

# Sai Ram

## 20181CSE0621

### 7 - CSE - 10

### DV-Assignment 2

## Importing Necessary Libraries

```
In [ ]: 1 import numpy as np
        2 import pandas as pd
        3 import matplotlib.pyplot as plt
        4 %matplotlib inline
        5 import seaborn as sns
        6 import datetime
        7 from matplotlib import rc,rcParams
        8 plt.style.use(['dark_background', 'ggplot', 'seaborn-ticks'])
        9 import warnings; warnings.simplefilter('ignore')
```

## Loading Covid Data

```
In [ ]: 1 raw = pd.read_csv(r'covid_19_india.csv')
        2 raw.head()
        3 raw['Date'] = pd.to_datetime(raw['Date'])
        4 raw.dtypes
        5 # Choosing date from June 2021
        6 df = raw.loc[(raw['Date'] >= '2021-06-01')]
        7 df.columns
```

```
Out[42]: Index(['Sno', 'Date', 'Time', 'State/UnionTerritory',
                'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
                'Deaths', 'Confirmed'],
                dtype='object')
```

```
In [ ]: 1 df.isnull().sum()
```

```
Out[43]: Sno                0
        Date                0
        Time                0
        State/UnionTerritory 0
        ConfirmedIndianNational 0
        ConfirmedForeignNational 0
        Cured                0
        Deaths              0
        Confirmed            0
        dtype: int64
```

In [ ]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 288 entries, 15518 to 15805
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Sno                                    288 non-null    int64
1   Date                                  288 non-null    datetime64[ns]
2   Time                                  288 non-null    object
3   State/UnionTerritory                 288 non-null    object
4   ConfirmedIndianNational             288 non-null    object
5   ConfirmedForeignNational            288 non-null    object
6   Cured                                288 non-null    int64
7   Deaths                              288 non-null    int64
8   Confirmed                            288 non-null    int64
dtypes: datetime64[ns](1), int64(4), object(4)
memory usage: 22.5+ KB
```

## Deriving Active Cases

In [ ]:

```
1 df = df.drop(['Sno', 'ConfirmedIndianNational', 'ConfirmedForeignNational'], axis=1)
2 df['Active'] = df['Confirmed'] - df['Cured'] - df['Deaths']
3 display(df.info())
4 display(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 288 entries, 15518 to 15805
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                  288 non-null    datetime64[ns]
1   Time                                  288 non-null    object
2   State/UnionTerritory                 288 non-null    object
3   Cured                                288 non-null    int64
4   Deaths                              288 non-null    int64
5   Confirmed                            288 non-null    int64
6   Active                               288 non-null    int64
dtypes: datetime64[ns](1), int64(4), object(2)
memory usage: 18.0+ KB
```

None

	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Active
15518	2021-06-01	8:00 AM	Andaman and Nicobar Islands	6719	115	7005	171
15519	2021-06-01	8:00 AM	Andhra Pradesh	1528360	10930	1693085	153795
15520	2021-06-01	8:00 AM	Arunachal Pradesh	23402	115	27272	3755
15521	2021-06-01	8:00 AM	Assam	354810	3365	411216	53041
15522	2021-06-01	8:00 AM	Bihar	685362	5163	706761	16236

## Making State as index

```
In [ ]: 1 india_cases = df[df['Date'] == df['Date'].max()].copy().fillna(0)
2 india_cases.index = india_cases["State/UnionTerritory"]
3 india_cases = india_cases.drop(['State/UnionTerritory', 'Date'], axis=1)
4 india_cases.head()
```

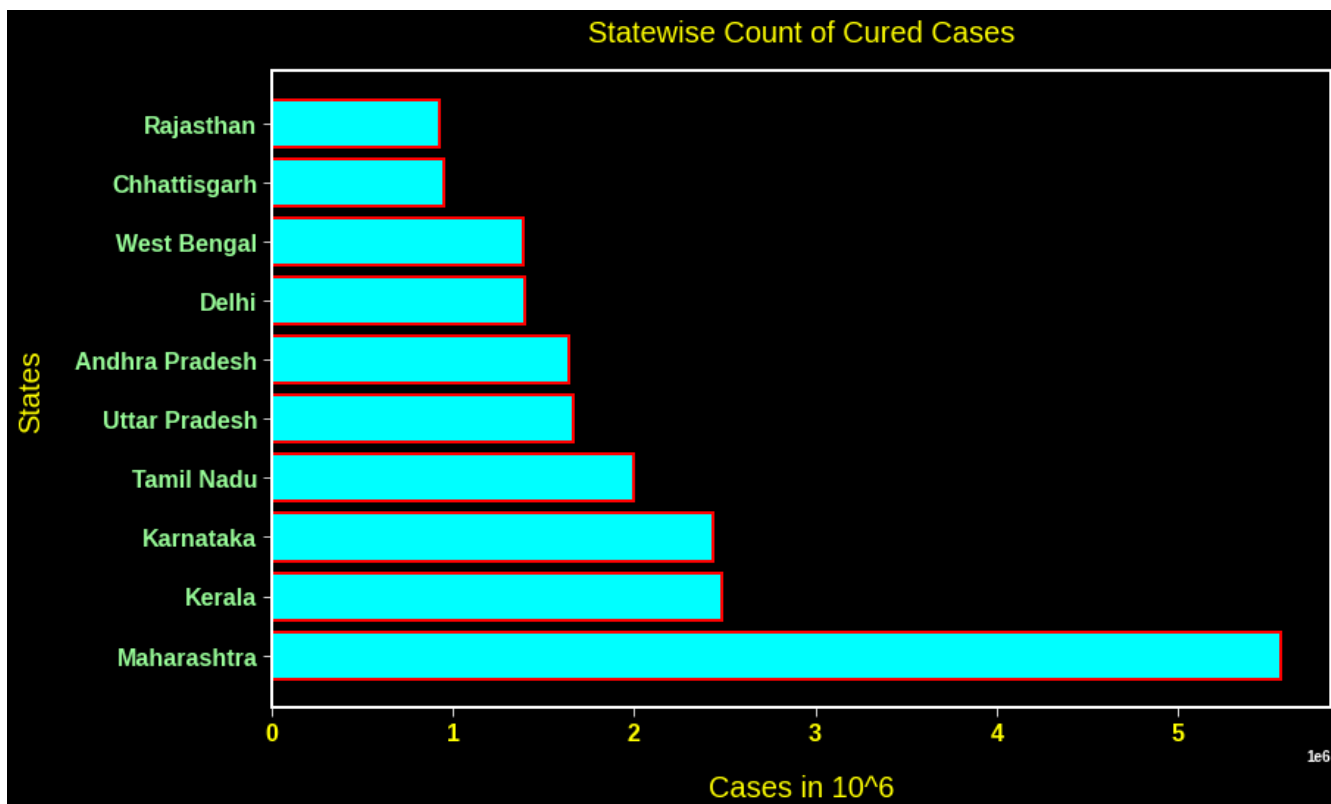
Out[46]:

	Time	Cured	Deaths	Confirmed	Active
State/UnionTerritory					
Andaman and Nicobar Islands	8:00 AM	6912	123	7131	96
Andhra Pradesh	8:00 AM	1637149	11552	1763211	114510
Arunachal Pradesh	8:00 AM	26131	125	29696	3440
Assam	8:00 AM	385032	3695	438746	50019
Bihar	8:00 AM	700224	5424	713879	8231

# 1 - Number of Cured Cases State-Wise

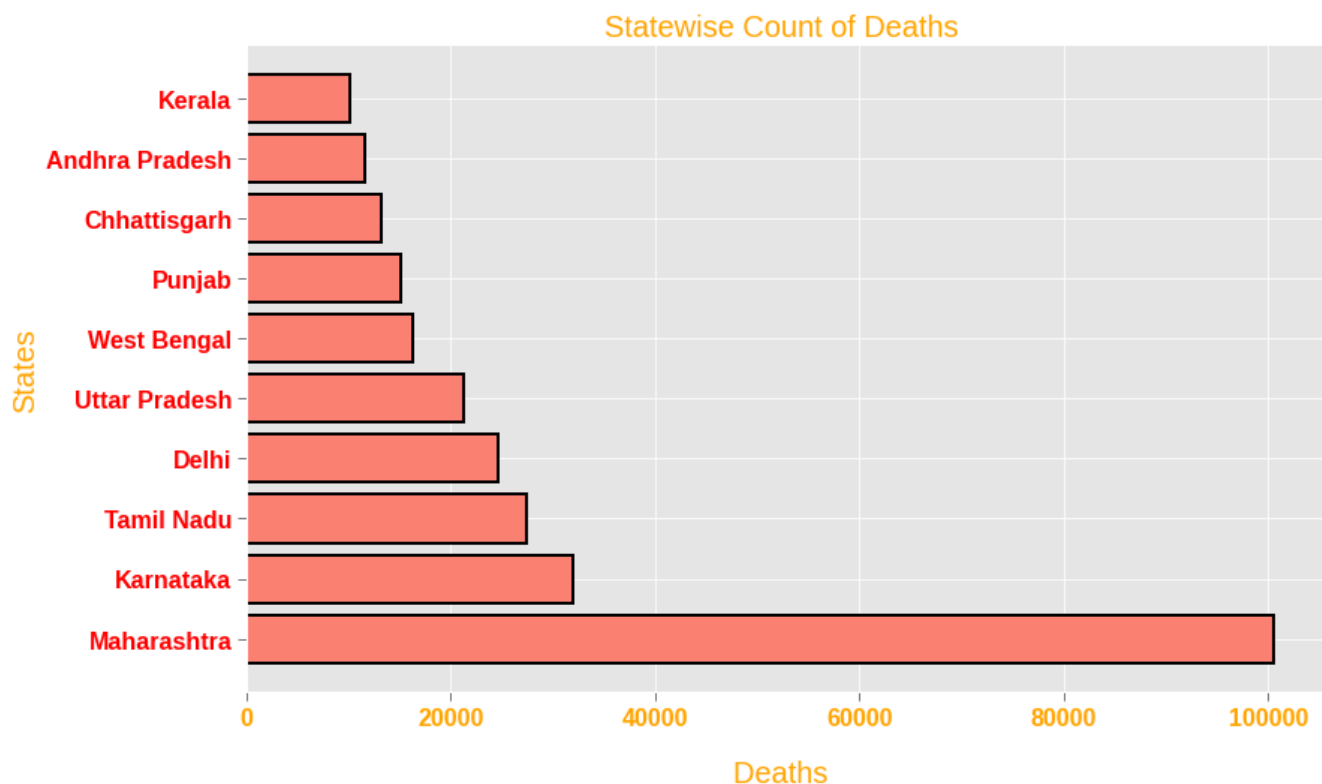
```
In [ ]: 1 rc('axes', linewidth=2)
2 rc('font', weight='bold')
3 rcParams['text.latex.preamble'] = [r'\usepackage{sfbmath} \boldmath']
```

```
In [ ]: 1 top_10_cured_cases = india_cases.sort_values('Cured',ascending = False)[:10]
2 with plt.style.context('dark_background'):
3     top_10_cured_cases.head()
4     fig = plt.figure(figsize=(10,6))
5     ax = fig.add_axes([0,0,1,1])
6     ax.barh(top_10_cured_cases.index,top_10_cured_cases.Cured,color='cyan',edgecolor=
7     plt.ylabel("States",size=20,color='yellow',labelpad=20) ; plt.xlabel("Cases in 10
8     plt.title('Statewise Count of Cured Cases',size=20,color='yellow',pad=20)
9     plt.xticks(fontsize=16,color='yellow') ; plt.yticks(fontsize=16,color='palegreen'
10    plt.tight_layout()
11    plt.show()
```



## 2 - Number of Deaths State-Wise

```
In [ ]: 1 top_10_death_states = india_cases.sort_values('Deaths',ascending = False)[:10]
2
3 with plt.style.context('ggplot'):
4     top_10_death_states.head()
5     fig = plt.figure(figsize=(10,6))
6     ax = fig.add_axes([0,0,1,1])
7     ax.barh(top_10_death_states.index,top_10_death_states.Deaths,color='salmon',edgec
8     plt.ylabel("States",size=20,color='Orange') ; plt.xlabel("Deaths ",size=20,color=
9     plt.title('Statewise Count of Deaths ',size=20,color='Orange')
10    plt.xticks(fontsize=16,color='Orange') ; plt.yticks(fontsize=16,color='red')
11    plt.tight_layout()
12    plt.show()
```

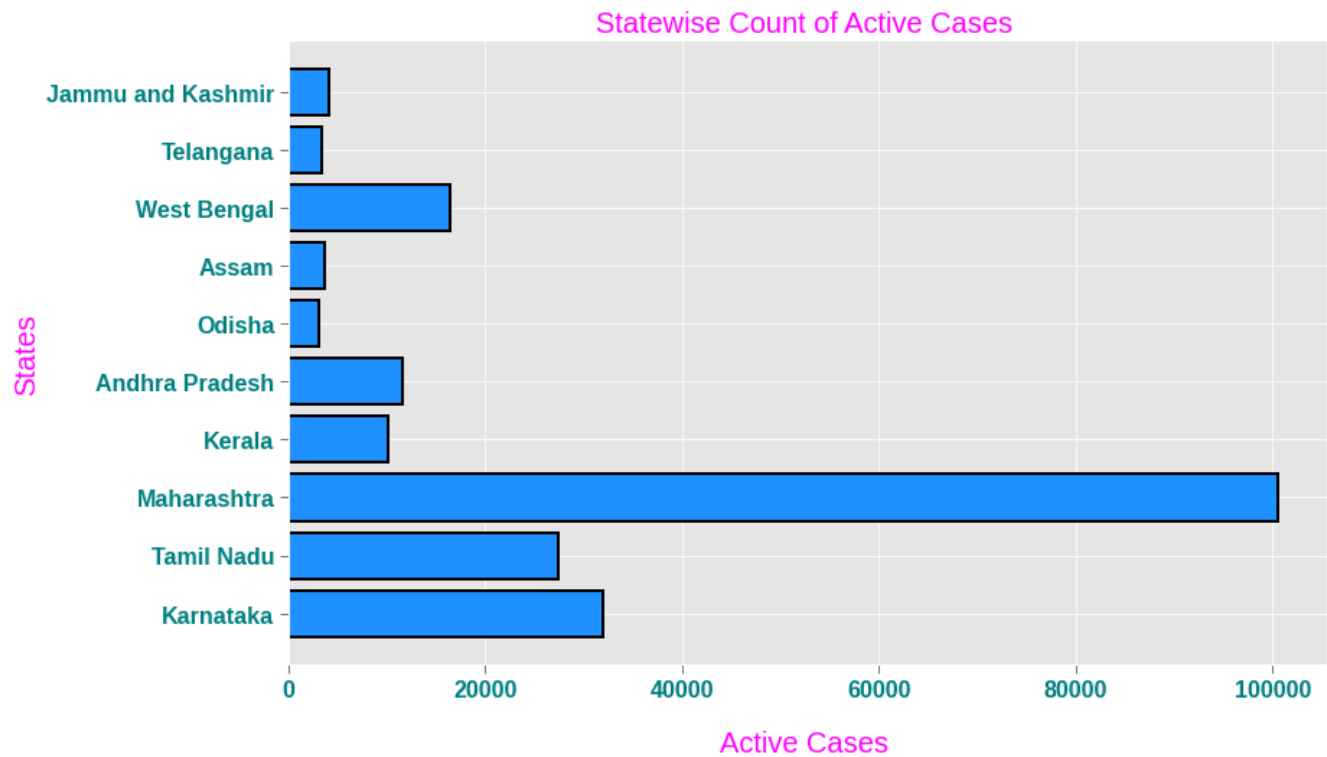


### 3 - Number of Active Cases State Wise

```

In [ ]: 1 top_10_active_states = india_cases.sort_values('Active',ascending = False)[:10]
        2
        3 with plt.style.context('ggplot'):
        4     fig = plt.figure(figsize=(10,6))
        5     ax = fig.add_axes([0,0,1,1])
        6     ax.barh(top_10_active_states.index,top_10_active_states.Deaths,color='dodgerblue')
        7     plt.ylabel("States",size=20,color='magenta') ; plt.xlabel("Active Cases ",size=20)
        8     plt.title('Statewise Count of Active Cases ',size=20,color='magenta')
        9     plt.xticks(fontsize=16,color='teal') ; plt.yticks(fontsize=16,color='teal')
       10     plt.tight_layout()
       11 plt.show()

```



## Loading Vaccination Data

```
In [ ]: 1 vaccination = pd.read_csv(r'covid_vaccine_statewise.csv')
        2 display(vaccination.shape)
        3 display(vaccination.isna().sum())
        4 vaccination = vaccination.drop(['Total Sputnik V Administered', 'AEFI', '18-45 years
        5 vaccination.columns
```

(5328, 18)

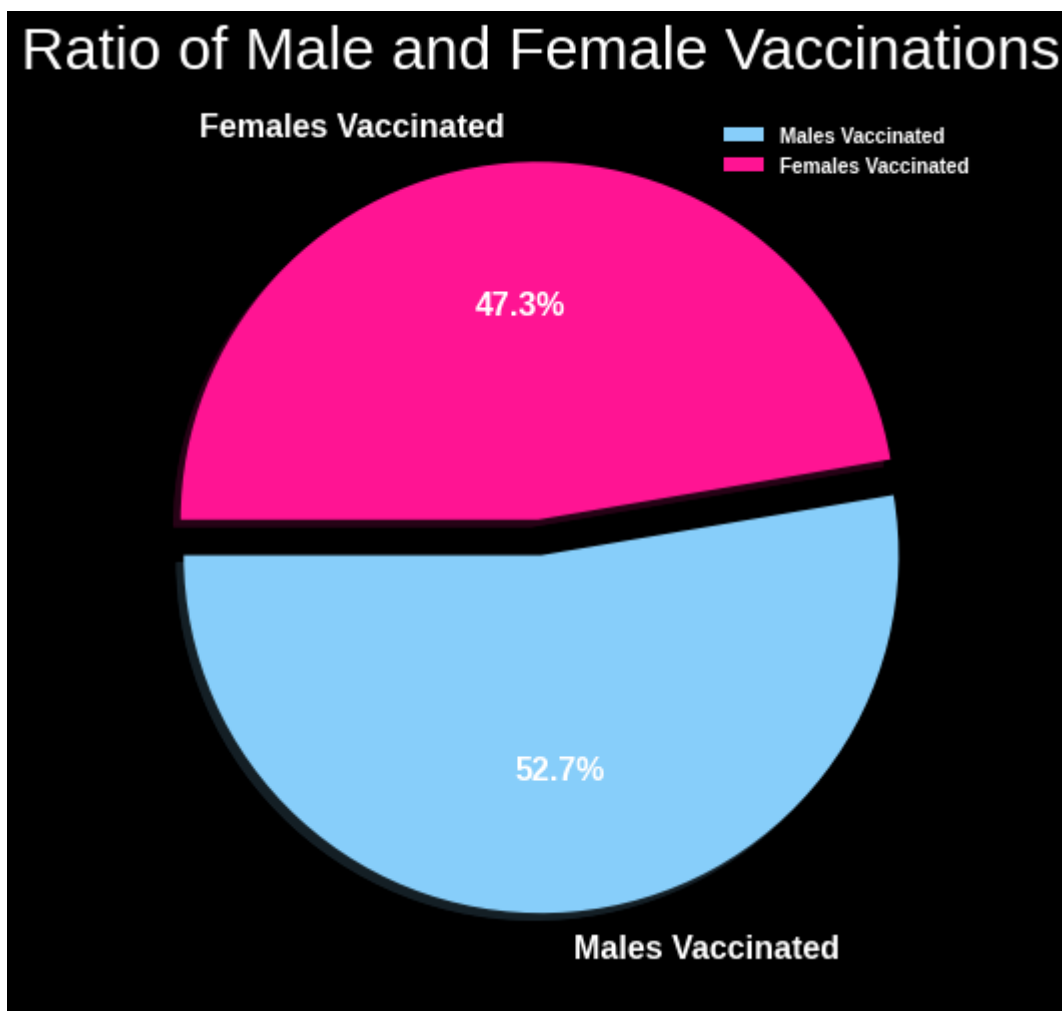
Updated On	0
State	0
Total Individuals Vaccinated	40
Total Sessions Conducted	40
Total Sites	40
First Dose Administered	40
Second Dose Administered	40
Male(Individuals Vaccinated)	40
Female(Individuals Vaccinated)	40
Transgender(Individuals Vaccinated)	40
Total Covaxin Administered	40
Total CoviShield Administered	40
Total Sputnik V Administered	4666
AEFI	2221
18-45 years (Age)	2226
45-60 years (Age)	2225
60+ years (Age)	2225
Total Doses Administered	37
dtype:	int64

```
Out[30]: Index(['Updated On', 'State', 'Total Individuals Vaccinated',
               'Total Sessions Conducted', 'Total Sites ', 'First Dose Administered',
               'Second Dose Administered', 'Male(Individuals Vaccinated)',
               'Female(Individuals Vaccinated)', 'Transgender(Individuals Vaccinated)',
               'Total Covaxin Administered', 'Total CoviShield Administered',
               'Total Doses Administered'],
              dtype='object')
```

## 4 - Ratio of Males & Females Vaccinated

In [ ]:

```
1 male = vaccination["Male(Individuals Vaccinated)"].sum()
2 female = vaccination["Female(Individuals Vaccinated)"].sum()
3
4
5 with plt.style.context('dark_background'):
6     fig = plt.figure(figsize =(8, 10))
7     labels = ["Males Vaccinated","Females Vaccinated"]
8     explode =(0,0.1)
9     plt.pie([male,female],labels=labels,colors=['lightskyblue','deeppink'],
10            startangle=180,explode=explode, autopct='%0.1f%%', shadow = True,textprops
11     plt.legend(labels, loc = 'upper right',prop={'size': 10})
12     plt.title('Ratio of Male and Female Vaccinations',pad=20,size=30)
13     plt.show()
```

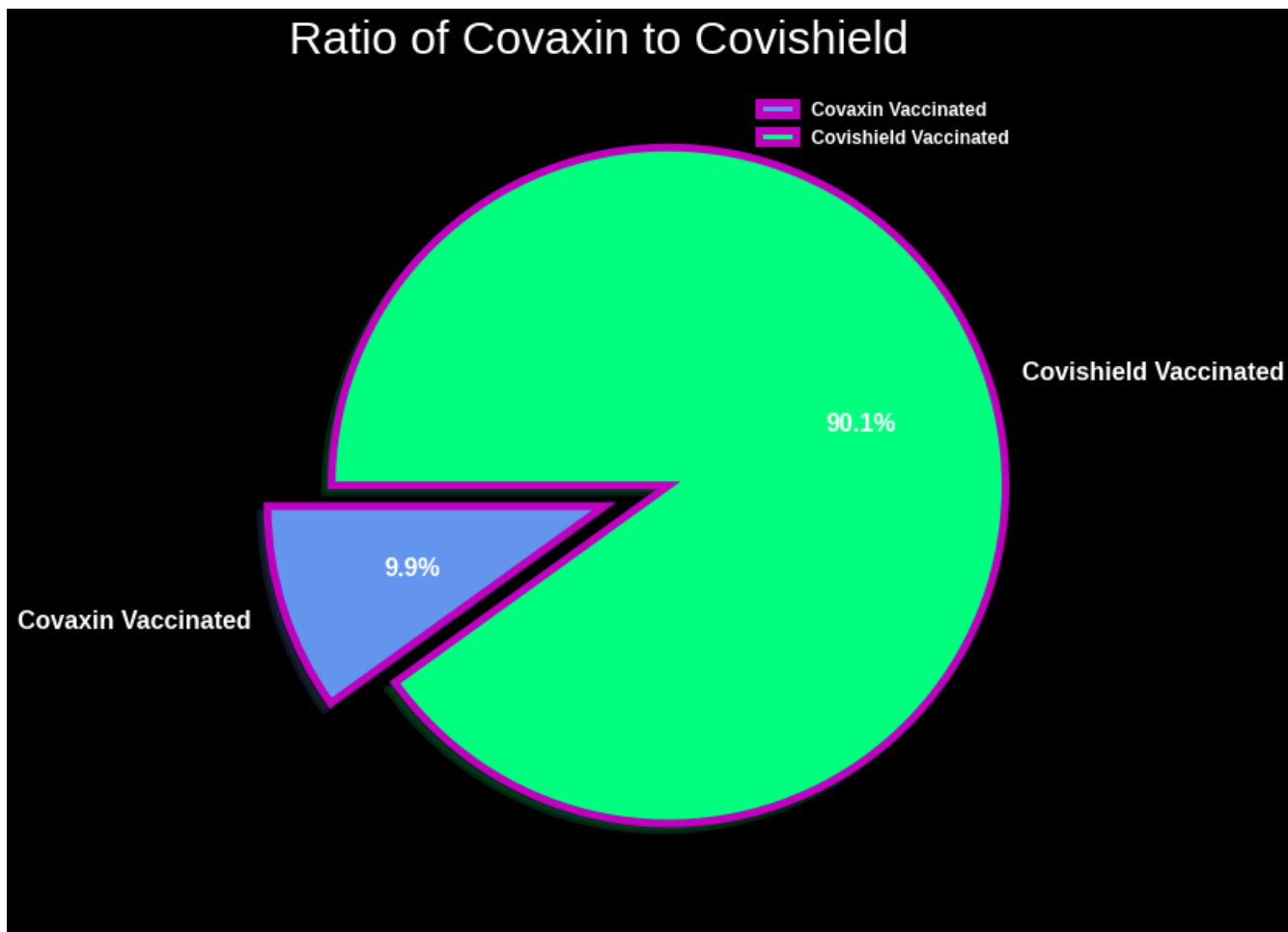




## 5 - Ratio of Covaxin & Covishield

In [ ]:

```
1 Covaxin = vaccination["Total Covaxin Administered"].sum()
2 Covishield = vaccination["Total CoviShield Administered"].sum()
3 #px.pie(names=["Covaxin Vaccinated", "Covishield Vaccinated"], values=[Covaxin, Covishield])
4
5 with plt.style.context('dark_background'):
6     fig = plt.figure(figsize=(10, 10))
7     labels = ["Covaxin Vaccinated", "Covishield Vaccinated"]
8     explode = (0, 0.2)
9     plt.pie([Covaxin, Covishield], labels=labels, colors=['cornflowerblue', 'springgreen'],
10             startangle=180, explode=explode, autopct='%1f%', shadow = True, textprops=
11             wedgeprops={"edgecolor": "m", 'linewidth': 5, 'linestyle': 'solid', 'antialiased': True})
12     plt.legend(labels, loc = 'upper right', prop={'size': 12})
13     plt.title('Ratio of Covaxin to Covishield ', pad=20, size=30)
14     plt.show()
```



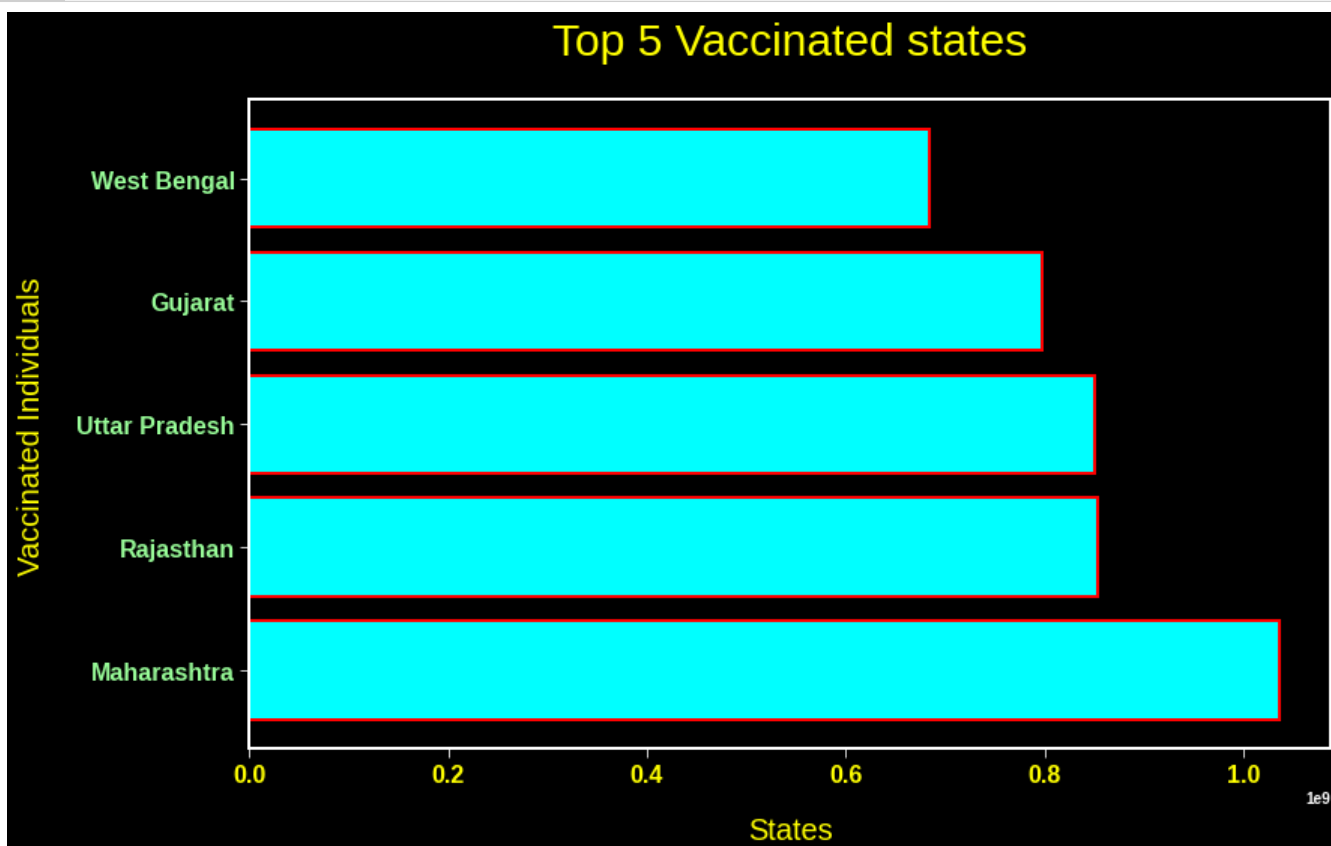
## 6 - Top 5 Vaccinated States

```
In [ ]: 1 v=vaccination[vaccination.State!='India']
2 v.rename(columns = {"Total Individuals Vaccinated": "Total"}, inplace = True)
3 States_grouped=v.groupby('State')['Total'].sum().to_frame('Total')
4 States_grouped=States_grouped.sort_values('Total',ascending=False).head(5)
5 States_grouped
```

```
Out[33]:
```

	Total
State	
Maharashtra	1.036216e+09
Rajasthan	8.529093e+08
Uttar Pradesh	8.506090e+08
Gujarat	7.976807e+08
West Bengal	6.839649e+08

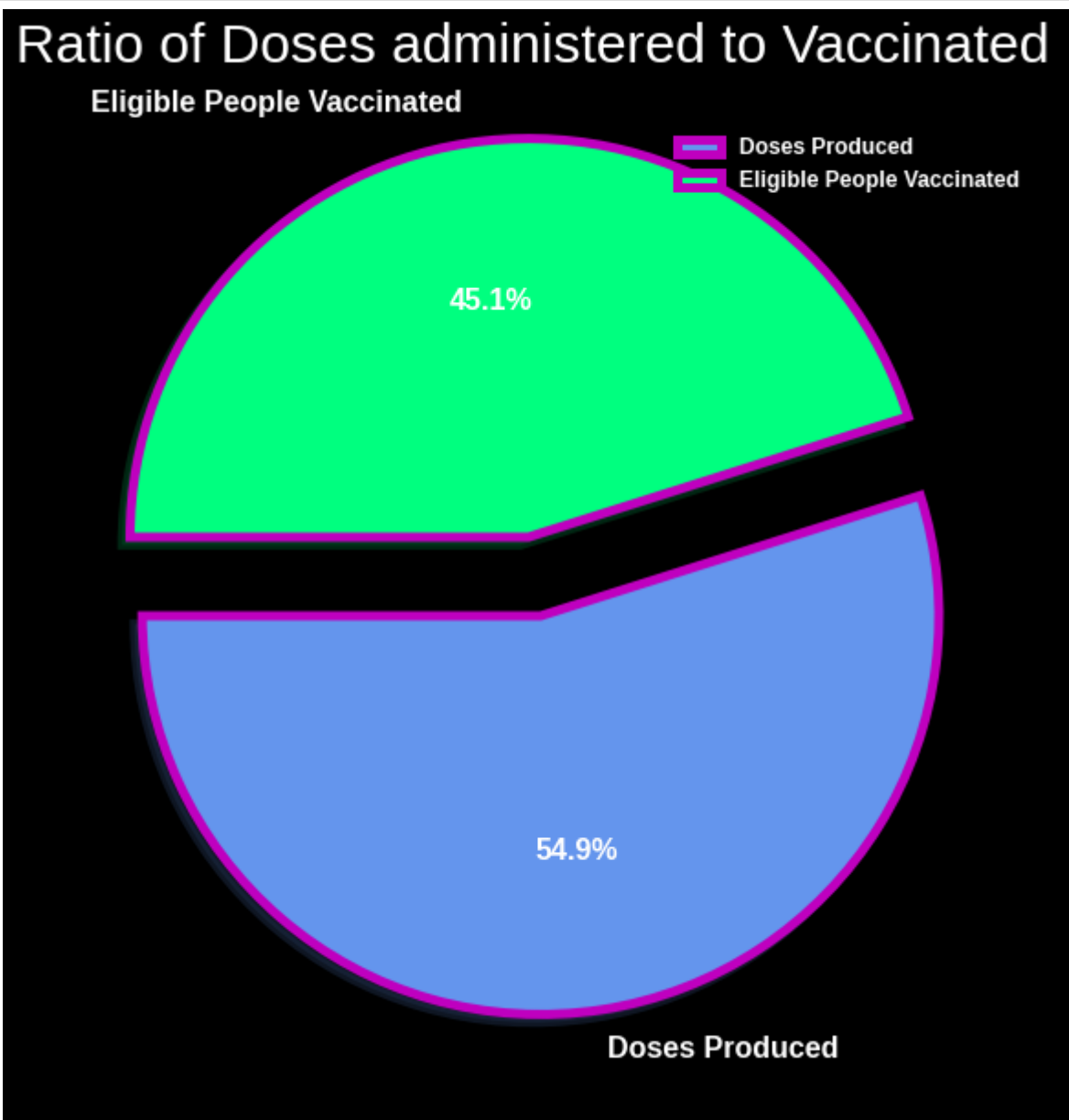
```
In [ ]: 1 with plt.style.context('dark_background'):
2 fig=plt.figure(figsize=(10,6))
3 ax = fig.add_axes([0,0,1,1])
4 ax.barh(States_grouped.index,States_grouped.Total,color='cyan',edgecolor='red',li
5 plt.title("Top 5 Vaccinated states", size=30,pad=30,color='yellow')
6 plt.xlabel("States",size=20,color='yellow',labelpad=20) ; plt.ylabel("Vaccinated
7 plt.xticks(fontsize=16,color='yellow') ; plt.yticks(fontsize=16,color='palegreen'
8 plt.tight_layout()
9 plt.show()
```



## 7 - Doses of Vaccine

In [ ]:

```
1 Doses = vaccination["Total Doses Administered"].sum()
2 Vaccinated = vaccination["Total Individuals Vaccinated"].sum()
3 #Out of 55 doses produced 45 are vaccinated
4
5 with plt.style.context('dark_background'):
6     fig = plt.figure(figsize =(10, 10))
7     labels = ["Doses Produced","Eligible People Vaccinated"]
8     explode =(0,0.2)
9     plt.pie([Doses,Vaccinated],labels=labels,colors=['cornflowerblue','springgreen'],
10            startangle=180,explode=explode, autopct='%1f%%', shadow = True,textprops
11            wedgeprops={"edgecolor":"m", 'linewidth': 5, 'linestyle': 'solid', 'antialiased': True})
12     plt.legend(labels, loc = 'upper right',prop={'size': 12})
13     plt.title('Ratio of Doses administered to Vaccinated ',pad=30,size=30)
14     plt.show()
```



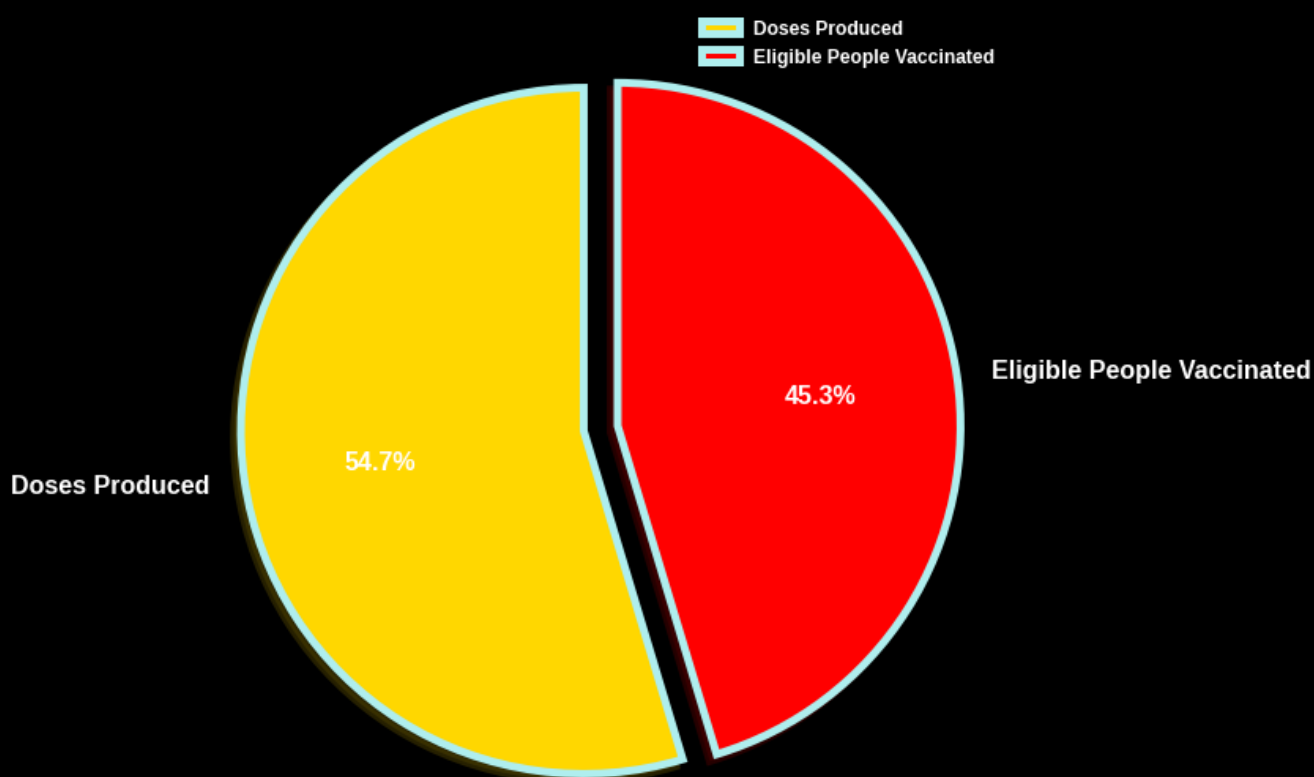
```
In [ ]: 1 #x = vaccination['Total Individuals Vaccinated']
2 #y = vaccination['Total Doses Administered']
3 #d={}
4 #for a,b in zip(x,y):
5 #     if (a==b):continue
6 #     else : d[a]=b
7 #print(d)
8 print(Doses,Vaccinated) ; print((Vaccinated/Doses))
```

743461418.0 616699028.0  
0.8294970163468524

## 8 - Vaccination in Karnataka

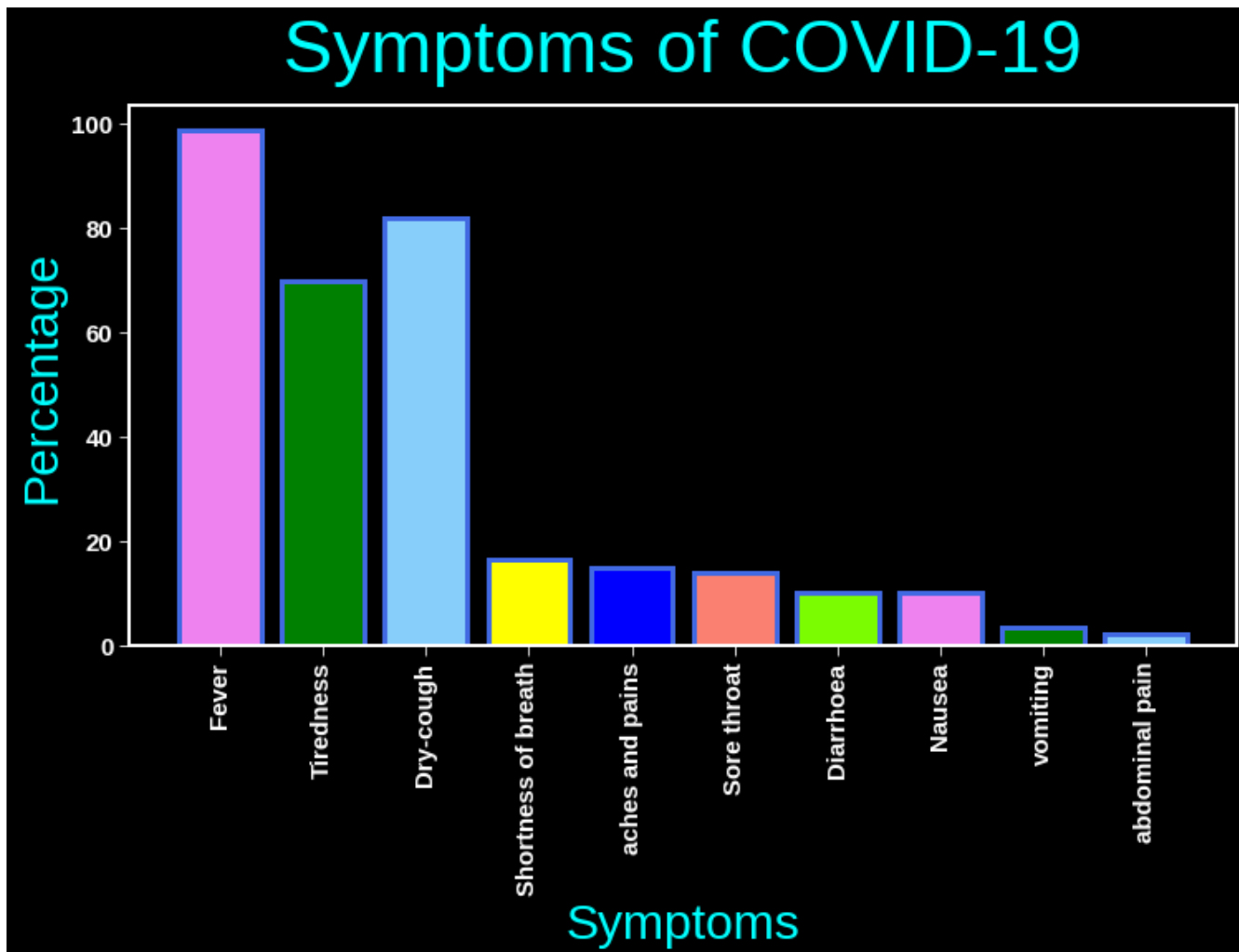
```
In [ ]: 1 df_Karnataka=vaccination[vaccination['State']=='Karnataka']
2 Doses = df_Karnataka["Total Doses Administered"].sum()
3 Vaccinated = df_Karnataka["Total Individuals Vaccinated"].sum()
4
5 with plt.style.context('dark_background'):
6     fig = plt.figure(figsize =(10, 10))
7     labels = ["Doses Produced","Eligible People Vaccinated"]
8     explode =(0,0.1)
9     plt.pie([Doses,Vaccinated],labels=labels,colors=['gold','red'],
10             startangle=90,explode=explode, autopct='%1f%%', shadow = True,textprops=
11             wedgeprops={"edgecolor":"paleturquoise",'linewidth': 5, 'linestyle': 'sol
12 plt.legend(labels, loc = 'upper right',prop={'size': 12})
13 plt.title('Ratio of Doses administered to Vaccinated In Karnataka',pad=20,size=30
14 plt.show()
15
16
```

Ratio of Doses administered to Vaccinated In Karnataka

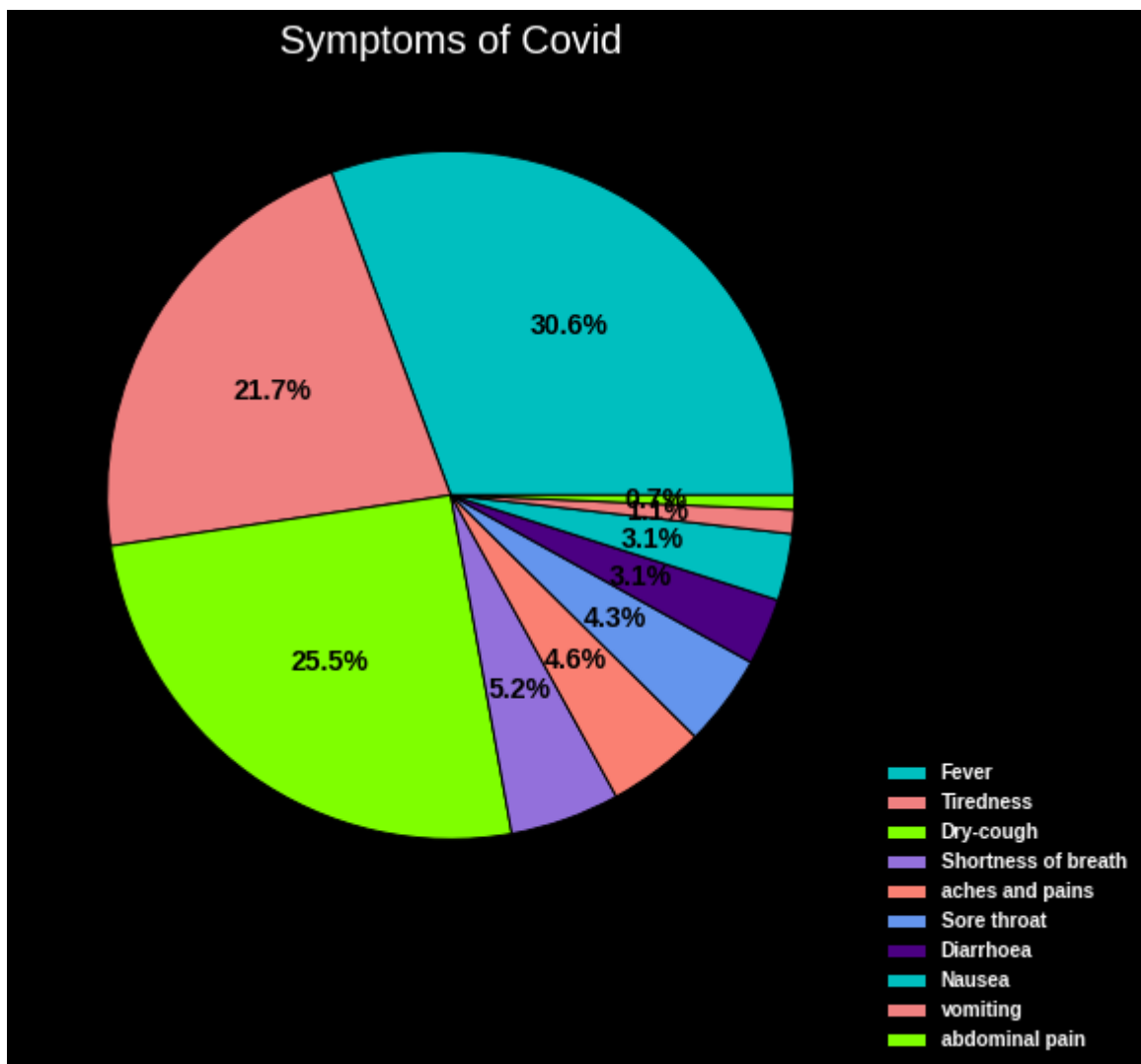


## 9 & 10 - Symptoms of Covid - 19

```
In [ ]: 1 symptoms={'symptoms':['Fever','Tiredness','Dry-cough','Shortness of breath','aches  
2         'percentage':[98.6,69.9,82,16.6,14.8,13.9,10.1,10.1,3.6,2.2]  
3         }  
4 symptoms=pd.DataFrame(data=symptoms,index=range(10))  
5 with plt.style.context('dark_background'):  
6     plt.figure(figsize=(12,6))  
7     height=symptoms.percentage  
8     bars=symptoms.symptoms  
9     y_pos = np.arange(len(bars))  
10    colors = ['violet','green','lightskyblue','yellow','blue','salmon','lawngreen']  
11    plt.bar(y_pos, height,color=colors,edgecolor='royalblue',linewidth=3)  
12    plt.xticks(y_pos, bars)  
13    plt.xticks(rotation=90,size=15) ; plt.yticks(size=15)  
14    plt.xlabel("Symptoms", size=30,color='cyan')  
15    plt.ylabel("Percentage", size=30,color='cyan')  
16    plt.title("Symptoms of COVID-19", size=45,pad=20,color='cyan')  
17    plt.show()
```

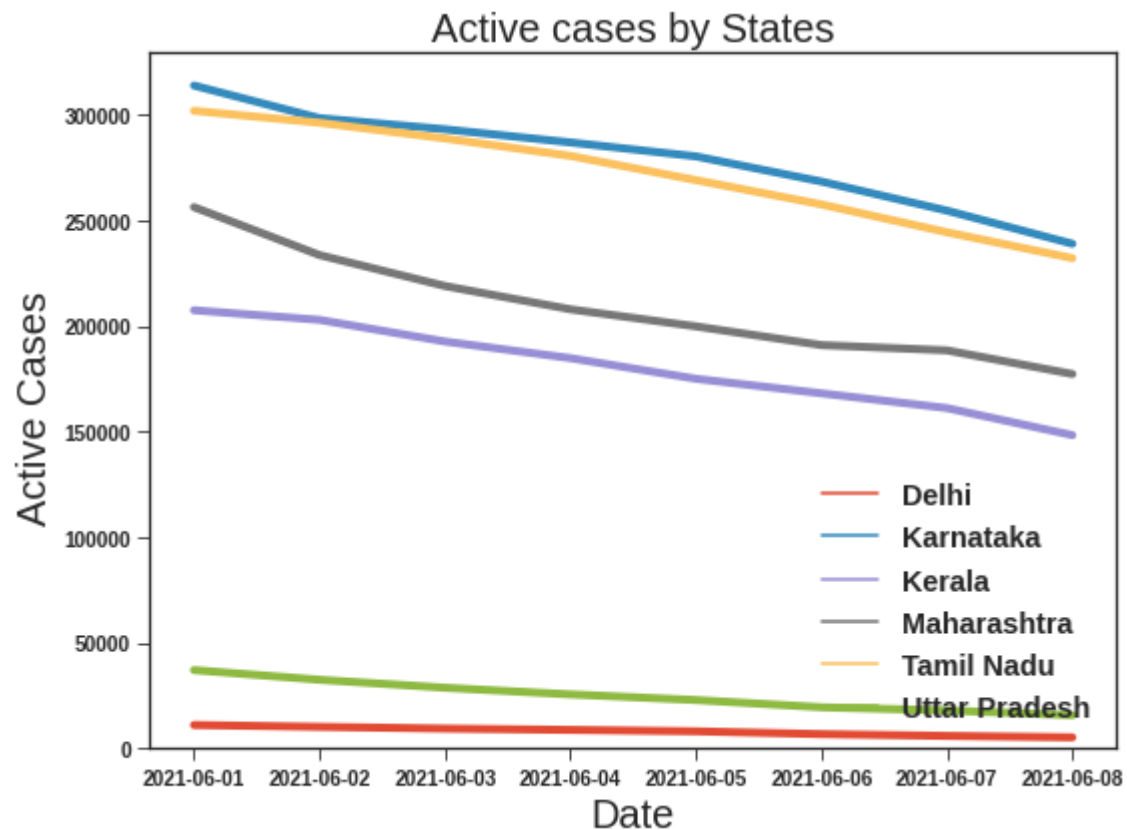


```
In [ ]: 1 with plt.style.context('dark_background'):
2         plt.figure(figsize=(10,8))
3         plt.title("Symptoms of Covid",fontsize=20)
4         colors=['c','lightcoral','chartreuse','mediumpurple','salmon','cornflowerblue','i
5         plt.pie(symptoms["percentage"],colors = colors,autopct="%1.1f%%",
6                 textprops={'fontsize': 14,'color':'black'},wedgeprops= { 'linewidth' : 1
7         #plt.legend(symptoms['symptoms'],loc='upper right')
8         plt.legend(symptoms['symptoms'],bbox_to_anchor=(1,0), loc="lower right",bbox_tran
9         plt.show()
```



## 11 - State-Wise Active Cases of Major Affected States

```
In [ ]: 1 with plt.style.context('seaborn-ticks'):
2         fig=plt.figure(figsize=(8,6))
3         major_states = ['Kerala','Tamil Nadu','Delhi','Maharashtra','Uttar Pradesh','Karnataka']
4         ax=sns.lineplot(data=df[df['State/UnionTerritory'].isin(major_states)],x='Date',y='Active Cases')
5         ax.set_title("Active cases by States", size=20) ; plt.legend(fontsize='x-large',loc='best')
6         ax.set_ylabel("Active Cases ",fontsize=20) ; ax.set_xlabel("Date",fontsize=20)
7         #ax.tick_params(labelsize=5)
8         plt.tight_layout()
9         plt.ylim(0,300000)
10        plt.show()
```



## 12 - Country Wide Cases Distribution

In [ ]:

```
1 x = df['Confirmed'].sum()
2 y = df['Cured'].sum()
3 z = df['Deaths'].sum()
4 active=df['Active'].sum()
5
6 with plt.style.context('seaborn-ticks'):
7     plt.figure(figsize=(10,6))
8     sns.set(font_scale = 2)
9     sns.barplot(x=['Confirmed', 'Cured', 'Deaths', 'Active'], y=[x,y,z,active], palette=['
10
```

