In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv(r"C:\Users\karan\OneDrive\Desktop\Latest Covid-19 India Status.
```

In [4]:
```
df.head()
```

Out[4]:

	State/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio	Death Ratio	Populatio
0	Andaman and Nicobar	10766	0	10637	129	0.0	98.80	1.20	1008966
1	Andhra Pradesh	2340676	0	2325943	14733	0.0	99.37	0.63	1285003
2	Arunachal Pradesh	67049	0	66753	296	0.0	99.56	0.44	6580
3	Assam	746159	5	738119	8035	0.0	98.92	1.08	2904
4	Bihar	855267	1	842952	12314	0.0	98.56	1.44	401003

In [8]:
```
df.shape
```

Out[8]:
```
(36, 9)
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   State/UTs        36 non-null     object 
 1   Total Cases      36 non-null     int64  
 2   Active           36 non-null     int64  
 3   Discharged       36 non-null     int64  
 4   Deaths           36 non-null     int64  
 5   Active Ratio     36 non-null     float64
 6   Discharge Ratio  36 non-null     float64
 7   Death Ratio      36 non-null     float64
 8   Population       36 non-null     int64  
dtypes: float64(3), int64(5), object(1)
memory usage: 2.7+ KB
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
State/UTs          0
Total Cases        0
Active             0
Discharged         0
Deaths             0
Active Ratio       0
Discharge Ratio    0
Death Ratio        0
Population         0
dtype: int64
```

In [12]:
```
df.describe()
```

Out[12]:

	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio
count	3.600000e+01	36.000000	3.600000e+01	36.000000	36.000000	36.000000
mean	1.249975e+06	50.333333	1.235146e+06	14778.527778	0.005000	98.897222
std	1.846038e+06	207.118048	1.820328e+06	27221.140636	0.026673	0.506854
min	1.076600e+04	0.000000	1.063700e+04	4.000000	0.000000	97.410000
25%	1.065430e+05	0.000000	1.055395e+05	1124.250000	0.000000	98.690000
50%	6.140910e+05	1.500000	6.076750e+05	6551.000000	0.000000	98.935000
75%	1.331951e+06	9.000000	1.322329e+06	14325.750000	0.000000	99.160000
max	8.171048e+06	1233.000000	8.022276e+06	148558.000000	0.160000	99.970000

In [13]:
```
df.rename(columns={'State/UTs':'States'},inplace=True)
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['States', 'Total Cases', 'Active', 'Discharged', 'Deaths',
       'Active Ratio', 'Discharge Ratio', 'Death Ratio', 'Population'],
      dtype='object')
```

In [15]:
```
print(f"Total Cases:{df["Total Cases"].sum()}")
```

```
Total Cases:44999085
```

In [16]:
```
print(f"Total Active Cases:{df['Active'].sum()}")
```

```
Total Active Cases:1812
```

In [17]:
```
print(f"Total Discharged :{df['Discharged'].sum()}")
```

```
Total Discharged :44465246
```

In [18]:
```
print(f"Total Deaths:{df['Deaths'].sum()}")
```

```
Total Deaths:532027
```

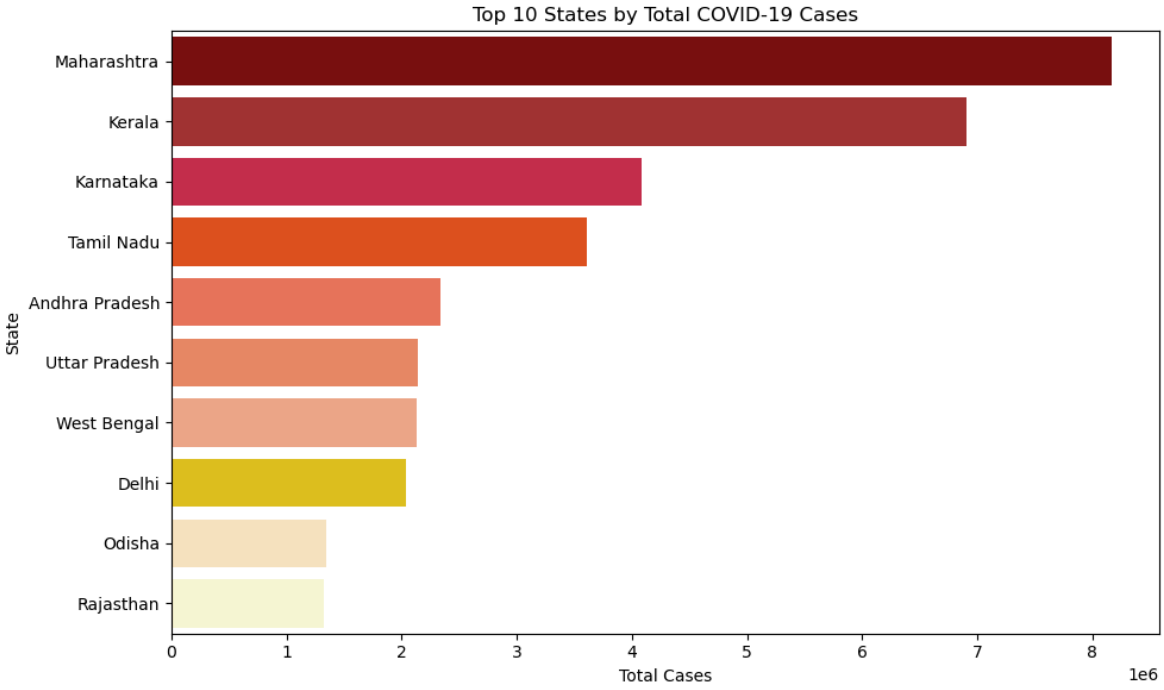
In [24]:
```
top10 = df.sort_values("Total Cases",ascending=False).head(10)
print(top10)
```

	States	Total Cases	Active	Discharged	Deaths	Active Ratio	\
20	Maharashtra	8171048	214	8022276	148558	0.00	
16	Kerala	6907241	18	6835181	72042	0.00	
15	Karnataka	4088769	12	4048399	40358	0.00	
30	Tamil Nadu	3610655	5	3572569	38081	0.00	
1	Andhra Pradesh	2340676	0	2325943	14733	0.00	
33	Uttar Pradesh	2145431	57	2121662	23712	0.00	
35	West Bengal	2126282	135	2104592	21555	0.01	
8	Delhi	2040910	14	2014230	26666	0.00	
25	Odisha	1348409	59	1339135	9215	0.00	
28	Rajasthan	1326465	2	1316727	9736	0.00	

	Discharge Ratio	Death Ratio	Population
20	98.18	1.82	399001
16	98.96	1.04	91702478
15	99.01	0.99	1711947
30	98.95	1.05	35998752
1	99.37	0.63	128500364
33	98.89	1.11	1158040
35	98.98	1.01	32199722
8	98.69	1.31	773997
25	99.31	0.68	19301096
28	99.27	0.73	1521992

In [35]:
```
top10 = df.sort_values("Total Cases", ascending=False).head(10)
custom_colors = [
    "#8B0000",  # Dark Red
    "#B22222",  # Firebrick
    "#DC143C",  # Crimson
    "#FF4500",  # Orange Red
    "#FF6347",  # Tomato
    "#FF7F50",  # Coral
    "#FFA07A",  # Light Salmon
    "#FFD700",  # Gold
    "#FFE4B5",  # Moccasin
    "#FFFACD"   # Lemon Chiffon
]


plt.figure(figsize=(10, 6))
sns.barplot(data=top10, x="Total Cases", y="States",palette=custom_colors)
plt.title("Top 10 States by Total COVID-19 Cases")
plt.xlabel("Total Cases")
plt.ylabel("State")
plt.figtext(0.5, -0.05, "Note: 1 unit = 10,00,000 cases", wrap=True, horizontala
plt.tight_layout()
plt.show()
```



Note: 1 unit = 10,00,000 cases

In [39]:
```
total_active_cases=df["Active"].sum()
print("Total active cases :",total_active_cases)
```

```
Total active cases : 1812
```

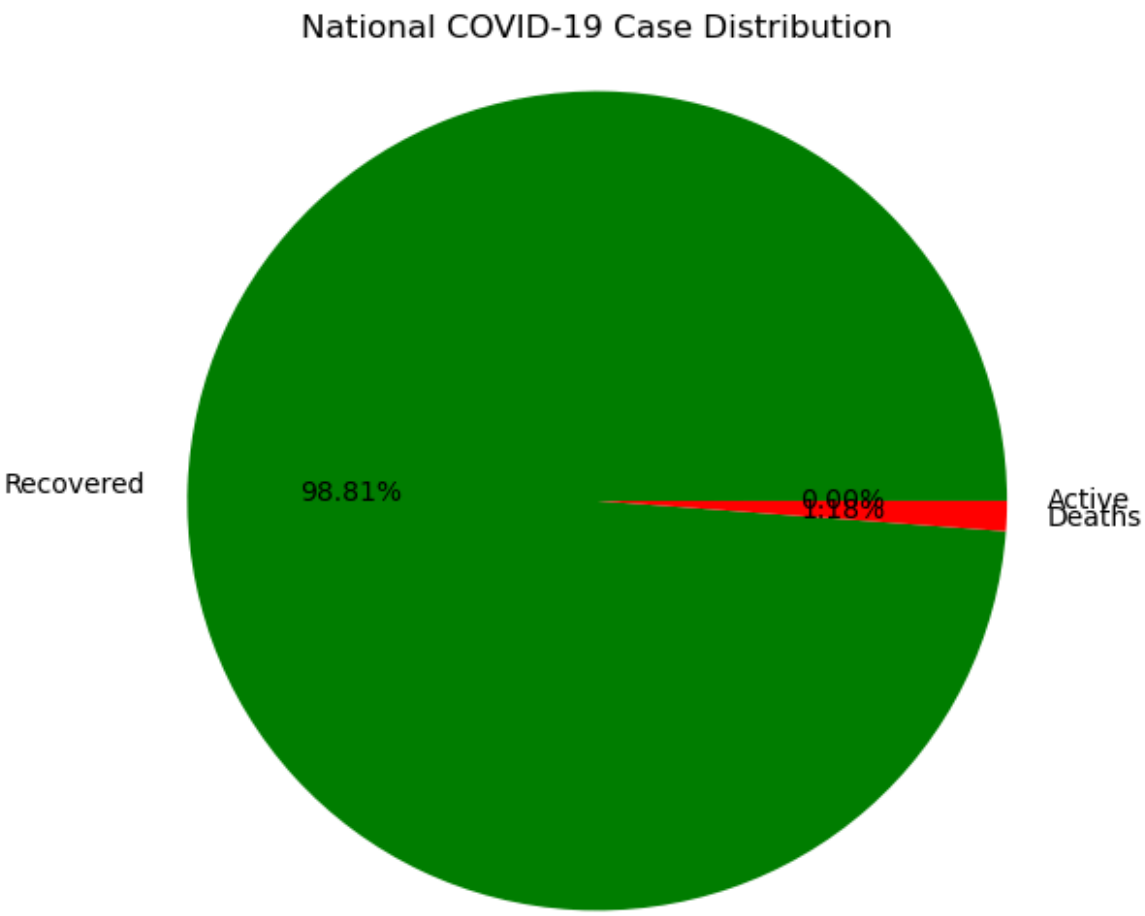
In [44]:
```
total_discharged=df['Discharged'].sum()
print("Total Recovered:",total_discharged)
```

```
Total Recovered: 44465246
```

In [43]:
```
total_Deaths=df['Deaths'].sum()
print("Total Death:",total_Deaths)
```

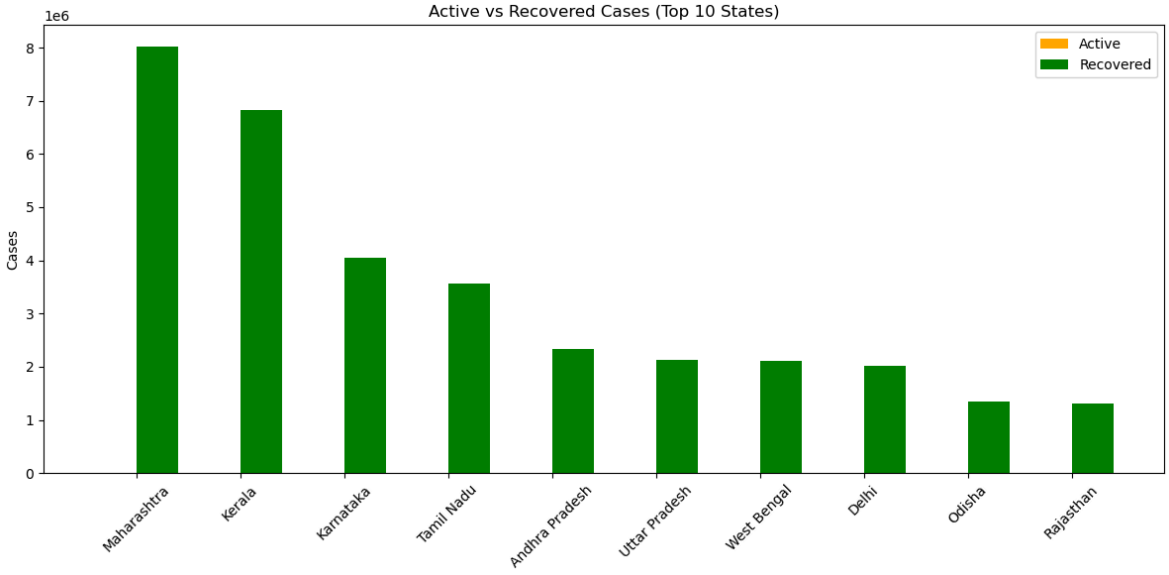
```
Total Death: 532027
```

In [45]:
```
plt.figure(figsize=(6,6))
plt.pie([total_active_cases,total_discharged,total_Deaths],
        labels=["Active","Recovered","Deaths"],
       autopct='%1.2f%%',colors=["orange","green","red"])
plt.title("National COVID-19 Case Distribution")
plt.axis("equal")
plt.show()
```



In [47]:
```
top10 = df.sort_values("Total Cases", ascending=False).head(10)

plt.figure(figsize=(12, 6))
x = range(len(top10))
plt.bar(x, top10['Active'], width=0.4, label='Active', align='center', color='or
plt.bar([i + 0.4 for i in x], top10['Discharged'], width=0.4, label='Recovered',
plt.xticks([i + 0.2 for i in x], top10['States'], rotation=45)
plt.title("Active vs Recovered Cases (Top 10 States)")
plt.ylabel("Cases")
plt.legend()
plt.tight_layout()
plt.show()
```



In [ ]: