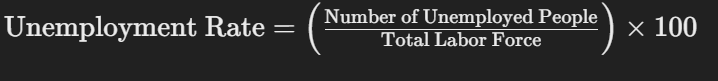
# Unemployment Analysis with Python

Objective  
The objective of this project is to analyse the unemployment rate in India using data science techniques. The unemployment rate is a critical economic indicator that reflects the number of people who are unemployed as a percentage of the total labour force. Analysing the unemployment rate, especially during significant events like the COVID-19 pandemic, can provide valuable insights into the economic impact and help in policy-making decisions.  
  
Solution  
To achieve the objective, the solution involves the following steps:  
  
**Data Collection**: Gather datasets containing unemployment rates and related economic indicators.  
**Data Preprocessing**: Clean and preprocess the data to ensure consistency and accuracy.  
**Data Analysis**: Perform exploratory data analysis to understand trends and patterns.  
**Data Visualization**: Visualize the data using various plots and charts to gain insights.  
**Correlation Analysis**: Analyze the correlation between different economic indicators.  
**Conclusion**: Summarize the findings and provide insights based on the analysis.

Procedure  
**Step 1: Data Collection**The data used in this project includes two CSV files:

* Unemployment\_Rate\_upto\_11\_2020.csv
* Unemployment in India.csv

**Step 2: Data Preprocessing  
Load the Data**: Use pandas to load the CSV files into DataFrames.  
**Inspect the Data**: Display the first few rows and check the structure of the data.  
**Check for Missing Values**: Identify any missing values in the datasets.  
**Clean the Data**: Strip leading/trailing spaces from column names and date values.  
**Convert Data Types**: Convert date columns to datetime format and ensure consistency in column names and data types.  
**Step 3: Data Analysis  
Descriptive Statistics**: Compute descriptive statistics for the unemployment rates.  
**Trend Analysis**: Analyze the trend of unemployment rates over time using line plots.  
**Regional Analysis**: Analyze regional unemployment rates using bar plots.  
**Correlation Analysis**: Analyze the correlation between different economic indicators using heatmaps.  
**Step 4: Data Visualization**  
Visualize the data using seaborn and matplotlib to create line plots, bar plots, and heatmaps.  
  
**Step 5: Conclusion**  
Summarize the findings from the analysis and provide insights into the trends and patterns observed in the unemployment rates.  
  
Theory and Algorithms  
**Unemployment Rate**  
The unemployment rate is calculated as:



**Exploratory Data Analysis (EDA)**EDA involves summarizing the main characteristics of the data, often with visual methods. It helps in understanding the distribution, identifying patterns, and detecting anomalies.  
  
**Data Visualization**  
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.  
  
**Correlation Analysis**Correlation analysis is used to evaluate the strength and direction of the linear relationship between two quantitative variables. The correlation coefficient ranges from -1 to 1, where:

* 1 indicates a strong positive relationship,
* -1 indicates a strong negative relationship,
* 0 indicates no relationship.

Code Execution  
Here is the complete code used for the analysis:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the CSV files into pandas DataFrames

file1 = 'Unemployment\_Rate\_upto\_11\_2020.csv'

file2 = 'Unemployment in India.csv'

data1 = pd.read\_csv(file1)

data2 = pd.read\_csv(file2)

# Display the first few rows of each DataFrame to understand their structure

print("Data from Unemployment\_Rate\_upto\_11\_2020.csv:")

print(data1.head())

print("\nData from Unemployment in India.csv:")

print(data2.head())

# Print the column names of each DataFrame

print("\nColumn names in data1:")

print(data1.columns)

print("\nColumn names in data2:")

print(data2.columns)

# Data Cleaning and Preprocessing

# Check for missing values

print("\nMissing values in data1:")

print(data1.isnull().sum())

print("\nMissing values in data2:")

print(data2.isnull().sum())

# Strip leading/trailing spaces from column names and date values

data1.columns = data1.columns.str.strip()

data2.columns = data2.columns.str.strip()

data1['Date'] = data1['Date'].str.strip()

data2['Date'] = data2['Date'].str.strip()

# Convert date columns to datetime format

data1['Date'] = pd.to\_datetime(data1['Date'], format='%d-%m-%Y')

data2['Date'] = pd.to\_datetime(data2['Date'], format='%d-%m-%Y')

# Ensure consistency in column names and data types

data1.columns = data1.columns.str.replace(' ', '\_')

data2.columns = data2.columns.str.replace(' ', '\_')

# Descriptive statistics of the unemployment rates

print("\nDescriptive statistics for data1:")

print(data1['Estimated\_Unemployment\_Rate\_(%)'].describe())

print("\nDescriptive statistics for data2:")

print(data2['Estimated\_Unemployment\_Rate\_(%)'].describe())

# Trend analysis over time

plt.figure(figsize=(12, 6))

sns.lineplot(data=data1, x='Date', y='Estimated\_Unemployment\_Rate\_(%)', label='Data1')

sns.lineplot(data=data2, x='Date', y='Estimated\_Unemployment\_Rate\_(%)', label='Data2')

plt.title('Unemployment Rate Over Time')

plt.xlabel('Date')

plt.ylabel('Estimated Unemployment Rate (%)')

plt.legend()

plt.show()

# Regional analysis of unemployment rates

plt.figure(figsize=(12, 6))

sns.barplot(data=data1, x='Region', y='Estimated\_Unemployment\_Rate\_(%)')

plt.title('Regional Unemployment Rates (Data1)')

plt.xlabel('Region')

plt.ylabel('Estimated Unemployment Rate (%)')

plt.xticks(rotation=90)

plt.show()

plt.figure(figsize=(12, 6))

sns.barplot(data=data2, x='Region', y='Estimated\_Unemployment\_Rate\_(%)')

plt.title('Regional Unemployment Rates (Data2)')

plt.xlabel('Region')

plt.ylabel('Estimated Unemployment Rate (%)')

plt.xticks(rotation=90)

plt.show()

# Heatmap for correlation analysis

plt.figure(figsize=(10, 8))

sns.heatmap(data1.select\_dtypes(include='number').corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap (Data1)')

plt.show()

plt.figure(figsize=(10, 8))

sns.heatmap(data2.select\_dtypes(include='number').corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap (Data2)')

plt.show()

Here is the complete output used for the analysis:

Data from Unemployment\_Rate\_upto\_11\_2020.csv:

Region Date Frequency Estimated Unemployment Rate (%) \

0 Andhra Pradesh 31-01-2020 M 5.48

1 Andhra Pradesh 29-02-2020 M 5.83

2 Andhra Pradesh 31-03-2020 M 5.79

3 Andhra Pradesh 30-04-2020 M 20.51

4 Andhra Pradesh 31-05-2020 M 17.43

Estimated Employed Estimated Labour Participation Rate (%) Region.1 \

0 16635535 41.02 South

1 16545652 40.90 South

2 15881197 39.18 South

3 11336911 33.10 South

4 12988845 36.46 South

longitude latitude

0 15.9129 79.74

1 15.9129 79.74

2 15.9129 79.74

3 15.9129 79.74

4 15.9129 79.74

Data from Unemployment in India.csv:

Region Date Frequency Estimated Unemployment Rate (%) \

0 Andhra Pradesh 31-05-2019 Monthly 3.65

1 Andhra Pradesh 30-06-2019 Monthly 3.05

2 Andhra Pradesh 31-07-2019 Monthly 3.75

3 Andhra Pradesh 31-08-2019 Monthly 3.32

4 Andhra Pradesh 30-09-2019 Monthly 5.17

Estimated Employed Estimated Labour Participation Rate (%) Area

0 11999139.0 43.24 Rural

1 11755881.0 42.05 Rural

2 12086707.0 43.50 Rural

3 12285693.0 43.97 Rural

4 12256762.0 44.68 Rural

Column names in data1:

Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',

' Estimated Employed', ' Estimated Labour Participation Rate (%)',

'Region.1', 'longitude', 'latitude'],

dtype='object')

Column names in data2:

Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',

' Estimated Employed', ' Estimated Labour Participation Rate (%)',

'Area'],

dtype='object')

Missing values in data1:

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

Missing values in data2:

Region 28

Date 28

Frequency 28

Estimated Unemployment Rate (%) 28

Estimated Employed 28

Estimated Labour Participation Rate (%) 28

Area 28

dtype: int64

Descriptive statistics for data1:

count 267.000000

mean 12.236929

std 10.803283

min 0.500000

25% 4.845000

50% 9.650000

75% 16.755000

max 75.850000

Name: Estimated\_Unemployment\_Rate\_(%), dtype: float64

Descriptive statistics for data2:

count 740.000000

mean 11.787946

std 10.721298

min 0.000000

25% 4.657500

50% 8.350000

75% 15.887500

max 76.740000

Name: Estimated\_Unemployment\_Rate\_(%), dtype: float64

A graph showing the amount of unemployment rate over time

Description automatically generated

A graph of unemployment rates

Description automatically generated

A graph of blue columns with white text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

## Conclusion

In this project, we analyzed the unemployment rate in India using data from two datasets. We performed data cleaning, preprocessing, and visualization to understand the trends and patterns in the unemployment rate over time and across regions. The analysis revealed significant insights into the impact of events like the COVID-19 pandemic on unemployment. The correlation analysis provided additional insights into the relationships between various economic indicators. This analysis can help policymakers and economists make informed decisions to address unemployment issues.