```
# Step 1: Install & Import
!pip install pandas matplotlib seaborn --quiet

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Unemploy…nt in India.csv

Unemployment in India.csv(text/csv) - 47043 bytes, last modified: 11/13/2025 - 100% done

```
Saving Unemployment in India.csv to Unemployment in India (1).csv
```

```
# Load the dataset
df = pd.read_csv("Unemployment in India (1).csv")

# Display first 5 rows
df.head()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139.0	43.24	Rural
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881.0	42.05	Rural
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707.0	43.50	Rural

Next steps: Generate code with df New interactive sheet

```
# Check basic info
df.info()

# Check for missing values
print("\nMissing values in each column:\n", df.isnull().sum())

# Check basic statistics
print("\nDataset Statistics:\n", df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   Region                                   740 non-null    object
 1    Date                                    740 non-null    object
 2    Frequency                               740 non-null    object
 3    Estimated Unemployment Rate (%)         740 non-null    float64
 4    Estimated Employed                      740 non-null    float64
 5    Estimated Labour Participation Rate (%)  740 non-null    float64
 6   Area                                     740 non-null    object
dtypes: float64(3), object(4)
memory usage: 42.1+ KB

Missing values in each column:
 Region                                    28
 Date                                     28
 Frequency                                28
 Estimated Unemployment Rate (%)          28
 Estimated Employed                       28
 Estimated Labour Participation Rate (%)  28
Area                                      28
dtype: int64

Dataset Statistics:
        Estimated Unemployment Rate (%)  Estimated Employed  \
count                       740.000000        7.400000e+02
```

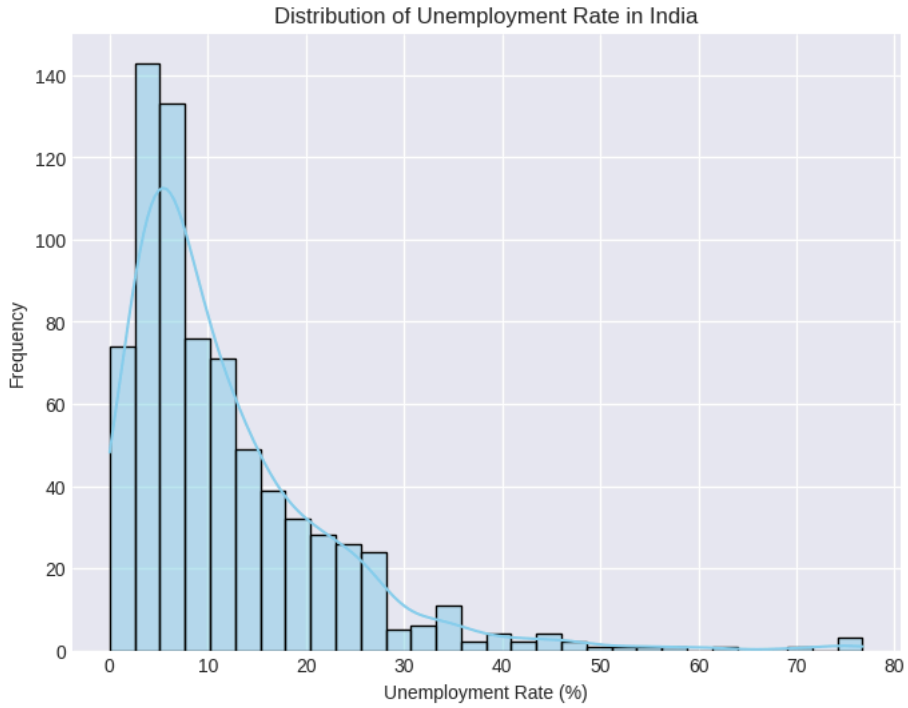
```
mean                          11.787946        7.204460e+06
std                           10.721298        8.087988e+06
min                            0.000000        4.942000e+04
25%                            4.657500        1.190404e+06
50%                            8.350000        4.744178e+06
75%                           15.887500        1.127549e+07
max                           76.740000        4.577751e+07

       Estimated Labour Participation Rate (%)
count                              740.000000
mean                                42.630122
std                                  8.111094
min                                 13.330000
25%                                 38.062500
50%                                 41.160000
75%                                 45.505000
max                                 72.570000
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style
plt.style.use('seaborn-v0_8-darkgrid')

# 1️⃣ Overall Unemployment Rate Distribution
plt.figure(figsize=(8,6))
sns.histplot(df['Estimated Unemployment Rate (%)'], bins=30, color='skyblue', kde=True)
plt.title("Distribution of Unemployment Rate in India")
plt.xlabel("Unemployment Rate (%)")
plt.ylabel("Frequency")
plt.show()

# 2️⃣ State-wise Average Unemployment Rate
state_unemp = df.groupby('Region')['Estimated Unemployment Rate (%)'].mean().sort_values(ascending=False)

plt.figure(figsize=(10,8))
sns.barplot(x=state_unemp.values, y=state_unemp.index, palette='coolwarm')
plt.title("Average Unemployment Rate by State")
plt.xlabel("Average Unemployment Rate (%)")
plt.ylabel("State/Region")
plt.show()
```

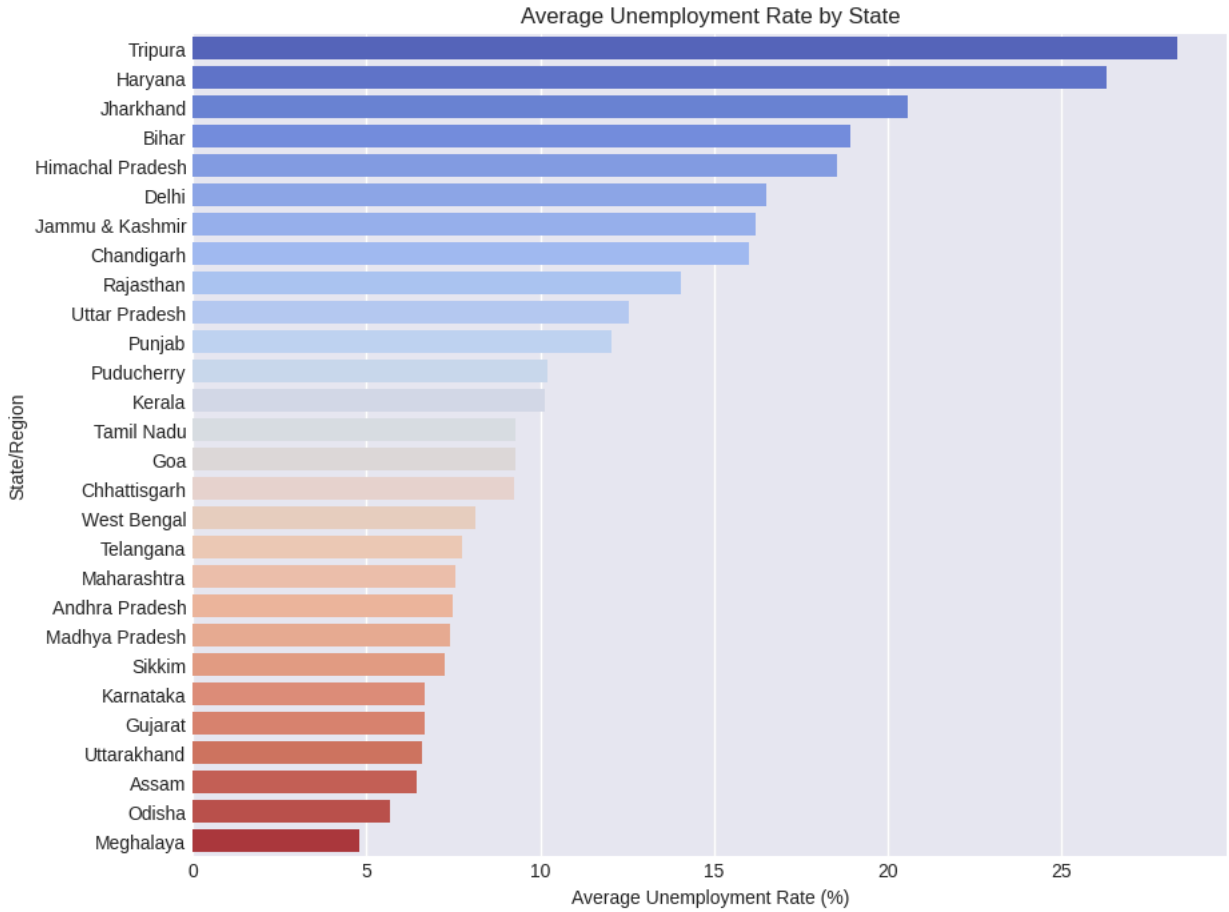
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Estimated Unemployment Rate (%)'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                                          2 frames
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'Estimated Unemployment Rate (%)'

<Figure size 800x600 with 0 Axes>
```

Next steps: Explain error

```
# 🔹 Step 1: Clean column names (remove spaces)
df.columns = df.columns.str.strip()

# 🔹 Step 2: Import visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('seaborn-v0_8-darkgrid')

# 🔹 Step 3: Plot 1 – Distribution of Unemployment Rate
plt.figure(figsize=(8,6))
sns.histplot(df['Estimated Unemployment Rate (%)'], bins=30, color='skyblue', kde=True)
plt.title("Distribution of Unemployment Rate in India")
plt.xlabel("Unemployment Rate (%)")
plt.ylabel("Frequency")
plt.show()

# 🔹 Step 4: Plot 2 – Average Unemployment Rate by State
state_unemp = df.groupby('Region')['Estimated Unemployment Rate (%)'].mean().sort_values(ascending=False)

plt.figure(figsize=(10,8))
sns.barplot(x=state_unemp.values, y=state_unemp.index, palette='coolwarm')
plt.title("Average Unemployment Rate by State")
plt.xlabel("Average Unemployment Rate (%)")
plt.ylabel("State/Region")
plt.show()
```

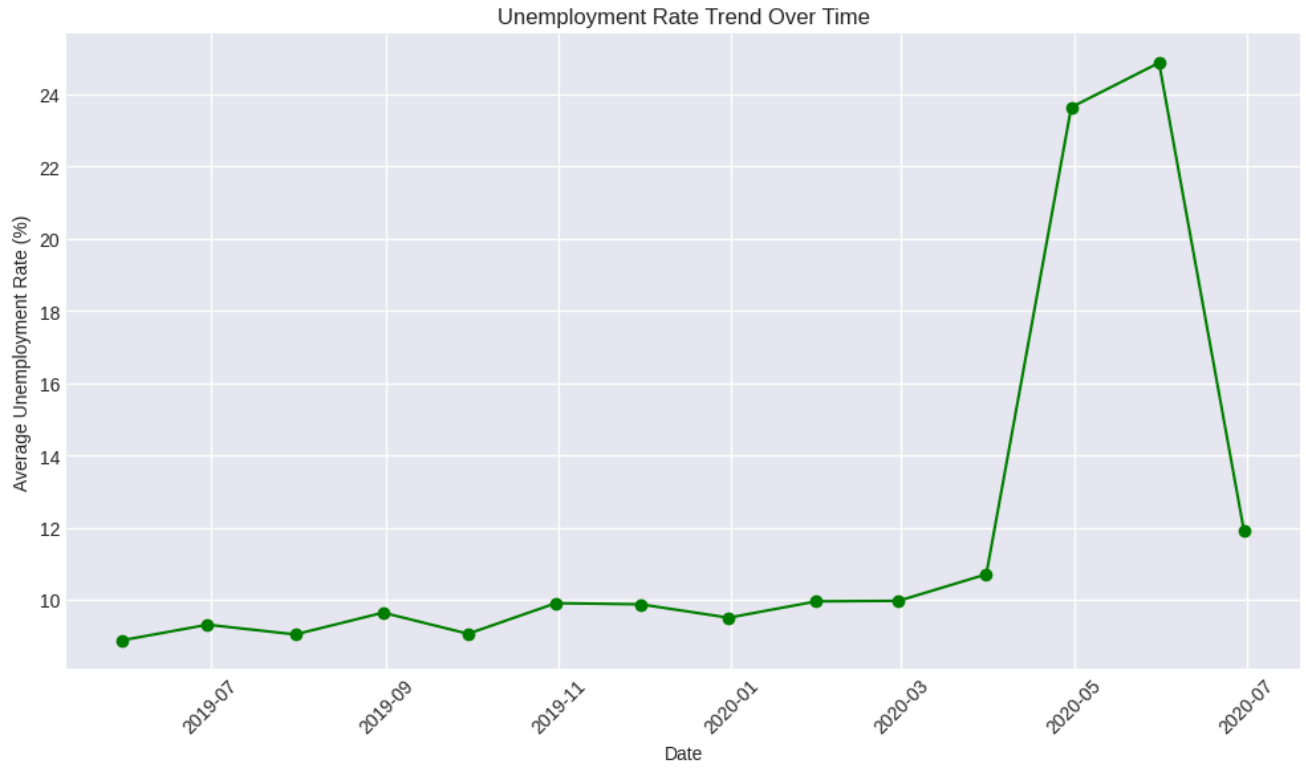



```
/tmp/ipython-input-974622293.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `

  sns.barplot(x=state_unemp.values, y=state_unemp.index, palette='coolwarm')
```




```
# Step 6: Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True, errors='coerce')

# Step 7: Group by date and calculate mean unemployment rate
```

```
trend = df.groupby('Date')['Estimated Unemployment Rate (%)'].mean()

# Step 8: Plot the trend over time
plt.figure(figsize=(10,6))
plt.plot(trend.index, trend.values, color='green', marker='o', linestyle='-')
plt.title("Unemployment Rate Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Average Unemployment Rate (%)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Step 7: Compare Urban vs Rural Unemployment Rate
area_unemp = df.groupby('Area')['Estimated Unemployment Rate (%)'].mean()

plt.figure(figsize=(7,5))
sns.barplot(x=area_unemp.index, y=area_unemp.values, palette='viridis')
plt.title("Urban vs Rural - Average Unemployment Rate")
plt.xlabel("Area Type")
plt.ylabel("Average Unemployment Rate (%)")
plt.show()
```