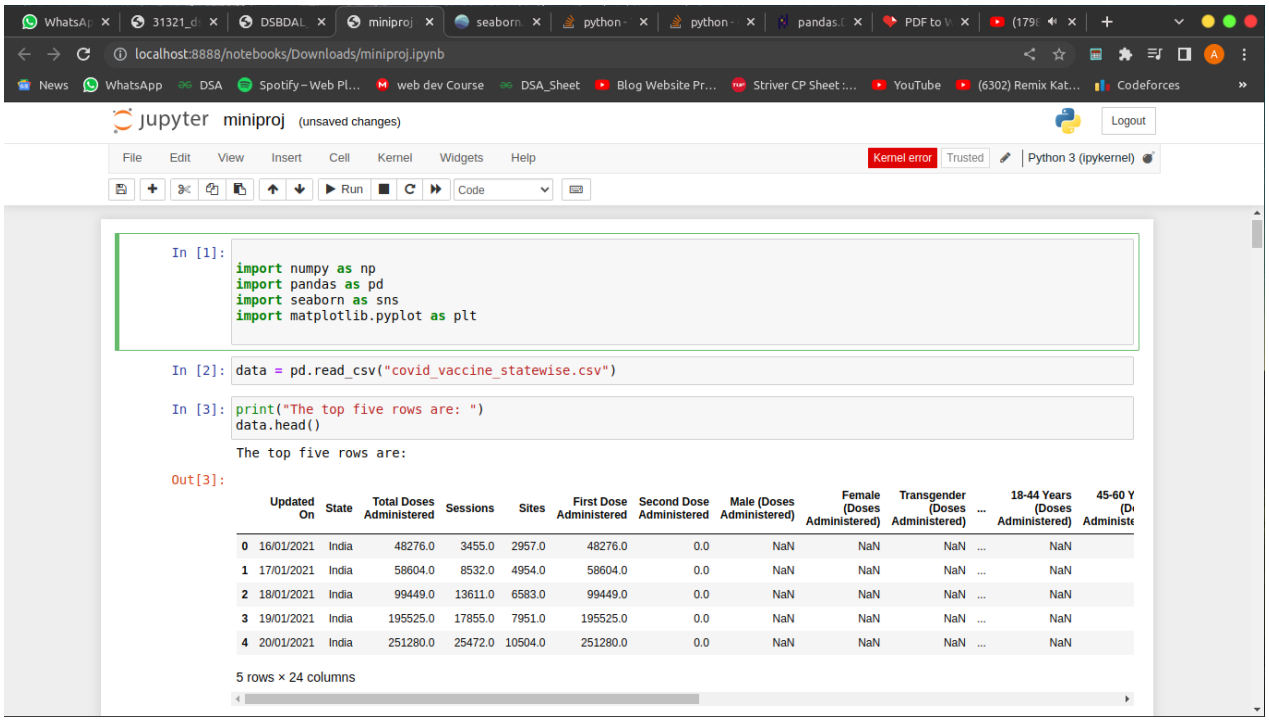
3. **Data visualization**

   After properly processing the data, we started creating visualizations to find a preliminary shape and structure as well s trends in the data. We initially created a graph of Total Doses Administered vs Date, then we created more useful and informative graphs such as Statewise doses administered (used lineplot to plot the data of all the states on the same graph for comparative analysis). We also plotted graphs to find the relation between ages and genders vs the number of vaccines administered by date.

4. **Data Analysis**:

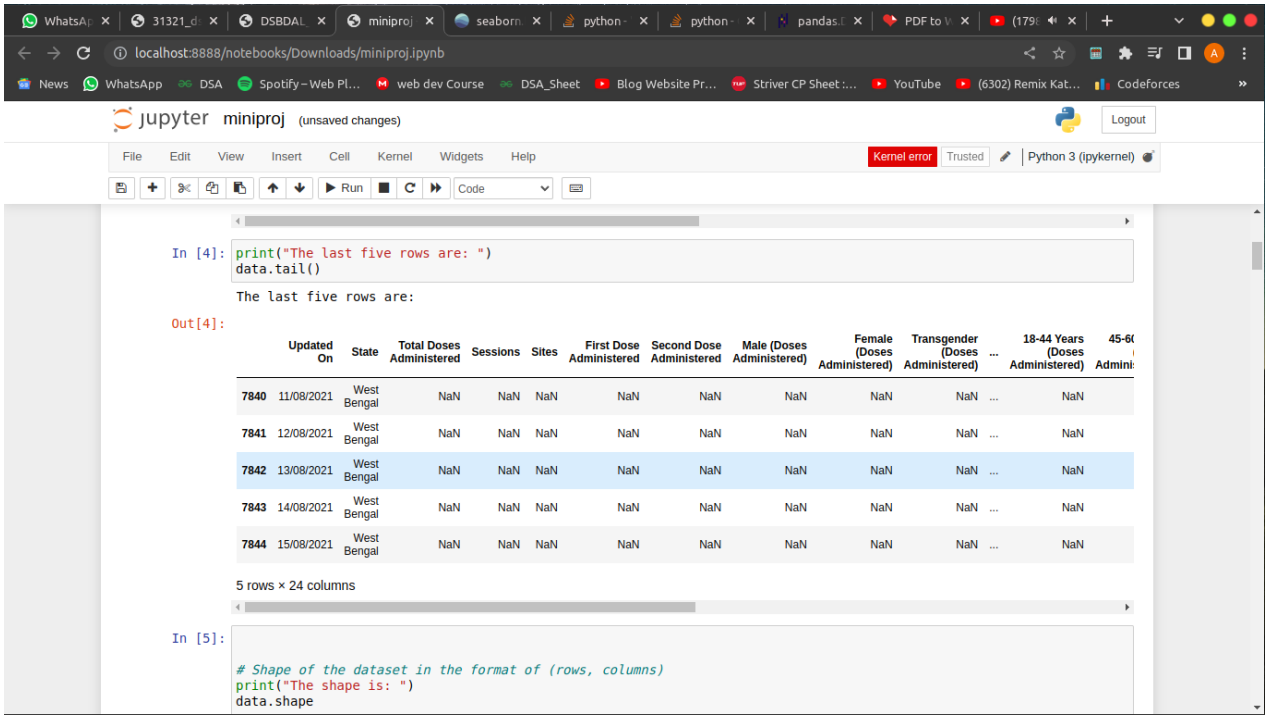
We used Scikit-learn's inbuilt LinearRegression module to find relations between the independent variable (Date) and the dependent variable (Number of doses administered). We also added extra criteria such as the state, the age of the person, and the gender to find out how the analysis varied.

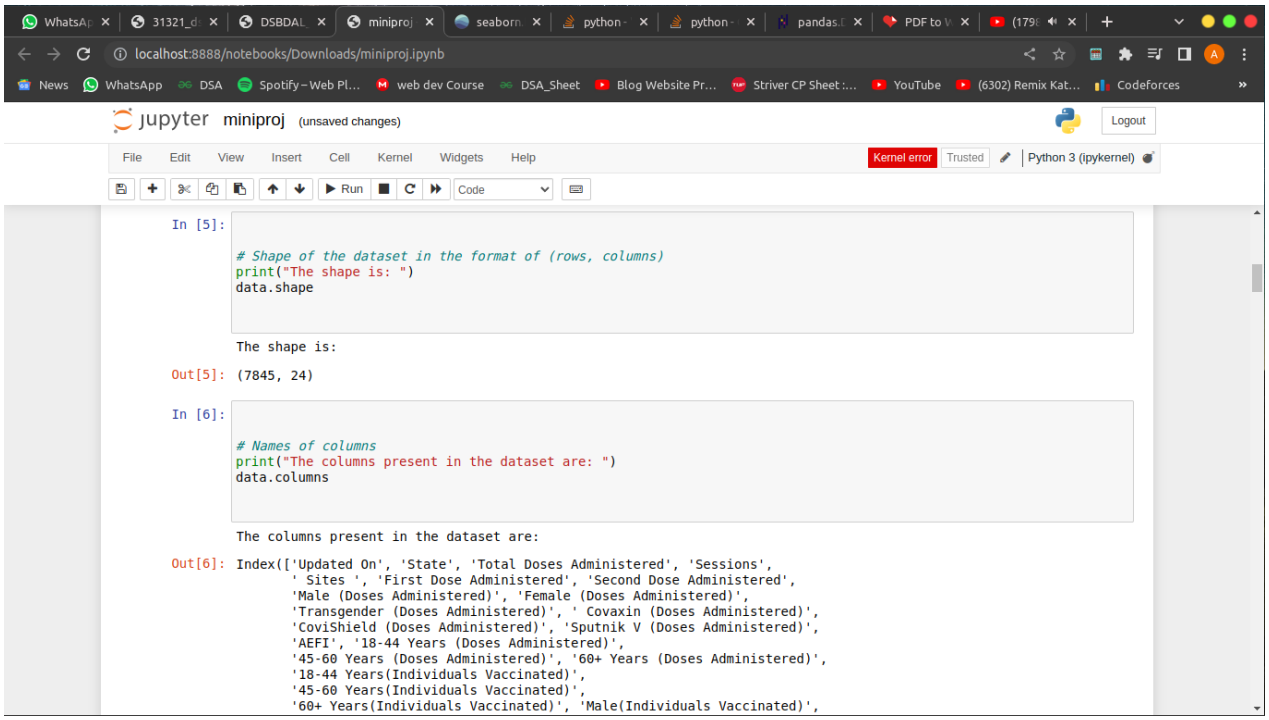
Code & Output:

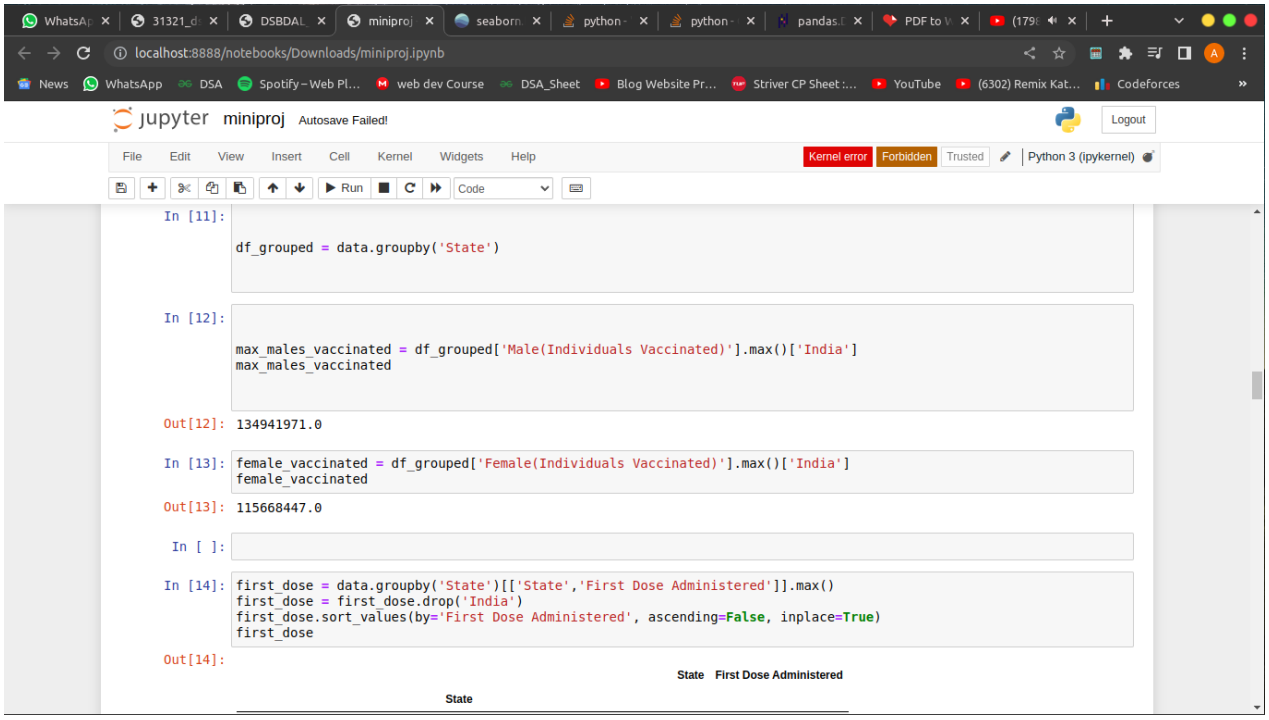


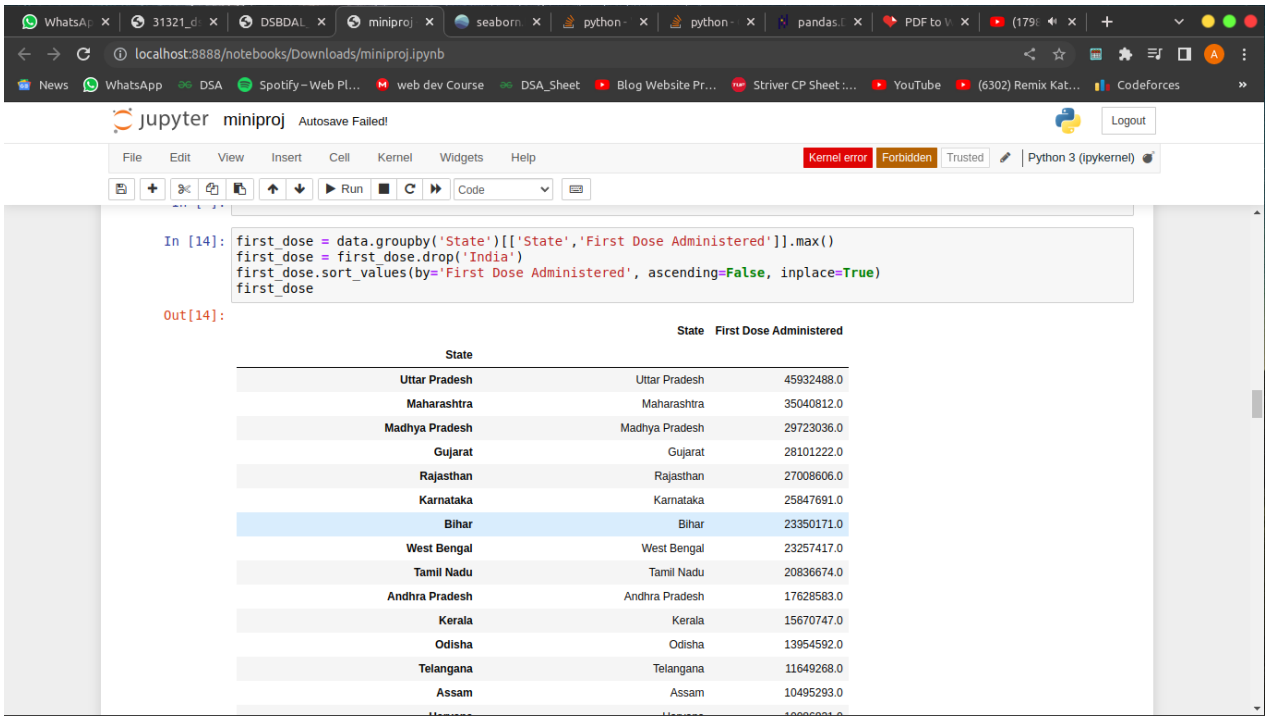
	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)	45-60 Y (D Administe
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0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

5 rows × 24 columns

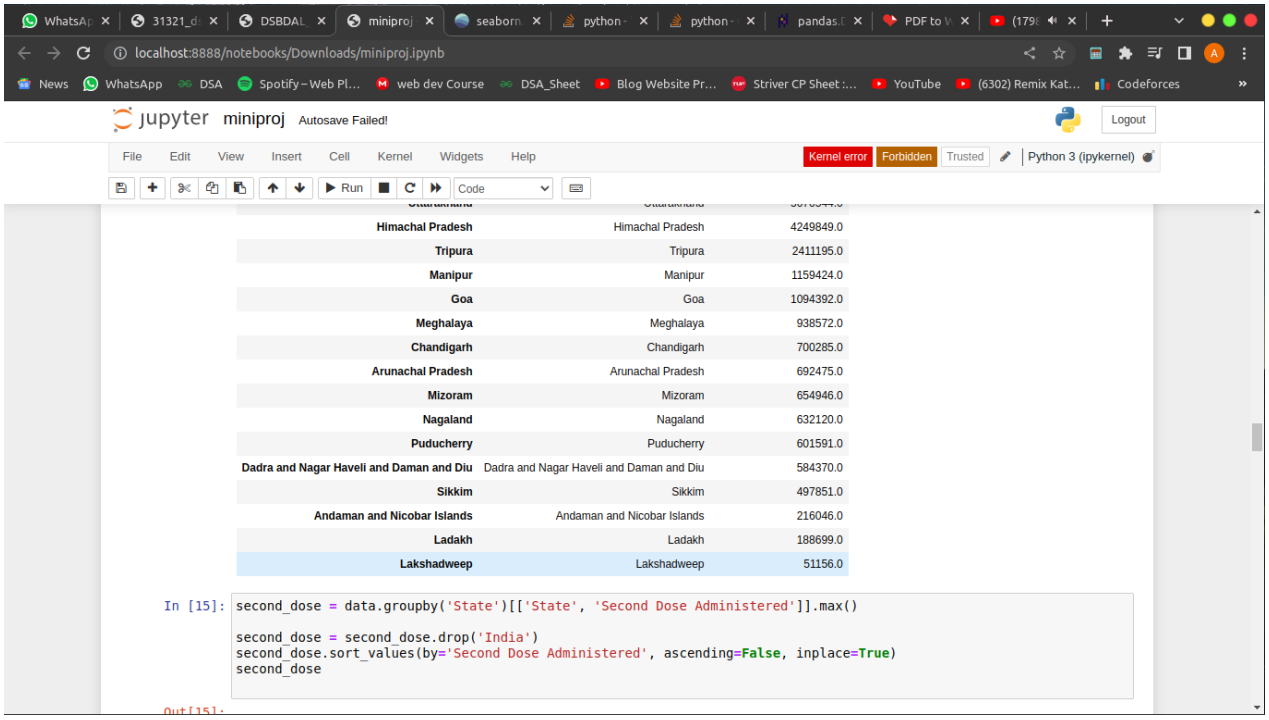

localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Trusted Python 3 (ipykernel)
Run Code
In [4]:
print("The last five rows are: ")
data.tail()
The last five rows are:
Out[4]:
Updated On State Total Doses Administered Sessions Sites First Dose Administered Second Dose Administered Male (Doses Administered) Female (Doses Administered) Transgender (Doses Administered) ... 18-44 Years (Doses Administered)
7840 11/08/2021 West Bengal NaN NaN NaN NaN NaN NaN NaN NaN ... NaN
7841 12/08/2021 West Bengal NaN NaN NaN NaN NaN NaN NaN NaN ... NaN
7842 13/08/2021 West Bengal NaN NaN NaN NaN NaN NaN NaN NaN ... NaN
7843 14/08/2021 West Bengal NaN NaN NaN NaN NaN NaN NaN NaN ... NaN
7844 15/08/2021 West Bengal NaN NaN NaN NaN NaN NaN NaN NaN ... NaN
5 rows × 24 columns
In [5]:
# Shape of the dataset in the format of (rows, columns)
print("The shape is: ")
data.shape


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Trusted Python 3 (ipykernel)
Run Code
In [5]:
# Shape of the dataset in the format of (rows, columns)
print("The shape is: ")
data.shape
The shape is:
Out[5]: (7845, 24)
In [6]:
# Names of columns
print("The columns present in the dataset are: ")
data.columns
The columns present in the dataset are:
Out[6]: 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)',





State	State	First Dose Administered
Uttar Pradesh	Uttar Pradesh	45932488.0
Maharashtra	Maharashtra	35040812.0
Madhya Pradesh	Madhya Pradesh	29723036.0
Gujarat	Gujarat	28101222.0
Rajasthan	Rajasthan	27008606.0
Karnataka	Karnataka	25847691.0
Bihar	Bihar	23350171.0
West Bengal	West Bengal	23257417.0
Tamil Nadu	Tamil Nadu	20836674.0
Andhra Pradesh	Andhra Pradesh	17628583.0
Kerala	Kerala	15670747.0
Odisha	Odisha	13954592.0
Telangana	Telangana	11649268.0
Assam	Assam	10495293.0



jupyter miniproj Autosave Failed!

Logout

File Edit View Insert Cell Kernel Widgets Help

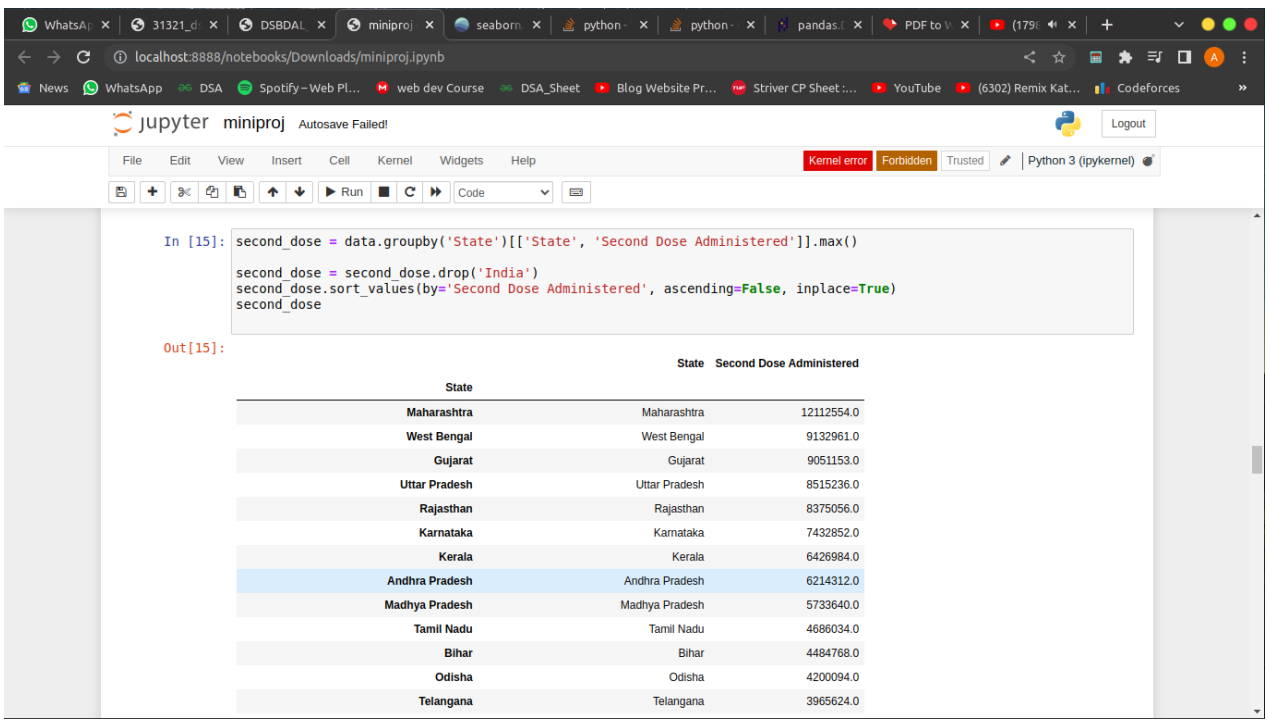
Kernel error Forbidden Trusted Python 3 (ipykernel)

Run Code

Himachal Pradesh	Himachal Pradesh	4249849.0
Tripura	Tripura	2411195.0
Manipur	Manipur	1159424.0
Goa	Goa	1094392.0
Meghalaya	Meghalaya	938572.0
Chandigarh	Chandigarh	700285.0
Arunachal Pradesh	Arunachal Pradesh	692475.0
Mizoram	Mizoram	654946.0
Nagaland	Nagaland	632120.0
Puducherry	Puducherry	601591.0
Dadra and Nagar Haveli and Daman and Diu	Dadra and Nagar Haveli and Daman and Diu	584370.0
Sikkim	Sikkim	497851.0
Andaman and Nicobar Islands	Andaman and Nicobar Islands	216046.0
Ladakh	Ladakh	188699.0
Lakshadweep	Lakshadweep	51156.0

In [15]:
```
second_dose = data.groupby('State')[['State', 'Second Dose Administered']].max()

second_dose = second_dose.drop('India')
second_dose.sort_values(by='Second Dose Administered', ascending=False, inplace=True)
second_dose
```

Out[15]:



jupyter miniproj Autosave Failed!

Logout

File Edit View Insert Cell Kernel Widgets Help

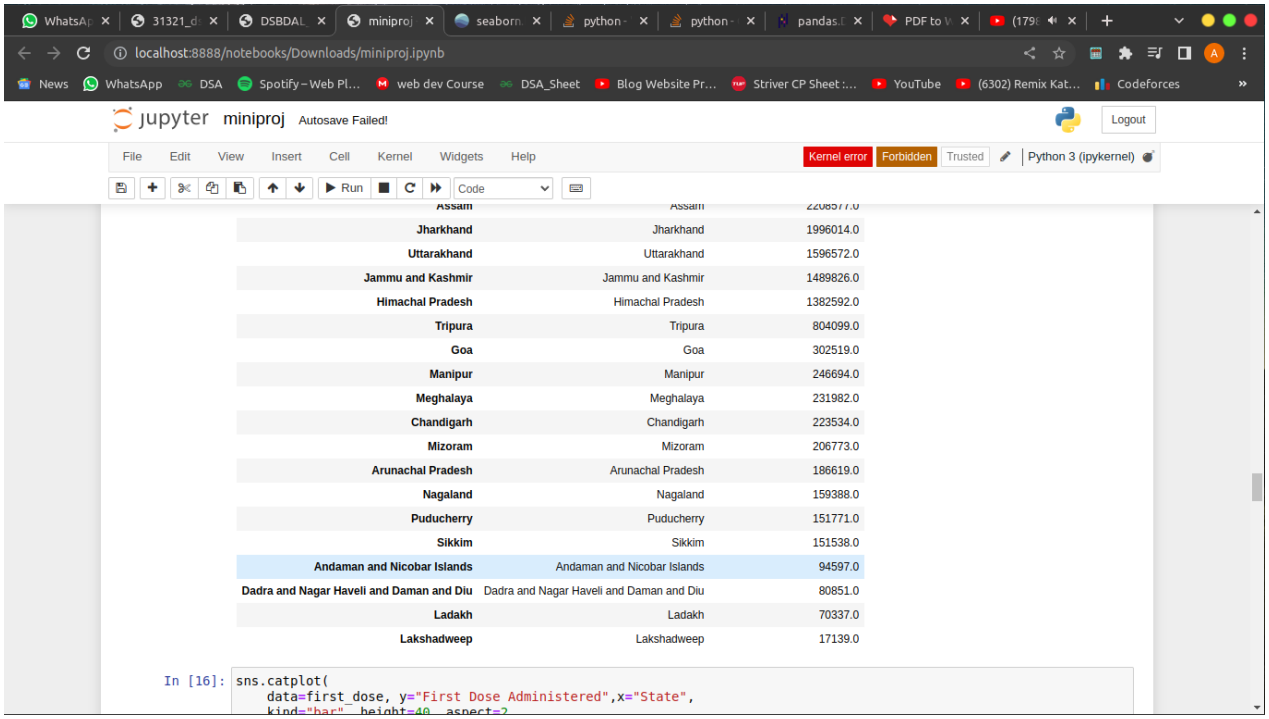
Kernel error Forbidden Trusted Python 3 (ipykernel)

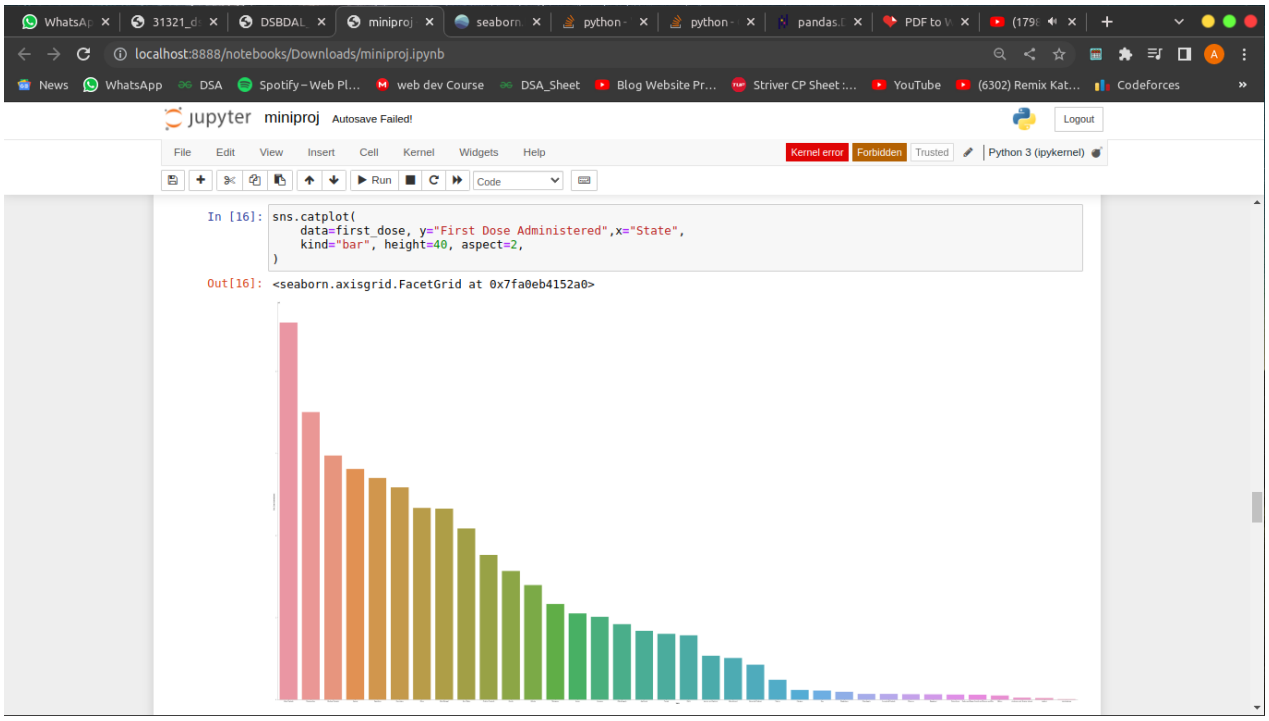
Run Code

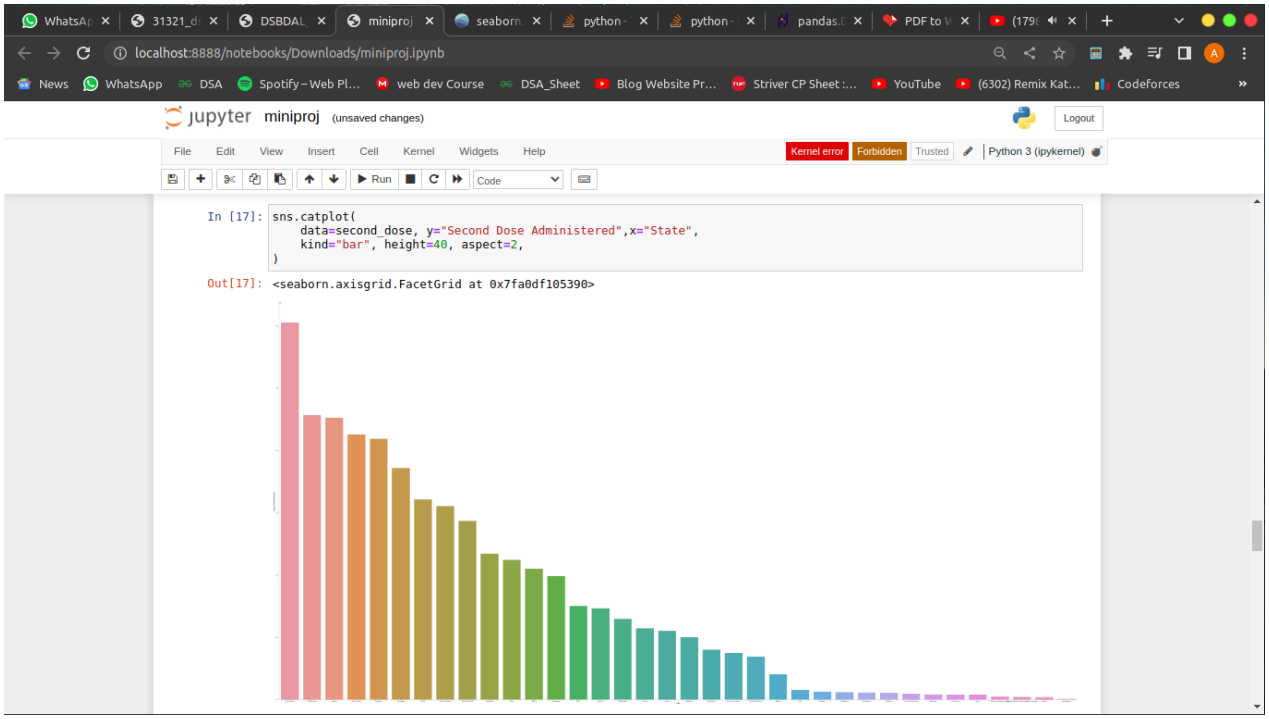
In [15]:
```
second_dose = data.groupby('State')[['State', 'Second Dose Administered']].max()

second_dose = second_dose.drop('India')
second_dose.sort_values(by='Second Dose Administered', ascending=False, inplace=True)
second_dose
```

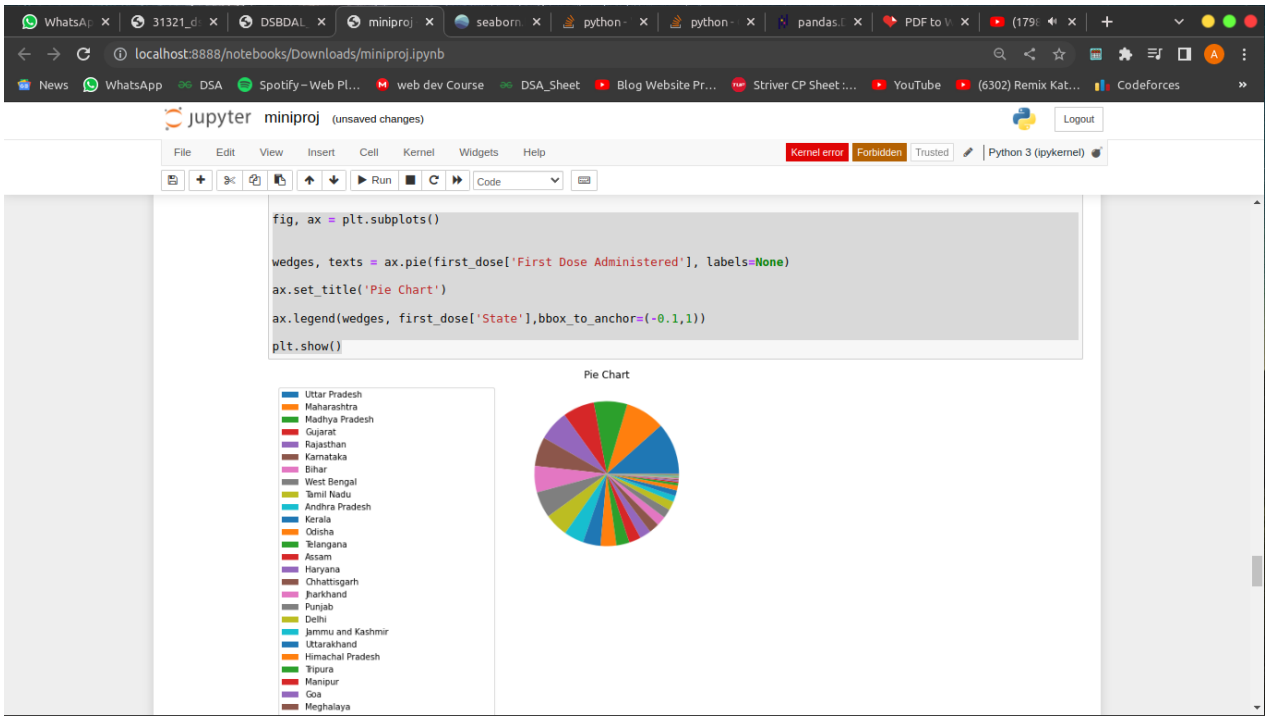
Out[15]:

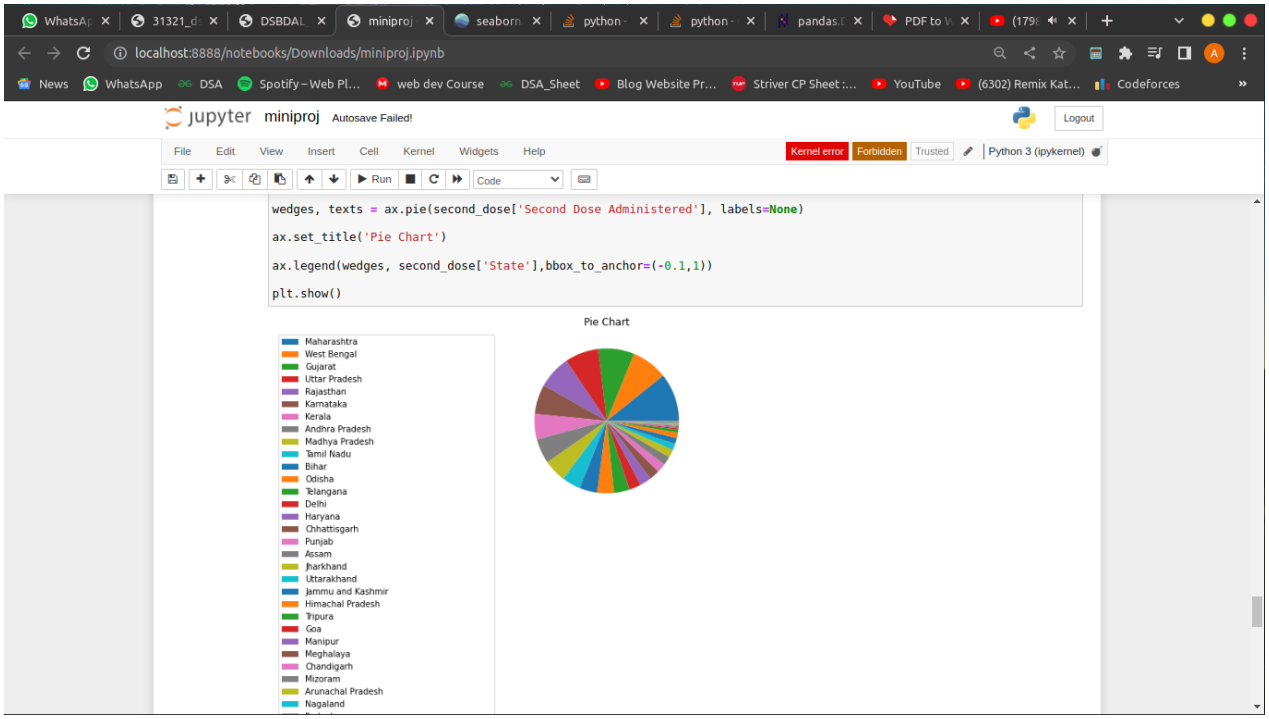
State	State	Second Dose Administered
Maharashtra	Maharashtra	12112554.0
West Bengal	West Bengal	9132961.0
Gujarat	Gujarat	9051153.0
Uttar Pradesh	Uttar Pradesh	8515236.0
Rajasthan	Rajasthan	8375056.0
Karnataka	Karnataka	7432852.0
Kerala	Kerala	6426984.0
Andhra Pradesh	Andhra Pradesh	6214312.0
Madhya Pradesh	Madhya Pradesh	5733640.0
Tamil Nadu	Tamil Nadu	4686034.0
Bihar	Bihar	4484768.0
Odisha	Odisha	4200094.0
Telangana	Telangana	3965624.0

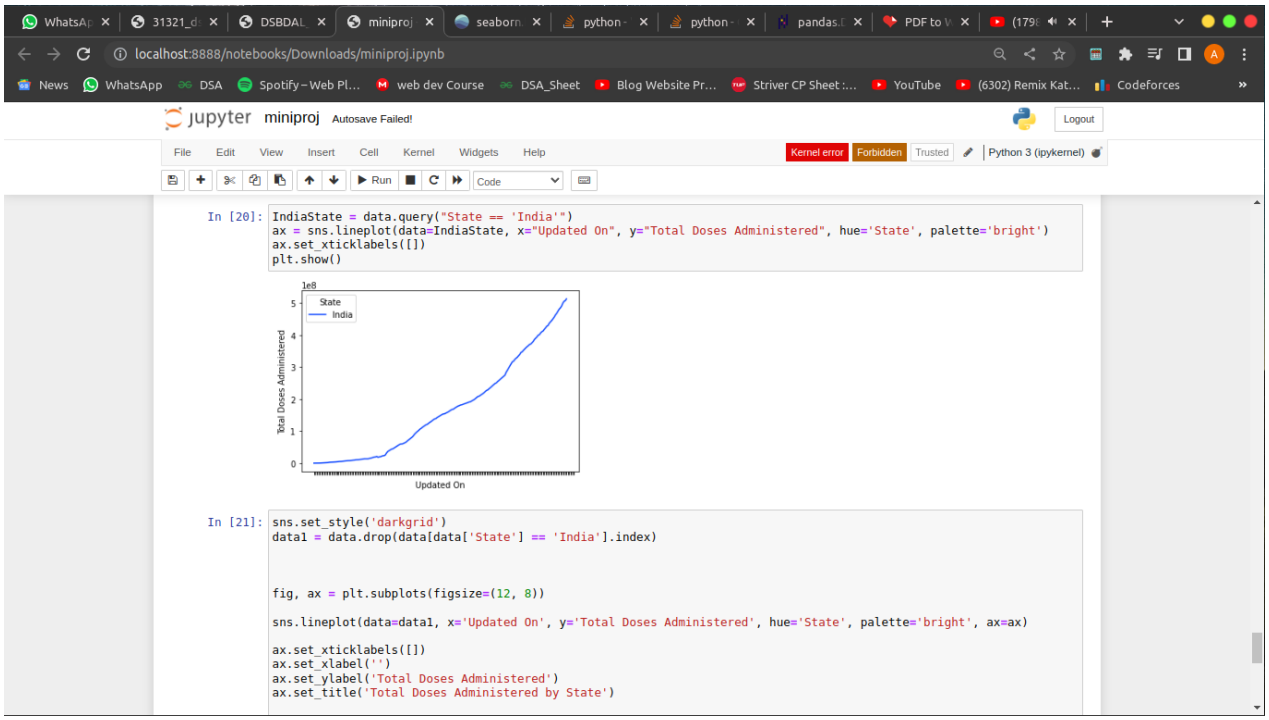

jupyter miniproj Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
Jharkhand Jharkhand 1996014.0
Uttarakhand Uttarakhand 1596572.0
Jammu and Kashmir Jammu and Kashmir 1489826.0
Himachal Pradesh Himachal Pradesh 1382592.0
Tripura Tripura 804099.0
Goa Goa 302519.0
Manipur Manipur 246694.0
Meghalaya Meghalaya 231982.0
Chandigarh Chandigarh 223534.0
Mizoram Mizoram 206773.0
Arunachal Pradesh Arunachal Pradesh 186619.0
Nagaland Nagaland 159388.0
Puducherry Puducherry 151771.0
Sikkim Sikkim 151538.0
Andaman and Nicobar Islands Andaman and Nicobar Islands 94597.0
Dadra and Nagar Haveli and Daman and Diu Dadra and Nagar Haveli and Daman and Diu 80851.0
Ladakh Ladakh 70337.0
Lakshadweep Lakshadweep 17139.0
In [16]: sns.catplot(
data=first_dose, y="First Dose Administered",x="State",

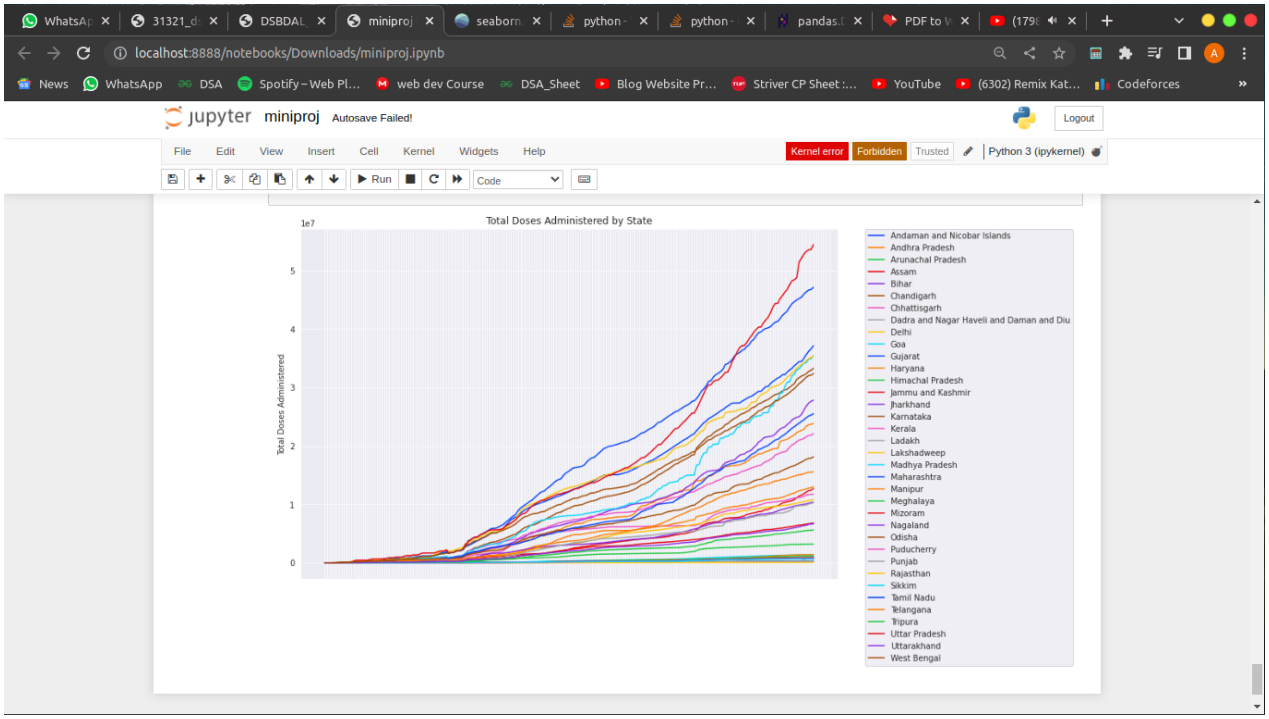

jupyter miniproj Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
In [16]: sns.catplot(
data=first_dose, y="First Dose Administered",x="State",
kind="bar", height=40, aspect=2,
)
Out[16]: <seaborn.axisgrid.FacetGrid at 0x7fa0eb4152a0>


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
In [17]:
sns.catplot(
data=second_dose, y="Second Dose Administered",x="State",
kind="bar", height=40, aspect=2,
)
Out[17]: <seaborn.axisgrid.FacetGrid at 0x7fa0df105390>


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
fig, ax = plt.subplots()
wedges, texts = ax.pie(first_dose['First Dose Administered'], labels=None)
ax.set_title('Pie Chart')
ax.legend(wedges, first_dose['State'],bbox_to_anchor=(-0.1,1))
plt.show()
Pie Chart
Uttar Pradesh
Maharashtra
Madhya Pradesh
Gujarat
Rajasthan
Karnataka
Bihar
West Bengal
Tamil Nadu
Andhra Pradesh
Kerala
Odisha
Telangana
Assam
Haryana
Chhattisgarh
Jharkhand
Punjab
Delhi
Jammu and Kashmir
Uttarakhand
Himachal Pradesh
Tripura
Manipur
Goa
Meghalaya


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
wedges, texts = ax.pie(second_dose['Second Dose Administered'], labels=None)
ax.set_title('Pie Chart')
ax.legend(wedges, second_dose['State'],bbox_to_anchor=(-0.1,1))
plt.show()
Pie Chart
Maharashtra
West Bengal
Gujarat
Uttar Pradesh
Rajasthan
Karnataka
Kerala
Andhra Pradesh
Madhya Pradesh
Tamil Nadu
Bihar
Odisha
Telangana
Delhi
Haryana
Chhattisgarh
Punjab
Assam
Jharkhand
Uttarakhand
Jammu and Kashmir
Himachal Pradesh
Tripura
Goa
Manipur
Meghalaya
Chandigarh
Mizoram
Arunachal Pradesh
Nagaland


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
In [20]:
IndiaState = data.query("State == 'India'")
ax = sns.lineplot(data=IndiaState, x="Updated On", y="Total Doses Administered", hue='State', palette='bright')
ax.set_xticklabels([])
plt.show()
1e8
State
India
Total Doses Administered
Updated On
In [21]:
sns.set_style('darkgrid')
data1 = data.drop(data[data['State'] == 'India'].index)
fig, ax = plt.subplots(figsize=(12, 8))
sns.lineplot(data=data1, x='Updated On', y='Total Doses Administered', hue='State', palette='bright', ax=ax)
ax.set_xticklabels([])
ax.set_xlabel('')
ax.set_ylabel('Total Doses Administered')
ax.set_title('Total Doses Administered by State')


localhost:8888/notebooks/Downloads/miniproj.ipynb
jupyter miniproj Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel error Forbidden Trusted Python 3 (ipykernel)
Run Code
Total Doses Administered by State
1e7
Total Doses Administered
0 1 2 3 4 5
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Dadra and Nagar Haveli and Daman and Diu
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Lakshadweep
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

Future Scope:

In the future, we could add extra features such as analyzing the number of new cases and deaths that were incurred during this period and finding the relationship between the cases, deaths, and vaccinations to find how effective the vaccines were at preventing deadly or highly injurious cases as well as casualties

Conclusion:

Thus, we successfully analyzed the available dataset about Covid Cases and vaccinations in India and were able to find how categorical values like State, Age, and Gender are related to the number of people vaccinated by Date.