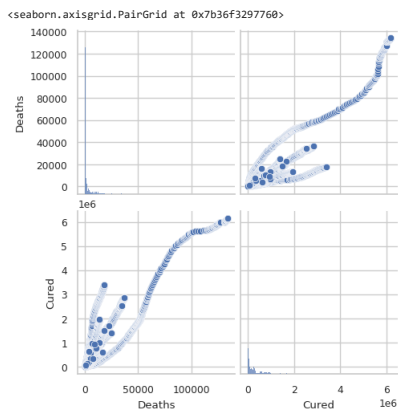
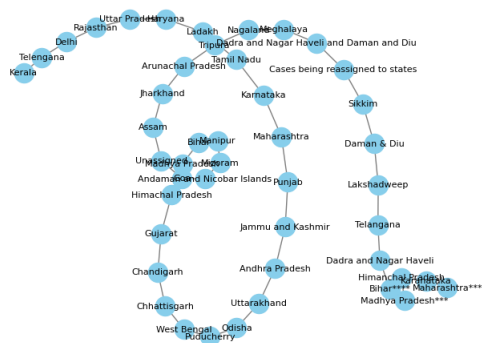


```
sns.pairplot(data)
```

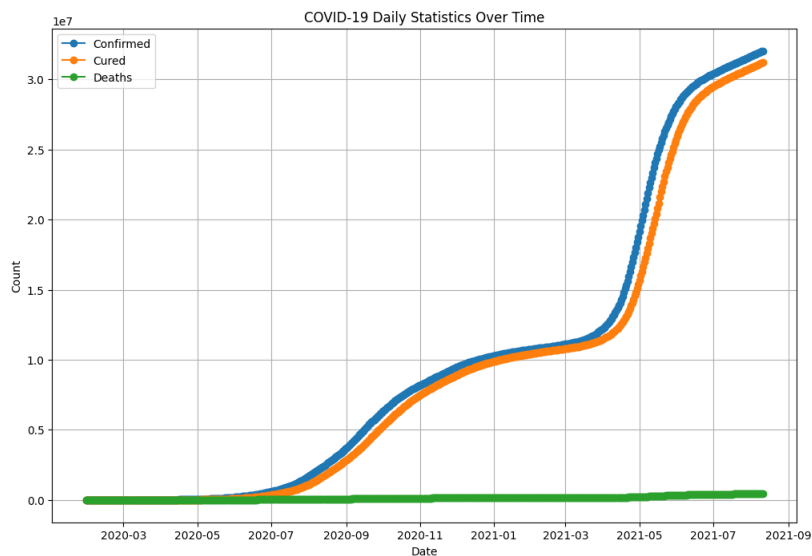


```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
covid_data = pd.read_csv('/content/covid_19_india (2).csv')
covid_data = covid_data[['State/UnionTerritory', 'Deaths', 'Cured', 'Latitude', 'Longitude']]
G = nx.Graph()
for state in covid_data['State/UnionTerritory']:
    G.add_node(state)
states = covid_data['State/UnionTerritory'].unique()
for i in range(len(states)-1):
    G.add_edge(states[i], states[i+1])
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, font_size=8, node_size=300, font_color='black', node_color='skyblue', edge_color='gray')
plt.title('Network of States/Union Territories')
plt.show()
```

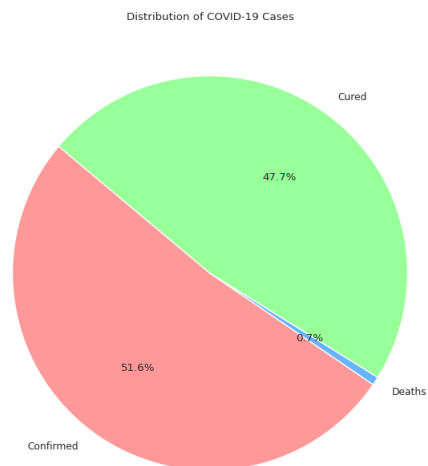
Network of States/Union Territories



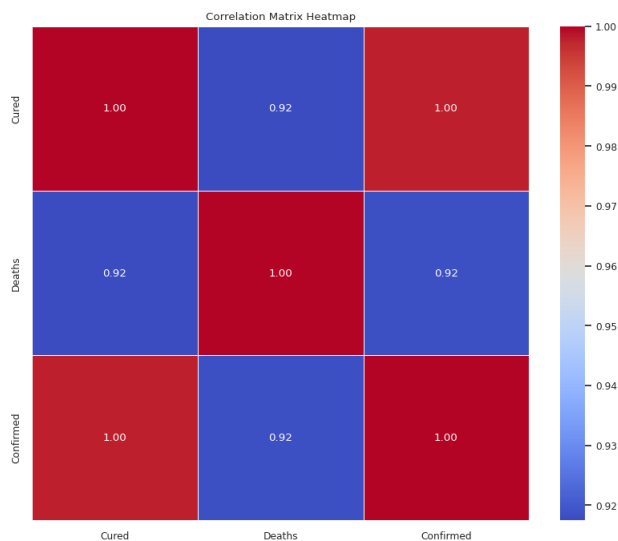
```
import pandas as pd
import matplotlib.pyplot as plt
covid_data = pd.read_csv('/content/covid_19_india (2).csv')
covid_data['Date'] = pd.to_datetime(covid_data['Date'], format='%d-%m-%Y') # Adjust the format based on your data
daily_stats = covid_data.groupby('Date').agg({
    'Confirmed': 'sum',
    'Cured': 'sum',
    'Deaths': 'sum'
}).reset_index()
plt.figure(figsize=(12, 8))
plt.plot(daily_stats['Date'], daily_stats['Confirmed'], label='Confirmed', marker='o')
plt.plot(daily_stats['Date'], daily_stats['Cured'], label='Cured', marker='o')
plt.plot(daily_stats['Date'], daily_stats['Deaths'], label='Deaths', marker='o')
plt.xlabel('Date')
plt.ylabel('Count')
plt.title('COVID-19 Daily Statistics Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



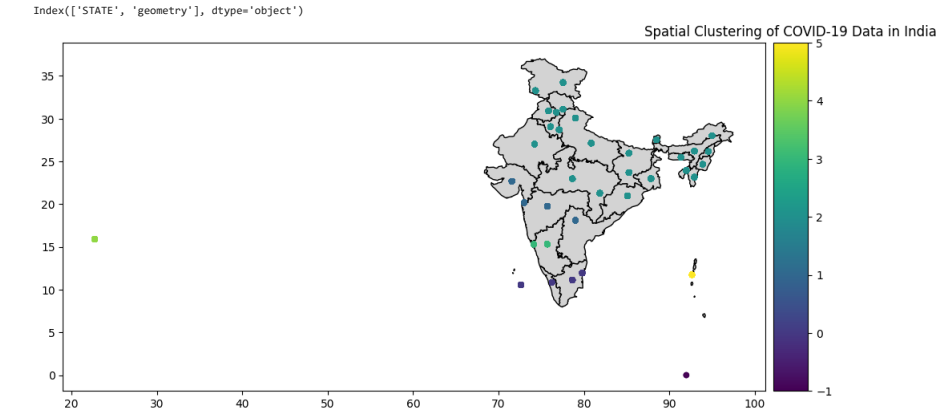
```
import pandas as pd
import matplotlib.pyplot as plt
covid_data = pd.read_csv('/content/covid_19_india (2).csv')
cases_data = covid_data[['Confirmed', 'Deaths', 'Cured']].sum()
plt.figure(figsize=(8, 8))
plt.pie(cases_data, labels=cases_data.index, autopct='%1.1f%%', startangle=140, colors=['#ff9999', '#66b3ff', '#99ff99'])
plt.title('Distribution of COVID-19 Cases')
plt.show()
```



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
covid_data = pd.read_csv('/content/covid_19_india (2).csv')
columns_of_interest = ['Cured', 'Deaths', 'Confirmed']
subset_data = covid_data[columns_of_interest]
correlation_matrix = subset_data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
covid_data = pd.read_csv('/content/covid_19_india (2) (1).csv')
covid_data = covid_data[['State/UnionTerritory', 'Deaths', 'Cured', 'Latitude', 'Longitude']]
covid_data['Deaths', 'Cured'] = covid_data[['Deaths', 'Cured']].apply(pd.to_numeric, errors='coerce')
covid_data.fillna(0, inplace=True)
geometry = gpd.points_from_xy(covid_data['Longitude'], covid_data['Latitude'])
gdf = gpd.GeoDataFrame(covid_data, geometry=geometry)
india_map = gpd.read_file('/content/india_st.shx')
gdf.crs = india_map.crs
print(india_map.columns)
merged_data = pd.merge(covid_data, india_map, left_on='State/UnionTerritory', right_on='STATE', how='left')
X = gdf[['Latitude', 'Longitude']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
dbscan = DBSCAN(eps=0.5, min_samples=5)
cluster_labels = dbscan.fit_predict(X_scaled)
gdf['Cluster'] = cluster_labels
fig, ax = plt.subplots(figsize=(12, 8))
divider = make_axes_locatable(ax)
cax = divider.append_axes("right", size="5%", pad=0.1)
india_map.plot(ax=ax, color='lightgrey', edgecolor='black')
gdf.plot(ax=ax, marker='o', column='Cluster', cmap='viridis', markersize=20, alpha=0.7, legend=True, cax=cax)
plt.title('Spatial Clustering of COVID-19 Data in India')
plt.show()
```



```
import pandas as pd
import geopandas as gpd
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

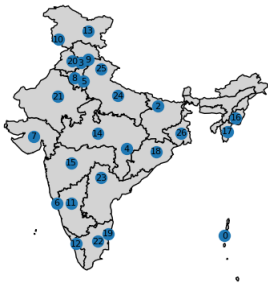
data = df[['Date', 'Cured', 'State/UnionTerritory', 'ConfirmedIndianNational', 'Longitude', 'Latitude']]
data['ConfirmedIndianNational'] = pd.to_numeric(data['ConfirmedIndianNational'], errors='coerce')
data = data.dropna(subset=['ConfirmedIndianNational'])
data['State/UnionTerritory'] = pd.Categorical(data['State/UnionTerritory'])
data['State/UnionTerritory'] = data['State/UnionTerritory'].cat.codes
gdf = gpd.GeoDataFrame(data, geometry=gpd.points_from_xy(data['Longitude'], data['Latitude']))
india_map = gpd.read_file('/content/india_st.shp')
G = nx.Graph()
G.add_nodes_from(gdf['State/UnionTerritory'])
for i, row in gdf.iterrows():
    neighbors = gdf[gdf.geometry.touches(row['geometry'])]['State/UnionTerritory'].tolist()
    G.add_edges_from((row['State/UnionTerritory'], neighbor) for neighbor in neighbors)
fig, ax = plt.subplots(figsize=(12, 12))
india_map.plot(ax=ax, color='lightgrey', edgecolor='black')
pos = {node: (gdf.loc[gdf['State/UnionTerritory'] == node, 'Longitude'].values[0],
               gdf.loc[gdf['State/UnionTerritory'] == node, 'Latitude'].values[0])
        for node in G.nodes}
nx.draw(G, pos=pos, with_labels=True, node_size=100, font_size=8, ax=ax, edge_color='blue')
plt.title('Spatial Network with India Map')
plt.show()
```

<ipython-input-26-6404c547498a>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
data['ConfirmedIndianNational'] = pd.to_numeric(data['ConfirmedIndianNational'], errors='coerce')
```

Spatial Network with India Map



Double-click (or enter) to edit