

DAV SAMPLE PROJECT

NAME: Anika

ROLL NO: CSC/21/38

BSc Hons COMPUTER SCIENCE

```
import pandas as pd
df = pd.read_csv("covid_19_india.csv")
print(df)
```

	Sno	Date	Time	State/UnionTerritory \
0	1	2020-01-30	6:00 PM	Kerala
1	2	2020-01-31	6:00 PM	Kerala
2	3	2020-02-01	6:00 PM	Kerala
3	4	2020-02-02	6:00 PM	Kerala
4	5	2020-02-03	6:00 PM	Kerala
...
18105	18106	2021-08-11	8:00 AM	Telangana
18106	18107	2021-08-11	8:00 AM	Tripura
18107	18108	2021-08-11	8:00 AM	Uttarakhand
18108	18109	2021-08-11	8:00 AM	Uttar Pradesh
18109	18110	2021-08-11	8:00 AM	West Bengal

	ConfirmedIndianNational	ConfirmedForeignNational	Cured Deaths \
0	1	0 0 0	
1	1	0 0 0	
2	2	0 0 0	
3	3	0 0 0	
4	3	0 0 0	
...
18105	-	- 638410 3831	
18106	-	- 77811 773	
18107	-	- 334650 7368	
18108	-	- 1685492 22775	
18109	-	- 1506532 18252	

	Confirmed
0	1
1	1
2	2
3	3
4	3
...	...
18105	650353
18106	80660
18107	342462
18108	1708812
18109	1534999

[18110 rows x 9 columns]

In [2]:

```

import matplotlib.pyplot as plt
# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')

# Task 1: For each state, find maximum cases reported for confirmed, deaths, and recovered
# for any three months of the year 2020.

# Filter data for the year 2020
df_2020 = df[(df['Date'] >= '2020-01-01') & (df['Date'] < '2021-01-01')]

# Group by state and find maximum cases for confirmed, deaths, and recovered
result = df_2020.groupby(['State/UnionTerritory', 'Date']).agg({
    'Confirmed': 'max',
    'Deaths': 'max',
    'Cured': 'max'
}).reset_index()

# Display the result
print("Task 1: Maximum cases reported for confirmed, deaths, and recovered individually for")
print(result)

```

Task 1: Maximum cases reported for confirmed, deaths, and recovered individually for any three months of 2020

	State/UnionTerritory	Date	Confirmed	Deaths	Cured
0	Andaman and Nicobar Islands	2020-03-26	1	0	0
1	Andaman and Nicobar Islands	2020-03-27	1	0	0
2	Andaman and Nicobar Islands	2020-03-28	6	0	0
3	Andaman and Nicobar Islands	2020-03-29	9	0	0
4	Andaman and Nicobar Islands	2020-03-30	9	0	0
...
10077	West Bengal	2020-12-27	546008	9569	522331
10078	West Bengal	2020-12-28	547443	9598	524071
10079	West Bengal	2020-12-29	548471	9625	525685
10080	West Bengal	2020-12-30	549715	9655	527272
10081	West Bengal	2020-12-31	550893	9683	528829

[10082 rows x 5 columns]

In [3]:

```

# Filter data for the specified states
selected_states = ['Karnataka', 'Gujarat', 'Haryana', 'Uttar Pradesh']
df_selected_states = df[df['State/UnionTerritory'].isin(selected_states)]

# Group by state and month, then sum the cured cases
df_cured_monthly = df_selected_states.groupby([df_selected_states['Date'].dt.to_period("M")
        'Cured': 'sum'
    ]).unstack()

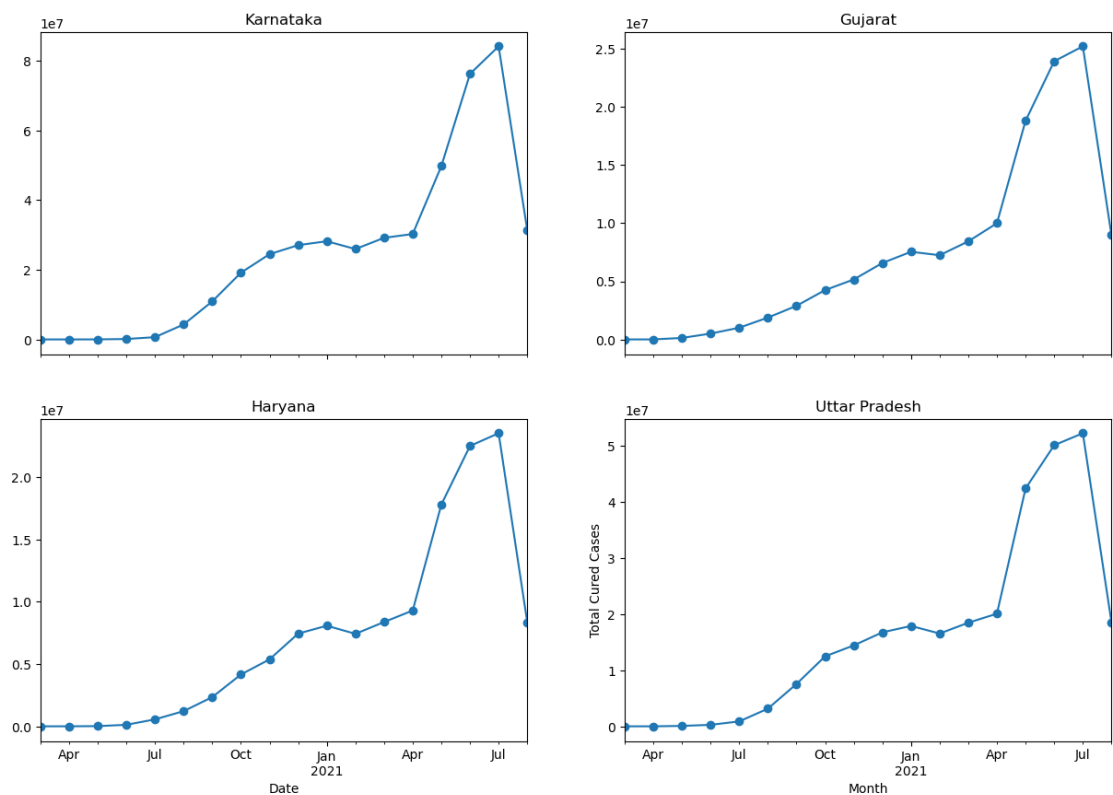
# Plot the subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10), sharex=True)

for i, state in enumerate(selected_states):
    df_cured_monthly['Cured'][state].plot(ax=axes[i // 2, i % 2], marker='o')
    axes[i // 2, i % 2].set_title(state)

plt.suptitle('Total Number of Cured Cases Month-wise (April 2020 to March 2021)')
plt.xlabel('Month')
plt.ylabel('Total Cured Cases')
plt.show()

```

Total Number of Cured Cases Month-wise (April 2020 to March 2021)



In [4]:

```

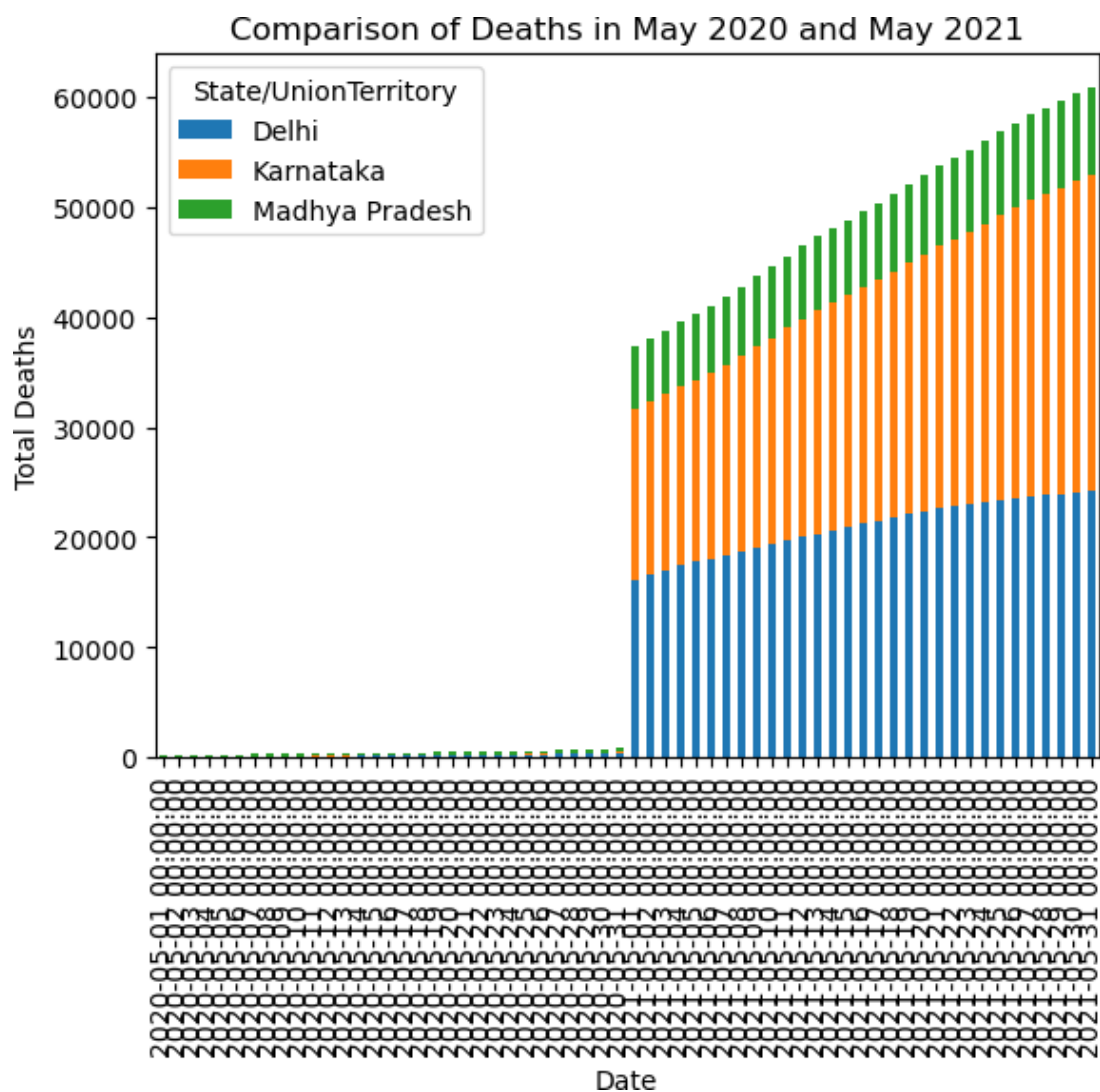
# Filter data for the specified states and months
selected_states_task3 = ['Karnataka', 'Delhi', 'Madhya Pradesh']
df_task3 = df[df['State/UnionTerritory'].isin(selected_states_task3) & (df['Date'].dt.month.isin(

# Group by state and sum the deaths
df_task3_grouped = df_task3.groupby(['State/UnionTerritory', 'Date']).agg({'Deaths': 'sum'}).r

# Pivot the table for plotting
df_task3_pivot = df_task3_grouped.pivot(index='Date', columns='State/UnionTerritory', value

# Plot the stacked bar chart
df_task3_pivot.plot(kind='bar', stacked=True)
plt.title('Comparison of Deaths in May 2020 and May 2021')
plt.xlabel('Date')
plt.ylabel('Total Deaths')
plt.show()

```



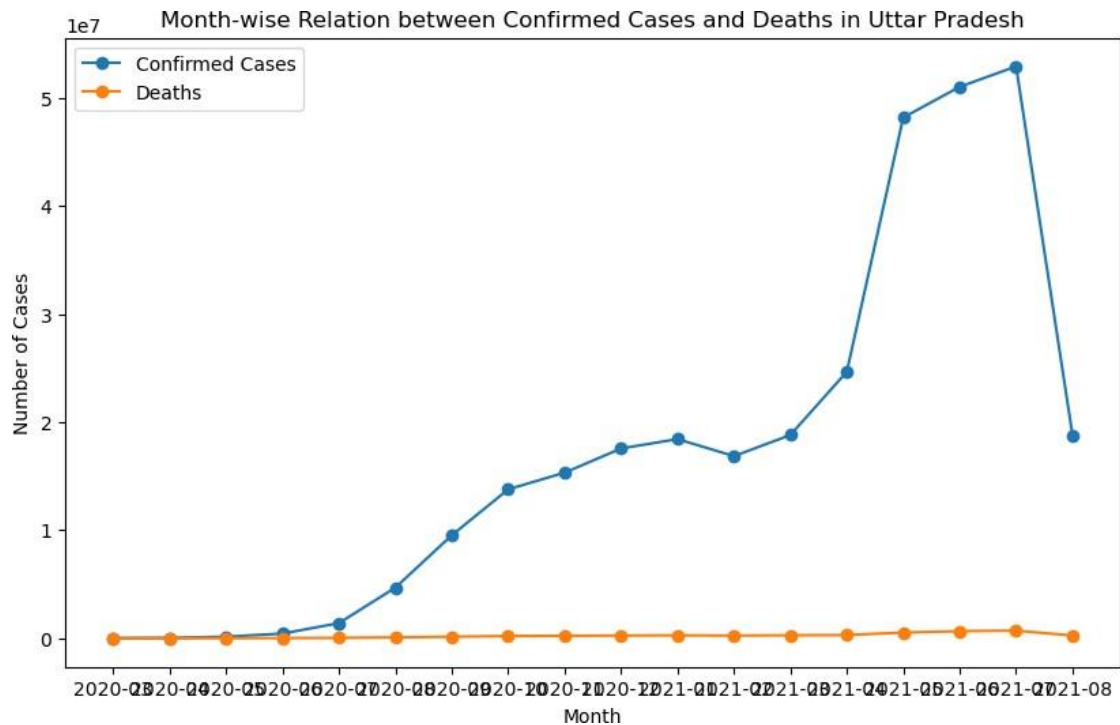
```

In [5]: # Filter data for Uttar Pradesh
df_up = df[df['State/UnionTerritory'] == 'Uttar Pradesh']

# Group by month and calculate the correlation
correlation_df = df_up.groupby(df_up['Date'].dt.to_period("M")).agg({
    'Confirmed': 'sum',
    'Deaths': 'sum'
})

# Plot the graph
plt.figure(figsize=(10, 6))
plt.plot(correlation_df.index.astype(str), correlation_df['Confirmed'], label='Confirmed Cases')
plt.plot(correlation_df.index.astype(str), correlation_df['Deaths'], label='Deaths', marker='o')
plt.title('Month-wise Relation between Confirmed Cases and Deaths in Uttar Pradesh')
plt.xlabel('Month')
plt.ylabel('Number of Cases')
plt.legend()
plt.show()

```



In []: