```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
data = pd.read_csv('Unemployment_Rate_upto_11_2020.csv')
```

```
data.head(5)
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longi
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.

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

```
# Check for missing values
print(data.isnull().sum())
```

```
Region                                    0
 Date                                     0
 Frequency                                0
 Estimated Unemployment Rate (%)          0
 Estimated Employed                       0
 Estimated Labour Participation Rate (%)  0
Region.1                                  0
longitude                                 0
latitude                                  0
dtype: int64
```

```
# Convert 'Date' column to datetime format
# Check if 'Date' column exists, otherwise try alternative names
if 'Date' in data.columns:
```

```
    data['Date'] = pd.to_datetime(data['Date'], format='%d-%m-%Y')
else:
    # Try alternative column names like 'date' or 'DATE'
    alternative_names = ['date', 'DATE']
    for name in alternative_names:
        if name in data.columns:
            data['Date'] = pd.to_datetime(data[name], format='%d-%m-%Y')
            print(f"Found date column with name: {name}")
            break
    else:
        print("Date column not found in any of the expected names.")
```

```
Date column not found in any of the expected names.
```

```
# Rename columns for easier access
data.rename(columns={
    'Estimated Unemployment Rate (%)': 'Unemployment_Rate',
    'Estimated Employed': 'Employed',
    'Estimated Labour Participation Rate (%)': 'Labour_Participation_Rate'
}, inplace=True)
```

```
# Get summary statistics
print(data.describe())
```

```
         Estimated Unemployment Rate (%)   Estimated Employed  \
count                        267.000000         2.670000e+02
mean                          12.236929         1.396211e+07
std                           10.803283         1.336632e+07
min                            0.500000         1.175420e+05
25%                            4.845000         2.838930e+06
50%                            9.650000         9.732417e+06
75%                           16.755000         2.187869e+07
max                           75.850000         5.943376e+07

         Estimated Labour Participation Rate (%)   longitude    latitude
count                                 267.000000  267.000000  267.000000
mean                                   41.681573   22.826048   80.532425
std                                     7.845419    6.270731    5.831738
min                                    16.770000   10.850500   71.192400
25%                                    37.265000   18.112400   76.085600
50%                                    40.390000   23.610200   79.019300
75%                                    44.055000   27.278400   85.279900
max                                    69.690000   33.778200   92.937600
```

```
print(data.columns)
```

```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Region.1', 'longitude', 'latitude'],
      dtype='object')
```

```
# Strip leading/trailing spaces from column names
data.columns = data.columns.str.strip()
```

```
# Verify cleaned column names
print(data.columns)
```

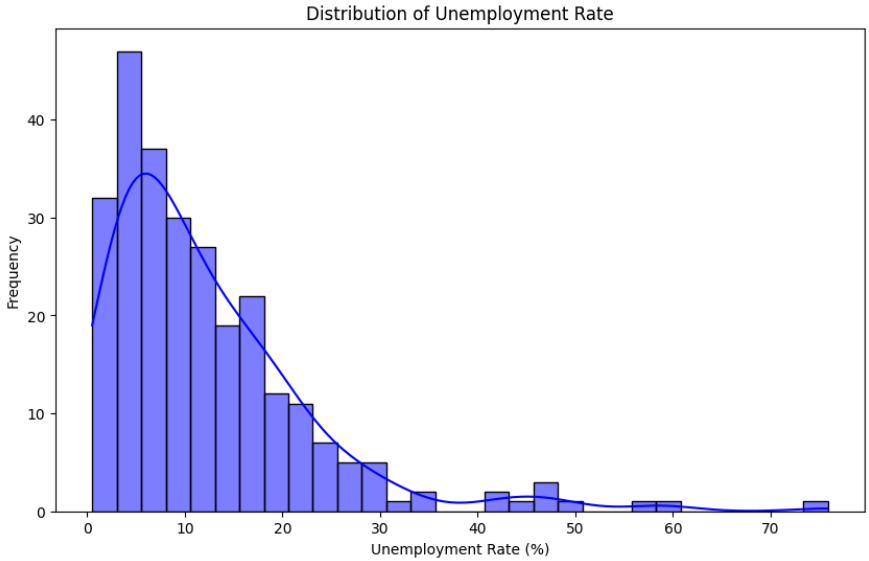
```
Index(['Region', 'Date', 'Frequency', 'Estimated Unemployment Rate (%)',
       'Estimated Employed', 'Estimated Labour Participation Rate (%)',
       'Region.1', 'longitude', 'latitude'],
      dtype='object')
```

```
# Rename columns for easier access
data.rename(columns={
    'Estimated Unemployment Rate (%)': 'Unemployment_Rate',
    'Estimated Employed': 'Employed',
    'Estimated Labour Participation Rate (%)': 'Labour_Participation_Rate'
}, inplace=True)

# Verify renamed columns
print(data.columns)
```

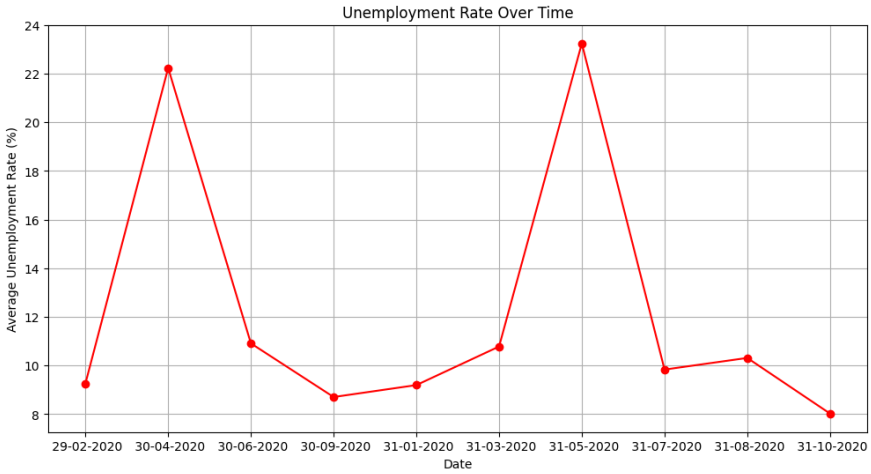
```
Index(['Region', 'Date', 'Frequency', 'Unemployment_Rate', 'Employed',
       'Labour_Participation_Rate', 'Region.1', 'longitude', 'latitude'],
      dtype='object')
```

```
# Plot the distribution of unemployment rate
plt.figure(figsize=(10, 6))
sns.histplot(data['Unemployment_Rate'], kde=True, bins=30, color='blue')
plt.title('Distribution of Unemployment Rate')
plt.xlabel('Unemployment Rate (%)')
plt.ylabel('Frequency')
plt.show()
```




```
# Group data by date and calculate average unemployment rate
unemployment_over_time = data.groupby('Date')['Unemployment_Rate'].mean().reset_index(

# Plot unemployment rate over time
plt.figure(figsize=(12, 6))
plt.plot(unemployment_over_time['Date'], unemployment_over_time['Unemployment_Rate'], r
plt.title('Unemployment Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Average Unemployment Rate (%)')
plt.grid()
plt.show()
```



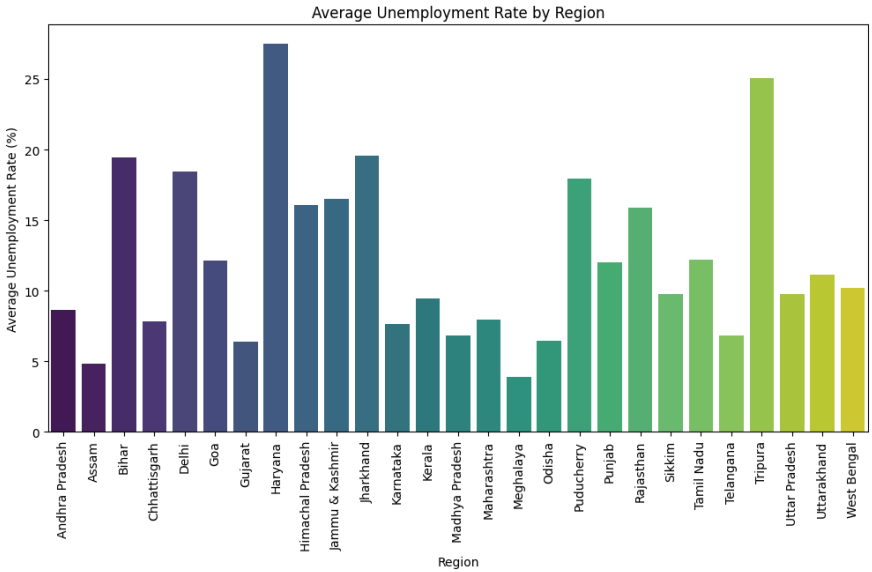

```
# Group data by region and calculate average unemployment rate
regional_unemployment = data.groupby('Region')['Unemployment_Rate'].mean().reset_index

# Plot regional unemployment rates
plt.figure(figsize=(12, 6))
sns.barplot(x='Region', y='Unemployment_Rate', data=regional_unemployment, palette='vi
plt.xticks(rotation=90)
plt.title('Average Unemployment Rate by Region')
plt.xlabel('Region')
plt.ylabel('Average Unemployment Rate (%)')
plt.show()
```

```
<ipython-input-28-555c71225fa3>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.:

  sns.barplot(x='Region', y='Unemployment_Rate', data=regional_unemployment, palet
```




```
# Use Plotly for an interactive map
fig = px.scatter_geo(data,
                     lat='latitude',
                     lon='longitude',
                     color='Unemployment_Rate',
                     size='Unemployment_Rate',
                     hover_name='Region',
                     animation_frame='Date',
                     title='Unemployment Rate Across Regions Over Time',
                     scope='asia',
                     center={'lat': 20.5937, 'lon': 78.9629},
                     height=600)
fig.show()
```


```
# Calculate correlation matrix
correlation_matrix = data[['Unemployment_Rate', 'Employed', 'Labour_Participation_Rate

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```