```
import pandas as pd
import numpy as np

df = pd.read_csv('/content/complete.csv')
df.head()
```

	Date	Name of State / UT	Latitude	Longitude	Total Confirmed cases	Death	Cured/Discharged/Migrated	New cases	New deaths	New recovered
0	2020-01-30	Kerala	10.8505	76.2711	1.0	0	0.0	0	0	0
1	2020-01-31	Kerala	10.8505	76.2711	1.0	0	0.0	0	0	0
2	2020-02-01	Kerala	10.8505	76.2711	2.0	0	0.0	1	0	0

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Problem Statements and their Solutions:-

# Q1. Display the first 10 rows of the dataset.
df.head(10)
```

	Date	Name of State / UT	Latitude	Longitude	Total Confirmed cases	Death	Cured/Discharged/Migrated	New cases	New deaths	New recovered
0	2020-01-30	Kerala	10.8505	76.2711	1.0	0.0	0.0	0	0	0
1	2020-01-31	Kerala	10.8505	76.2711	1.0	0.0	0.0	0	0	0
2	2020-02-01	Kerala	10.8505	76.2711	2.0	0.0	0.0	1	0	0
3	2020-02-02	Kerala	10.8505	76.2711	3.0	0.0	0.0	1	0	0
4	2020-02-03	Kerala	10.8505	76.2711	3.0	0.0	0.0	0	0	0
5	2020-02-04	Kerala	10.8505	76.2711	3.0	0.0	0.0	0	0	0

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Q2. Find the total number of rows and columns in the dataset.

df.shape
```

```
(4692, 10)
```

```
# Q3. List all the column names.

df.columns.tolist()
```

```
['Date',
 'Name of State / UT',
 'Latitude',
 'Longitude',
 'Total Confirmed cases',
 'Death',
 'Cured/Discharged/Migrated',
 'New cases',
 'New deaths',
 'New recovered']
```

```
# Q4. Check for any missing values in the dataset.

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

	0
Date	0
Name of State / UT	0
Latitude	0
Longitude	0
Total Confirmed cases	0
Death	0
Cured/Discharged/Migrated	0
New cases	0
New deaths	0
New recovered	0

dtype: int64

```
# Q5. Total number of confirmed cases across all states ?

total_confirmed = df['Total Confirmed cases'].sum()
print(f"1. Total Confirmed Cases: {total_confirmed}")
```

```
1. Total Confirmed Cases: 53460297.0
```

```
# Q6. Total number of deaths across all states ?

total_deaths = df['Death'].sum()
print(f"2. Total Deaths: {total_deaths}")
```

```
2. Total Deaths: 1366398.0
```

```
# Q7. Which state has reported the highest number of confirmed cases ?

state_max_confirmed = df.groupby('Name of State / UT')['Total Confirmed cases'].max().idxmax()
print(f"3. State with Maximum Confirmed Cases: {state_max_confirmed}")
```

```
3. State with Maximum Confirmed Cases: Maharashtra
```

```
# Q8. Which day saw the highest number of new cases ?

day_max_new_cases = df.groupby('Date')['New cases'].sum().idxmax()
print(f"5. Day with Maximum New Cases: {day_max_new_cases}")
```

```
5. Day with Maximum New Cases: 2020-07-18 00:00:00
```

```
# Q9. Number of states with more than 100,000 confirmed cases ?

states_over_100k = (df.groupby('Name of State / UT')['Total Confirmed cases'].max() > 100000).sum()
print(f"6. States with more than 100,000 cases: {states_over_100k}")
```

```
6. States with more than 100,000 cases: 6
```

```
# Q10. Total number of recoveries across all states ?

total_recoveries = df['Cured/Discharged/Migrated'].sum()
print(f"7. Total Recoveries: {total_recoveries}")
```

```
7. Total Recoveries: 32412949.0
```

```
# Q11. Overall death rate (deaths/confirmed cases) ?

overall_death_rate = total_deaths / total_confirmed
print(f"8. Overall Death Rate: {overall_death_rate:.2%}")
```

```
8. Overall Death Rate: 2.56%
```

```
# Q12. Top 5 states with the highest recovery rate ?
```

```
state_recovery_rate = (df.groupby('Name of State / UT')['Cured/Discharged/Migrated'].max() /
                       df.groupby('Name of State / UT')['Total Confirmed cases'].max()).sort_values(ascending=False).head(5)
print(f"9. Top 5 States by Recovery Rate:\n{state_recovery_rate}")
```

```
9. Top 5 States by Recovery Rate:
Name of State / UT
Delhi           0.899338
Haryana         0.829096
Tamil Nadu      0.785545
Telangana***    0.768765
Chhattisgarh    0.756318
dtype: float64
```

```
# Q13. Which state had zero new cases on the most number of days ?

zero_new_cases_state = df[df['New cases'] == 0]['Name of State / UT'].value_counts().idxmax()
print(f"10. State with Most Days of Zero New Cases: {zero_new_cases_state}")
```

```
10. State with Most Days of Zero New Cases: Mizoram
```

```
# Q14. Average number of new cases per day across India ?

average_new_cases_per_day = df.groupby('Date')['New cases'].sum().mean()
print(f"11. Average New Cases Per Day: {average_new_cases_per_day:.2f}")
```

```
11. Average New Cases Per Day: 10560.61
```

```
# Q15. Find the day with the maximum number of recoveries ?

day_max_recoveries = df.groupby('Date')['New recovered'].sum().idxmax()
print(f"12. Day with Maximum Recoveries: {day_max_recoveries}")
```

```
12. Day with Maximum Recoveries: 2020-08-05 00:00:00
```

```
# Q16. States that reported no deaths ?

states_no_deaths = df.groupby('Name of State / UT')['Death'].max()
states_no_deaths = states_no_deaths[states_no_deaths == 0].index.tolist()
print(f"13. States with No Deaths: {states_no_deaths}")
```

```
13. States with No Deaths: ['Mizoram', 'Union Territory of Chandigarh', 'Union Territory of Jammu and Kashmir', 'Union Territory of Lada
```

```
# Q17. Median of daily new cases across India ?

median_new_cases = df.groupby('Date')['New cases'].sum().median()
print(f"14. Median of Daily New Cases: {median_new_cases}")
```

```
14. Median of Daily New Cases: 1968.5
```

```
# Q18. First date when a death was reported ?

first_death_date = df[df['New deaths'] > 0]['Date'].min()
print(f"16. First Date when Death was Reported: {first_death_date}")
```

```
16. First Date when Death was Reported: NaT
```

```
# 19. Total number of days recorded for each state ?

days_recorded_per_state = df['Name of State / UT'].value_counts()
print(f"18. Days Recorded per State:\n{days_recorded_per_state}")
```

```
18. Days Recorded per State:
Name of State / UT
Kerala                                      186
Delhi                                       154
Haryana                                     152
Rajasthan                                   152
Uttar Pradesh                               152
Tamil Nadu                                  149
Maharashtra                                 147
Karnataka                                   147
Punjab                                      147
Andhra Pradesh                              144
Uttarakhand                                 141
```

```
Odisha                                      140
Puducherry                                  138
Chhattisgarh                                137
Gujarat                                     136
Jammu and Kashmir                           135
Ladakh                                      135
Madhya Pradesh                              135
Himachal Pradesh                            135
Chandigarh                                  135
Bihar                                       134
Manipur                                     132
Mizoram                                     131
Andaman and Nicobar Islands                 130
Goa                                         130
West Bengal                                 128
Assam                                       124
Jharkhand                                   124
Arunachal Pradesh                           122
Tripura                                     118
Meghalaya                                   111
Telengana                                   102
Dadra and Nagar Haveli and Daman and Diu     89
Sikkim                                       71
Nagaland                                     69
Telangana                                    51
Union Territory of Ladakh                    14
Union Territory of Jammu and Kashmir         12
Union Territory of Chandigarh                 2
Telangana***                                  1
Name: count, dtype: int64
```

```
# Q20. State with the highest number of new deaths in a single day ?

state_max_new_deaths = df.loc[df['New deaths'].idxmax(), 'Name of State / UT']
print(f"17. State with Maximum New Deaths in a Single Day: {state_max_new_deaths}")
```

```
17. State with Maximum New Deaths in a Single Day: Kerala
```