


```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("covid_19_cleaned dataset.csv")
df
```



	Sno	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Covaxin (Doses Administered)	CoviShield (Doses Administered)	Sputnik V (Doses Administered)	Male(Indivi Vaccir
0	1.0	1/30/2020	6:00 PM	Kerala	0.0	0.0	1.0	579.0	47697.0	NaN	2
1	2.0	1/31/2020	6:00 PM	Kerala	0.0	0.0	1.0	635.0	57969.0	NaN	2
2	3.0	2/1/2020	6:00 PM	Kerala	0.0	0.0	2.0	1299.0	98150.0	NaN	4
3	4.0	2/2/2020	6:00 PM	Kerala	0.0	0.0	3.0	3017.0	192508.0	NaN	8
4	5.0	2/3/2020	6:00 PM	Kerala	0.0	0.0	3.0	3946.0	247334.0	NaN	9
...	...	...	...	...	...	...	...	...	...	...	...
18105	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18106	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18107	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18108	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18109	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

18110 rows × 14 columns

```
start_row_number=7841
df.drop(df.index[start_row_number:], inplace=True)
```

df

	Sno	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Covaxin (Doses Administered)	CoviShield (Doses Administered)	Sputnik V (Doses Administered)	Male(1 V
0	1.0	1/30/2020	6:00 PM	Kerala	0.0	0.0	1.0	579.0	47697.0	NaN	
1	2.0	1/31/2020	6:00 PM	Kerala	0.0	0.0	1.0	635.0	57969.0	NaN	
2	3.0	2/1/2020	6:00 PM	Kerala	0.0	0.0	2.0	1299.0	98150.0	NaN	
3	4.0	2/2/2020	6:00 PM	Kerala	0.0	0.0	3.0	3017.0	192508.0	NaN	
4	5.0	2/3/2020	6:00 PM	Kerala	0.0	0.0	3.0	3946.0	247334.0	NaN	
...	...	...	...	...	...	...	...	...	...	...	...
7836	7837.0	10/29/2020	8:00 AM	Karnataka	733558.0	11046.0	812784.0	4077069.0	27910835.0	41421.0	
7837	7838.0	10/29/2020	8:00 AM	Kerala	316692.0	1403.0	411464.0	4091163.0	27965147.0	42458.0	
7838	7839.0	10/29/2020	8:00 AM	Ladakh	5369.0	74.0	6085.0	4168230.0	28178632.0	43516.0	
7839	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7840	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

7841 rows × 14 columns

```
df.shape
```

(7841, 14)

```
df.count()

Sno                7839
Date              7839
Time              7839
State/UnionTerritory 7839
Cured             7839
Deaths           7839
Confirmed         7839
  Covaxin (Doses Administered) 7621
  CoviShield (Doses Administered) 7621
  Sputnik V (Doses Administered) 2995
  Male(Individuals Vaccinated) 160
  Female(Individuals Vaccinated) 160
  Transgender(Individuals Vaccinated) 160
  Total Individuals Vaccinated 5919
dtype: int64


df.isnull()

      Sno  Date  Time  State/UnionTerritory  Cured  Deaths  Confirmed  Covaxin (Doses Administered)  CoviShield (Doses Administered)  Sputnik V (Doses Administered)  Male(Individuals Vaccinated)
0  False  False  False                    False  False  False      False          False          False          True          False
1  False  False  False                    False  False  False      False          False          False          True          False
2  False  False  False                    False  False  False      False          False          False          True          False
3  False  False  False                    False  False  False      False          False          False          True          False
4  False  False  False                    False  False  False      False          False          False          True          False
...     ...    ...    ...                    ...    ...    ...          ...          ...          ...          ...
7836  False  False  False                    False  False  False      False          False          False          False          True
7837  False  False  False                    False  False  False      False          False          False          False          True
7838  False  False  False                    False  False  False      False          False          False          False          True
7839   True   True   True                     True   True   True       True          True          True          True          True
7840   True   True   True                     True   True   True       True          True          True          True          True

7841 rows x 14 columns
```

```
df.isnull().sum()

Sno                2
Date              2
Time              2
State/UnionTerritory 2
Cured             2
Deaths           2
Confirmed         2
  Covaxin (Doses Administered) 220
  CoviShield (Doses Administered) 220
  Sputnik V (Doses Administered) 4846
  Male(Individuals Vaccinated) 7681
  Female(Individuals Vaccinated) 7681
  Transgender(Individuals Vaccinated) 7681
  Total Individuals Vaccinated 1922
dtype: int64


df.notnull().sum()

Sno                7839
Date              7839
Time              7839
State/UnionTerritory 7839
Cured             7839
Deaths           7839
Confirmed         7839
  Covaxin (Doses Administered) 7621
  CoviShield (Doses Administered) 7621
  Sputnik V (Doses Administered) 2995
  Male(Individuals Vaccinated) 160
  Female(Individuals Vaccinated) 160
  Transgender(Individuals Vaccinated) 160
  Total Individuals Vaccinated 5919
dtype: int64
```

```
df.drop(['Sno'],axis=1)
```

	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Covaxin (Doses Administered)	CoviShield (Doses Administered)	Sputnik V (Doses Administered)	Male(Individu Vaccinat
0	1/30/2020	6:00 PM	Kerala	0.0	0.0	1.0	579.0	47697.0	NaN	2375
1	1/31/2020	6:00 PM	Kerala	0.0	0.0	1.0	635.0	57969.0	NaN	2734
2	2/1/2020	6:00 PM	Kerala	0.0	0.0	2.0	1299.0	98150.0	NaN	4136
3	2/2/2020	6:00 PM	Kerala	0.0	0.0	3.0	3017.0	192508.0	NaN	8190
4	2/3/2020	6:00 PM	Kerala	0.0	0.0	3.0	3946.0	247334.0	NaN	9811
...	...	...	...	...	...	...	...	...	...	...
7836	10/29/2020	8:00 AM	Karnataka	733558.0	11046.0	812784.0	4077069.0	27910835.0	41421.0	1
7837	10/29/2020	8:00 AM	Kerala	316692.0	1403.0	411464.0	4091163.0	27965147.0	42458.0	1
7838	10/29/2020	8:00 AM	Ladakh	5369.0	74.0	6085.0	4168230.0	28178632.0	43516.0	1
7839	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
7840	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1

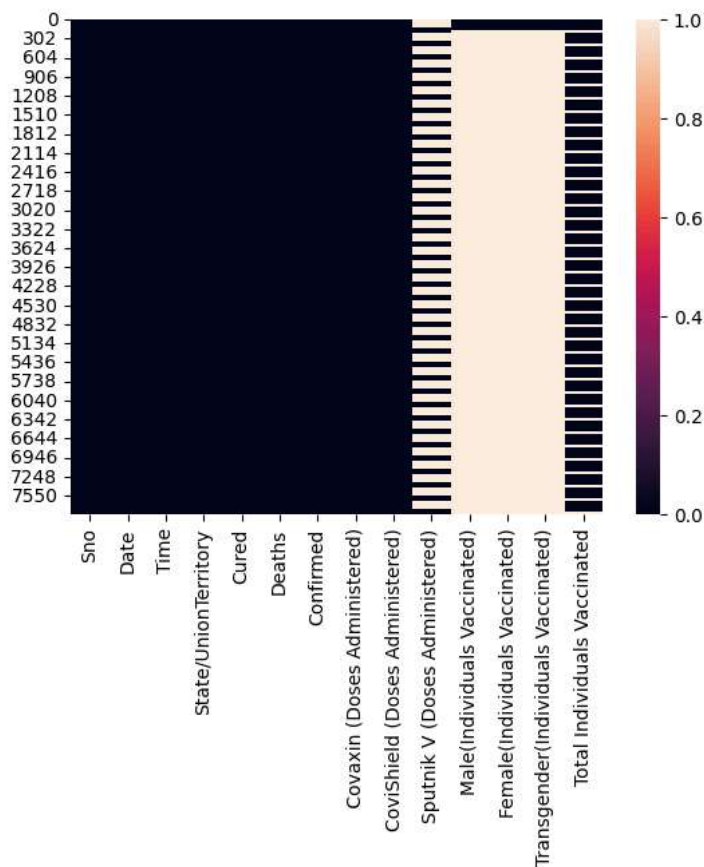
7841 rows × 13 columns

```
df.drop(['Time'],axis=1)
```

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Covaxin (Doses Administered)	CoviShield (Doses Administered)	Sputnik V (Doses Administered)	Male(Individu Vaccinat
0	1.0	1/30/2020	Kerala	0.0	0.0	1.0	579.0	47697.0	NaN	23
1	2.0	1/31/2020	Kerala	0.0	0.0	1.0	635.0	57969.0	NaN	27
2	3.0	2/1/2020	Kerala	0.0	0.0	2.0	1299.0	98150.0	NaN	41
3	4.0	2/2/2020	Kerala	0.0	0.0	3.0	3017.0	192508.0	NaN	81
4	5.0	2/3/2020	Kerala	0.0	0.0	3.0	3946.0	247334.0	NaN	98
...	...	...	...	...	...	...	...	...	...	...
7836	7837.0	10/29/2020	Karnataka	733558.0	11046.0	812784.0	4077069.0	27910835.0	41421.0	
7837	7838.0	10/29/2020	Kerala	316692.0	1403.0	411464.0	4091163.0	27965147.0	42458.0	
7838	7839.0	10/29/2020	Ladakh	5369.0	74.0	6085.0	4168230.0	28178632.0	43516.0	
7839	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7840	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

7841 rows × 13 columns

```
sns.heatmap(df.isnull())
plt.show()
```

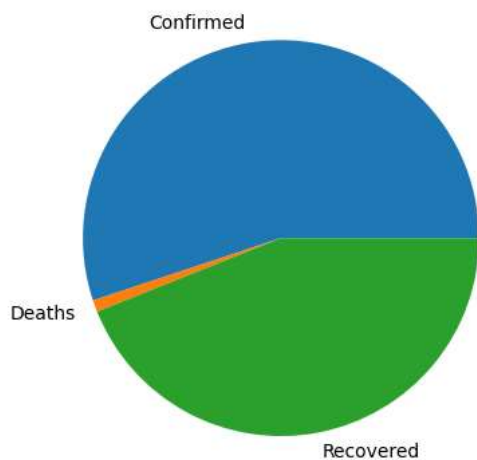


```
df["Date"] = pd.to_datetime(df["Date"])
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Create a DataFrame
data = {"State": [ "Confirmed", "Deaths", "Cured"],
        "Values": [ 481664691, 8346196, 384032915],
        "labels":["Confirmed", "Deaths", "Recovered"]}
df = pd.DataFrame(data)
```

```
# Plot the pie chart using only the "Values" column
plt.pie(df["Values"],labels=df["labels"])
plt.show()
```



```
df=pd.read_csv("covid_19_cleaned dataset.csv")
df["Confirmed"].sum()
```

```
481664691.0
```

```
df['Deaths'].sum()
```

8346196.0

```
df['Cured'].sum()
```

384032915.0

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Read the CSV file
df = pd.read_csv("covid_19_cleaned dataset.csv")
```

```
# Sort the DataFrame by date
df = df.sort_values(by="State/UnionTerritory")
```

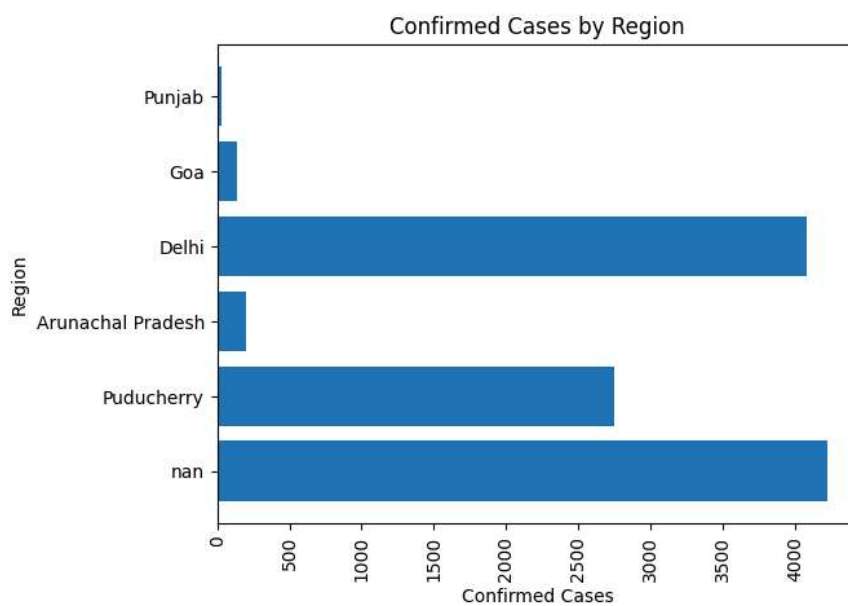
```
# Extract the confirmed cases and date columns
confirmed_cases = df["Confirmed"].head(20)
region = df["State/UnionTerritory"].sample(20)
region = region.astype(str)
```

```
# Create the bar chart
plt.barh( region,confirmed_cases)
```

```
# Add labels and title
plt.xlabel("Confirmed Cases")
plt.ylabel("Region")
plt.title("Confirmed Cases by Region")
```

```
# Rotate the x-axis labels for readability
plt.xticks(rotation=90)
```

```
# Display the chart
plt.show()
```



```

import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file
df = pd.read_csv("covid_19_cleaned dataset.csv")

# Sort the DataFrame by date
df = df.sort_values(by="State/UnionTerritory")

# Extract the confirmed cases and date columns
cured_cases = df["Cured"].sample(1000)
region = df["State/UnionTerritory"].sample(1000)
region = region.astype(str)

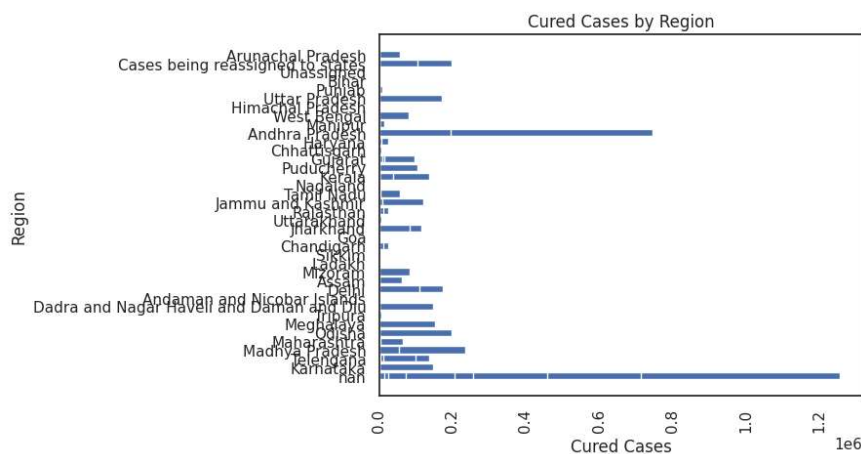
# Create the bar chart
plt.barh( region,cured_cases)

# Add labels and title
plt.xlabel("Cured Cases")
plt.ylabel("Region")
plt.title("Cured Cases by Region")

# Rotate the x-axis labels for readability
plt.xticks(rotation=90)

# Display the chart
plt.show()

```



Start coding or [generate](#) with AI.

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_dataset.csv' with your actual dataset)
data = pd.read_csv('covid_19_cleaned dataset.csv')

# Select the columns you want to use for the heatmap
selected_columns = ['Covaxin (Doses Administered)', 'CoviShield (Doses Administered)', 'Sputnik V (Doses Administered)']

# Create a subset of the data with the selected columns
data_subset = data[selected_columns]

# Calculate the correlation matrix
correlation_matrix = data_subset.corr()

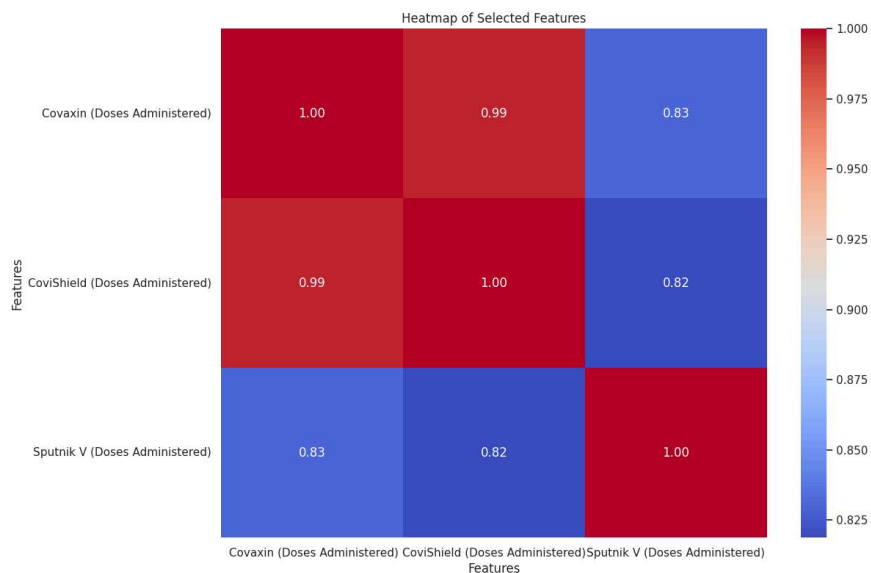
# Configure the heatmap
sns.set_theme(style="white")

# Generate the heatmap
plt.figure(figsize=(12, 9))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")

# Add a title and labels
plt.title("Heatmap of Selected Features")
plt.xlabel("Features")
plt.ylabel("Features")

# Show the heatmap
plt.show()

```



```
import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file
df = pd.read_csv("covid_19_cleaned dataset.csv")

# Sort the DataFrame by date
df = df.sort_values(by="State/UnionTerritory")

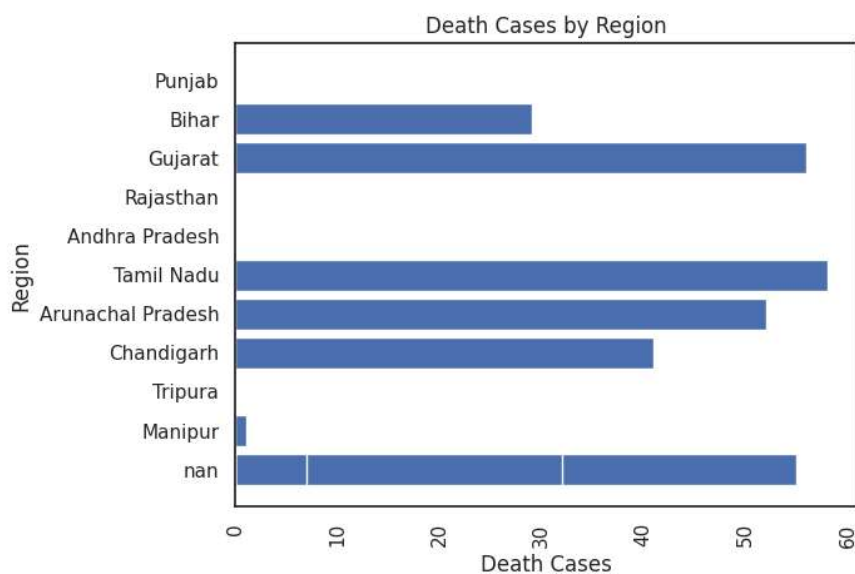
# Extract the confirmed cases and date columns
death_cases = df["Deaths"].head(20)
region = df["State/UnionTerritory"].sample(20)
region = region.astype(str)

# Create the bar chart
plt.barh( region,death_cases)

# Add labels and title
plt.xlabel("Death Cases")
plt.ylabel("Region")
plt.title("Death Cases by Region")

# Rotate the x-axis labels for readability
plt.xticks(rotation=90)

# Display the chart
plt.show()
```



Start coding or [generate](#) with AI.

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('covid_19_cleaned dataset.csv')
```

```
# Select the relevant columns
total_individuals_column = 'Total Individuals Vaccinated'
covaxin_column = 'Covaxin (Doses Administered)'
covishield_column = 'CoviShield (Doses Administered)'
sputnikv_column = 'Sputnik V (Doses Administered)'
```

```
# Create the scatter plot
plt.figure(figsize=(12, 6))
```

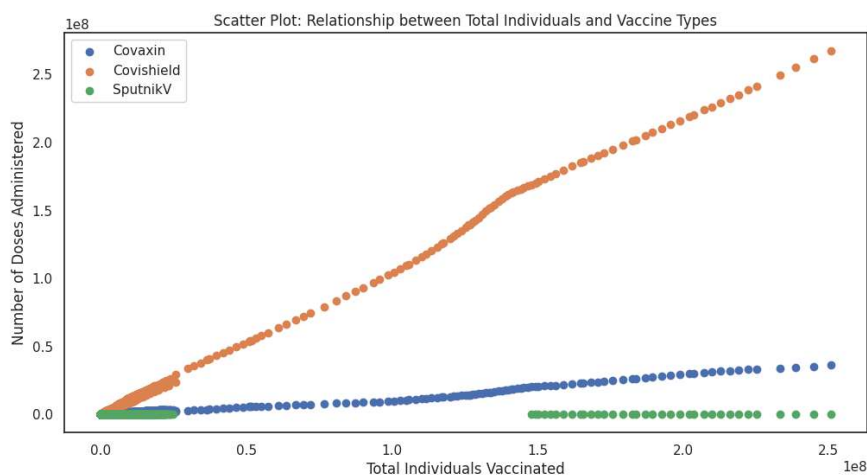
```
# Plot Covaxin vs Total Individuals
plt.scatter(data[total_individuals_column], data[covaxin_column], label='Covaxin')
```

```
# Plot Covishield vs Total Individuals
plt.scatter(data[total_individuals_column], data[covishield_column], label='Covishield')
```

```
# Plot SputnikV vs Total Individuals
plt.scatter(data[total_individuals_column], data[sputnikv_column], label='SputnikV')
```

```
# Add labels and title
plt.xlabel('Total Individuals Vaccinated')
plt.ylabel('Number of Doses Administered')
plt.title('Scatter Plot: Relationship between Total Individuals and Vaccine Types')
```

```
# Add legend and show the plot
plt.legend()
plt.show()
```





```

import pandas as pd
import matplotlib.pyplot as plt

# Load your dataset (replace 'your_dataset.csv' with your actual dataset)
data = pd.read_csv('covid_19_cleaned dataset.csv')

# Select the relevant columns
state_column = 'State/UnionTerritory'
cured_column = 'Cured'
confirmed_column = 'Confirmed'
death_column = 'Deaths'

# Group data by state
grouped_data = data.groupby(state_column)[[cured_column, confirmed_column, death_column]].sum()

# Normalize the data
normalized_data = grouped_data.apply(lambda x: (x - x.min()) / (x.max() - x.min()))

# Perform hierarchical clustering
from scipy.cluster.hierarchy import linkage, dendrogram

linkage_matrix = linkage(normalized_data.values, method='ward')

# Create the cluster plot
plt.figure(figsize=(16, 10))
dendrogram(linkage_matrix, labels=normalized_data.index, orientation='top')
plt.title('Cluster Plot: Relationship between States and COVID-19 Cases')
plt.xlabel('States')
plt.ylabel('Covid-19 Cases')
plt.show()

! pip install streamlit -q

```

```

===== 8.1/8.1 MB 29.3 MB/s eta 0:00:00
===== 207.3/207.3 kB 16.7 MB/s eta 0:00:00
===== 1.9/1.9 MB 13.0 MB/s eta 0:00:00

```