

Unemployment In India

Import Libraries

```
In [402]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Dataset

```
In [403]: Unemployment = pd.read_csv('Unemployment in India.csv')
Unemployment.head(10)
```

Out[403]:

	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
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693.0	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762.0	44.68	Rural
5	Andhra Pradesh	31-10-2019	Monthly	3.52	12017412.0	43.01	Rural
6	Andhra Pradesh	30-11-2019	Monthly	4.12	11397681.0	41.00	Rural
7	Andhra Pradesh	31-12-2019	Monthly	4.38	12528395.0	45.14	Rural
8	Andhra Pradesh	31-01-2020	Monthly	4.84	12016676.0	43.46	Rural
9	Andhra Pradesh	29-02-2020	Monthly	5.91	11723617.0	42.83	Rural

In [404]: `Unemployment.head(10)`

Out[404]:

	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
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693.0	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762.0	44.68	Rural
5	Andhra Pradesh	31-10-2019	Monthly	3.52	12017412.0	43.01	Rural
6	Andhra Pradesh	30-11-2019	Monthly	4.12	11397681.0	41.00	Rural
7	Andhra Pradesh	31-12-2019	Monthly	4.38	12528395.0	45.14	Rural
8	Andhra Pradesh	31-01-2020	Monthly	4.84	12016676.0	43.46	Rural
9	Andhra Pradesh	29-02-2020	Monthly	5.91	11723617.0	42.83	Rural

Data Overview

In [405]: `Unemployment.info()`

```
<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
```

In [406]: `Unemployment.describe()`

Out[406]:

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
count	740.000000	7.400000e+02	740.000000
mean	11.787946	7.204460e+06	42.630122
std	10.721298	8.087988e+06	8.111094
min	0.000000	4.942000e+04	13.330000
25%	4.657500	1.190404e+06	38.062500
50%	8.350000	4.744178e+06	41.160000
75%	15.887500	1.127549e+07	45.505000
max	76.740000	4.577751e+07	72.570000

In [407]: `Unemployment.describe(include = 'object')`

Out[407]:

	Region	Date	Frequency	Area
count	740	740	740	740
unique	28	14	2	2
top	Andhra Pradesh	31-10-2019	Monthly	Urban
freq	28	55	381	381

In [408]: `Unemployment.shape`

Out[408]: (768, 7)

In [409]:

```
# Number of rows & columns
print(f'The dataset has {Unemployment.shape[0]} rows')
print(f'The dataset has {Unemployment.shape[1]} columns')
```

The dataset has 768 rows
The dataset has 7 columns

Checking for missing values

In [410]: `Unemployment.isnull().sum()`

Out[410]:

Region	28
Date	28
Frequency	28
Estimated Unemployment Rate (%)	28
Estimated Employed	28
Estimated Labour Participation Rate (%)	28
Area	28
dtype: int64	

```
In [411]: Unemployment = Unemployment.dropna()
```

```
In [412]: Unemployment.isnull().sum()
```

```
Out[412]: Region      0
          Date        0
          Frequency    0
          Estimated Unemployment Rate (%)  0
          Estimated Employed      0
          Estimated Labour Participation Rate (%)  0
          Area      0
          dtype: int64
```

Ecxploratory Data Analysis

Updating & Rearranging Columns

```
In [413]: # Check the actual column names in your DataFrame
          Unemployment.columns
```

```
Out[413]: Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
                ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
                'Area'],
                dtype='object')
```

```
In [414]: Unemployment.columns = Unemployment.columns.str.strip()
          Unemployment.columns
```

```
Out[414]: Index(['Region', 'Date', 'Frequency', 'Estimated Unemployment Rate (%)',
                'Estimated Employed', 'Estimated Labour Participation Rate (%)',
                'Area'],
                dtype='object')
```

```
In [415]: # Rename columns for easier reference
          Unemployment.rename(columns={
                'Estimated Unemployment Rate (%)': 'Unemployment_Rate',
                'Estimated Employed': 'Employed',
                'Estimated Labour Participation Rate (%)': 'Labour_Participation_Rate'}, i
```

```
In [416]: # Display the first few rows after preprocessing
Unemployment.head()
```

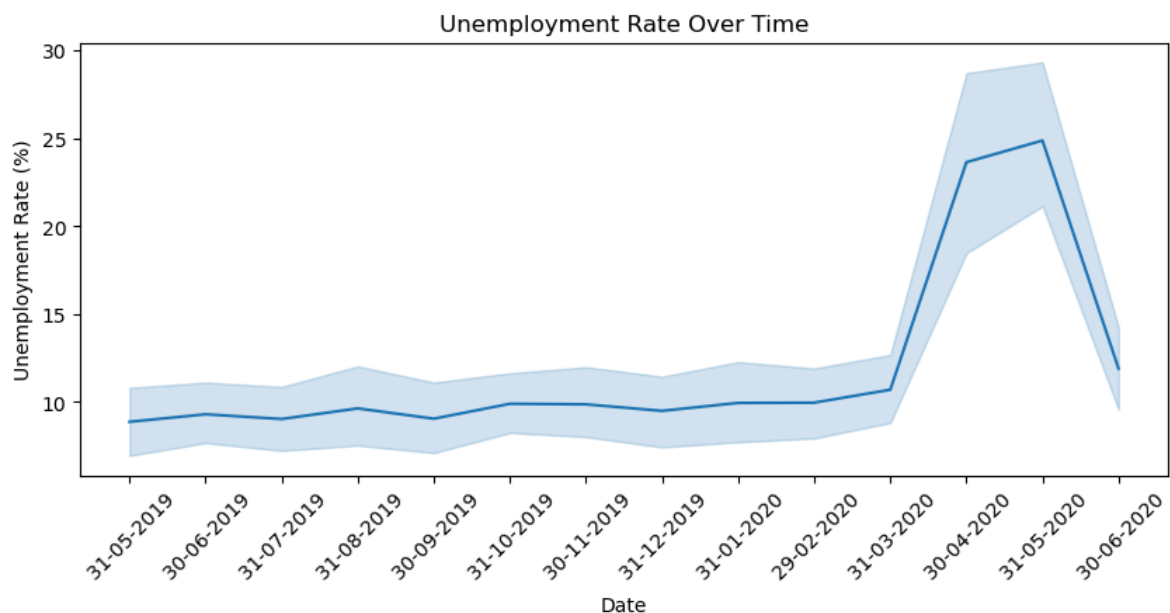
```
Out[416]:
```

	Region	Date	Frequency	Unemployment_Rate	Employed	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
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693.0	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762.0	44.68	Rural

```
In [417]: # Visualize unemployment rate over time
plt.figure(figsize=(10, 4))

sns.lineplot(data=Unemployment, x='Date', y='Unemployment_Rate')

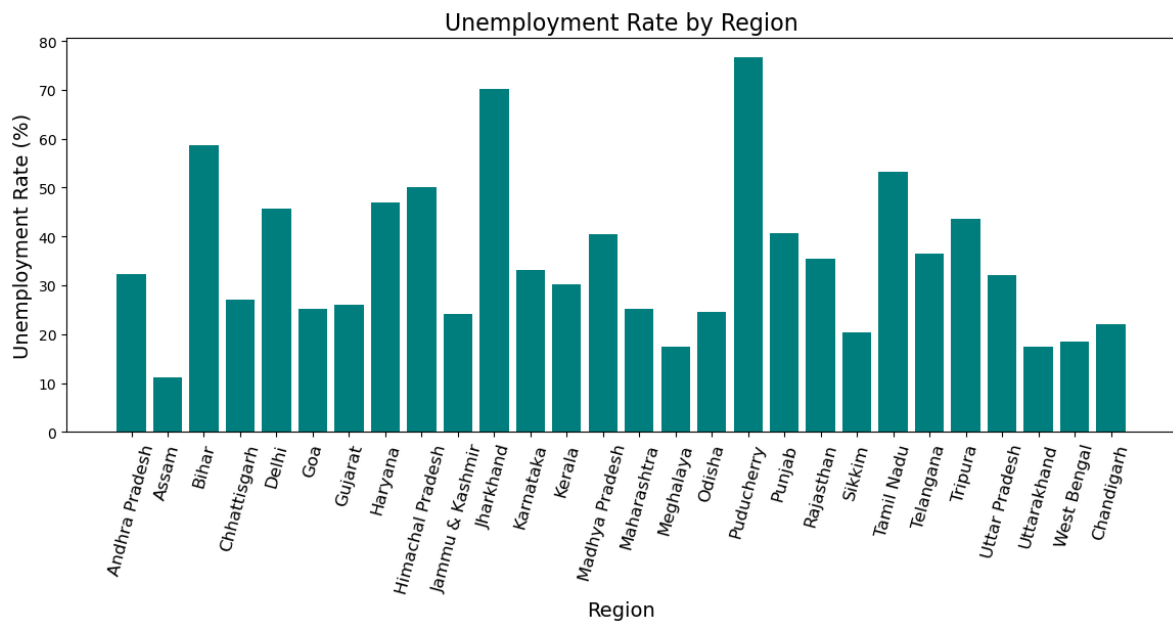
plt.title('Unemployment Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Unemployment Rate (%)')
plt.xticks(rotation=45)
plt.show()
```



```
In [418]: Unemployment [ 'Unemployment_Rate' ]
```

```
Out[418]: 0      3.65
          1      3.05
          2      3.75
          3      3.32
          4      5.17
          ...
          749    7.55
          750    6.67
          751    15.63
          752    15.22
          753     9.86
          Name: Unemployment_Rate, Length: 740, dtype: float64
```

```
In [419]: plt.figure(figsize=(14, 5))
          plt.bar(Unemployment[ 'Region' ], Unemployment[ 'Unemployment_Rate' ], color='teal')
          plt.title('Unemployment Rate by Region', fontsize=16)
          plt.xlabel('Region', fontsize=14)
          plt.ylabel('Unemployment Rate (%)', fontsize=14)
          plt.xticks(rotation=75, fontsize=12)
          plt.show()
```

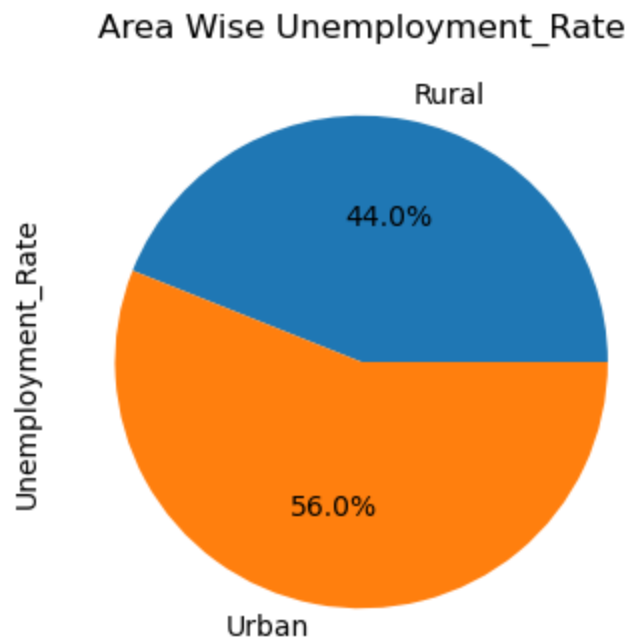


Area wise Unemployment_Rate

```
In [420]: Unemployment [ 'Area' ].value_counts()
```

```
Out[420]: Urban      381
          Rural      359
          Name: Area, dtype: int64
```

```
In [421]: # area wise unemployment rate using pie chart
plt.figure(figsize=(10,4))
Unemployment.groupby('Area')['Unemployment_Rate'].mean().plot(kind='pie', auto)
plt.title('Area Wise Unemployment_Rate')
plt.show()
```



Observation

- Its clearly visible that Urban Area has more (56.0%) Unemployment_Rate than Rural Area (44.0%)

Min, Max, Avg Unemployment Rate

```
In [422]: avg_unemployment_rate = Unemployment.groupby('Region')['Unemployment_Rate'].mean()
state_with_highest_unemployment = avg_unemployment_rate.idxmax()
high_unemployment_rate = avg_unemployment_rate.max()
state_with_lowest_unemployment = avg_unemployment_rate.idxmin()
low_unemployment_rate = avg_unemployment_rate.min()
print(f"State with Highest Unemployment : {state_with_highest_unemployment}")
print(f"Highest Unemployment Rate : {high_unemployment_rate}")
print(f"State with Lowest Unemployment : {state_with_lowest_unemployment}")
print(f"Low Employment Rate : {low_unemployment_rate}")
```

```
State with Highest Unemployment : Tripura
Highest Unemployment Rate : 28.350357142857142
State with Lowest Unemployment : Meghalaya
Low Employment Rate : 4.7988888888888885
```

```
In [423]: avg_unemployment_rate
```

```
Out[423]: Region
Andhra Pradesh      7.477143
Assam               6.428077
Bihar              18.918214
Chandigarh         15.991667
Chhattisgarh       9.240357
Delhi              16.495357
Goa                9.274167
Gujarat            6.663929
Haryana            26.283214
Himachal Pradesh   18.540357
Jammu & Kashmir     16.188571
Jharkhand           20.585000
Karnataka           6.676071
Kerala             10.123929
Madhya Pradesh     7.406429
Maharashtra        7.557500
Meghalaya          4.798889
Odisha             5.657857
Puducherry         10.215000
Punjab             12.031071
Rajasthan          14.058214
Sikkim             7.249412
Tamil Nadu         9.284286
Telangana          7.737857
Tripura            28.350357
Uttar Pradesh      12.551429
Uttarakhand        6.582963
West Bengal        8.124643
Name: Unemployment_Rate, dtype: float64
```

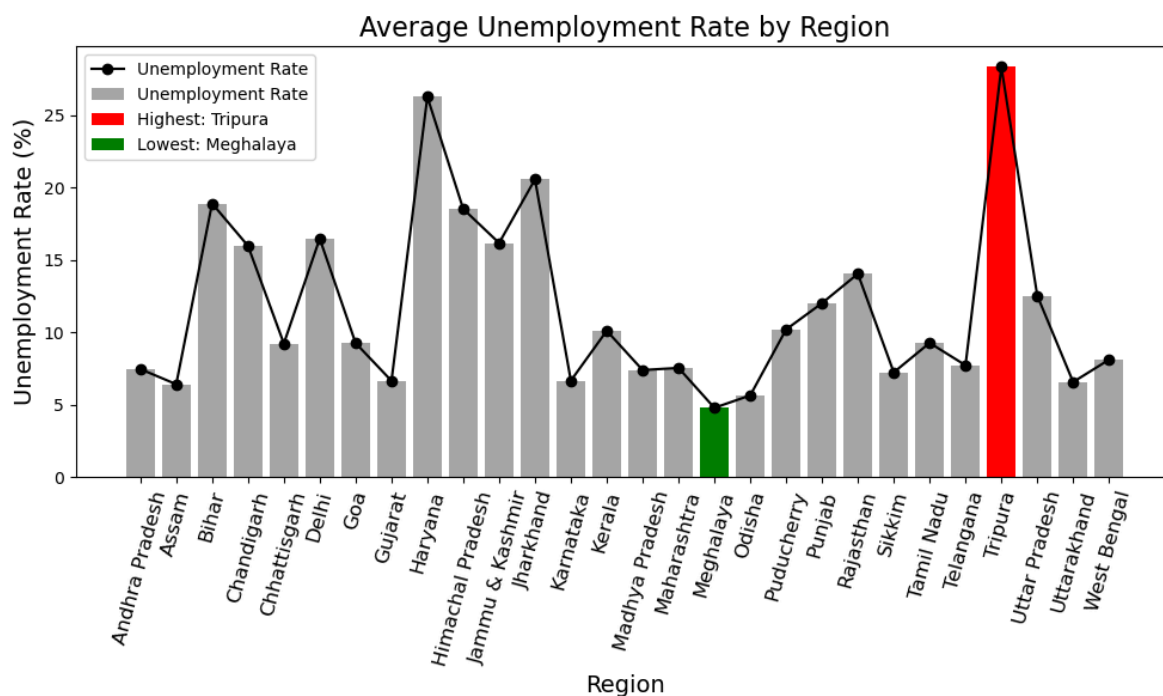

Average Unemployment Rate by Region

```
In [424]: # Create the bar plot
plt.figure(figsize=(10, 6))
plt.plot(avg_unemployment_rate.index, avg_unemployment_rate.values, marker='o')
plt.bar(avg_unemployment_rate.index, avg_unemployment_rate.values, label='Unem

# Highlight the highest and lowest unemployment rate regions
plt.bar(state_with_highest_unemployment, high_unemployment_rate, color='red',
plt.bar(state_with_lowest_unemployment, low_unemployment_rate, color='green',

# Add Labels and title
plt.title('Average Unemployment Rate by Region', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Unemployment Rate (%)', fontsize=14)
plt.xticks(rotation=75, fontsize=12)
plt.legend()

# Show the plot
plt.tight_layout()
plt.show()
```

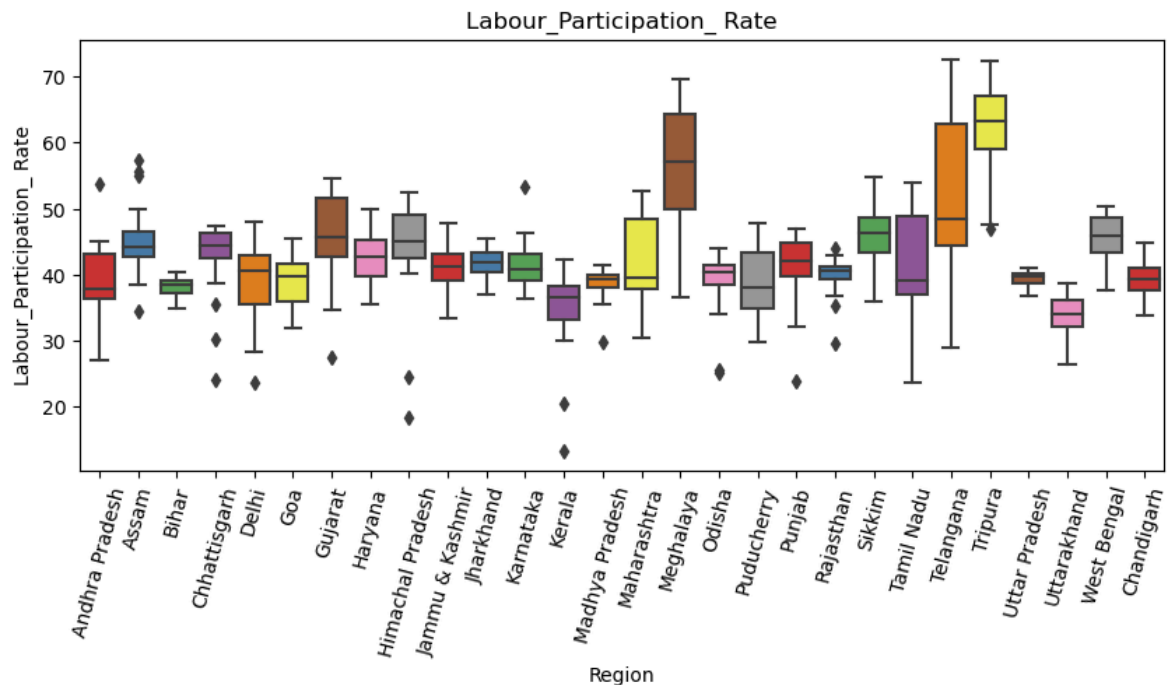


Observation

- Red bar indicates Highest Avg Employment Rate in "Tripura"
- Green bar indicates Lowest Avg Employment Rate in "Meghalaya"
- Grey/black plots indicate the unemployment by region

Labour_Participation_Rate by Region

```
In [433]: plt.figure(figsize=(10, 4))
sns.boxplot(x='Region', y='Labour_Participation_Rate', data = Unemployment ,pa
plt.title('Labour_Participation_Rate')
plt.xlabel('Region')
plt.ylabel('Labour_Participation_Rate')
plt.xticks(rotation = 75)
plt.show()
```

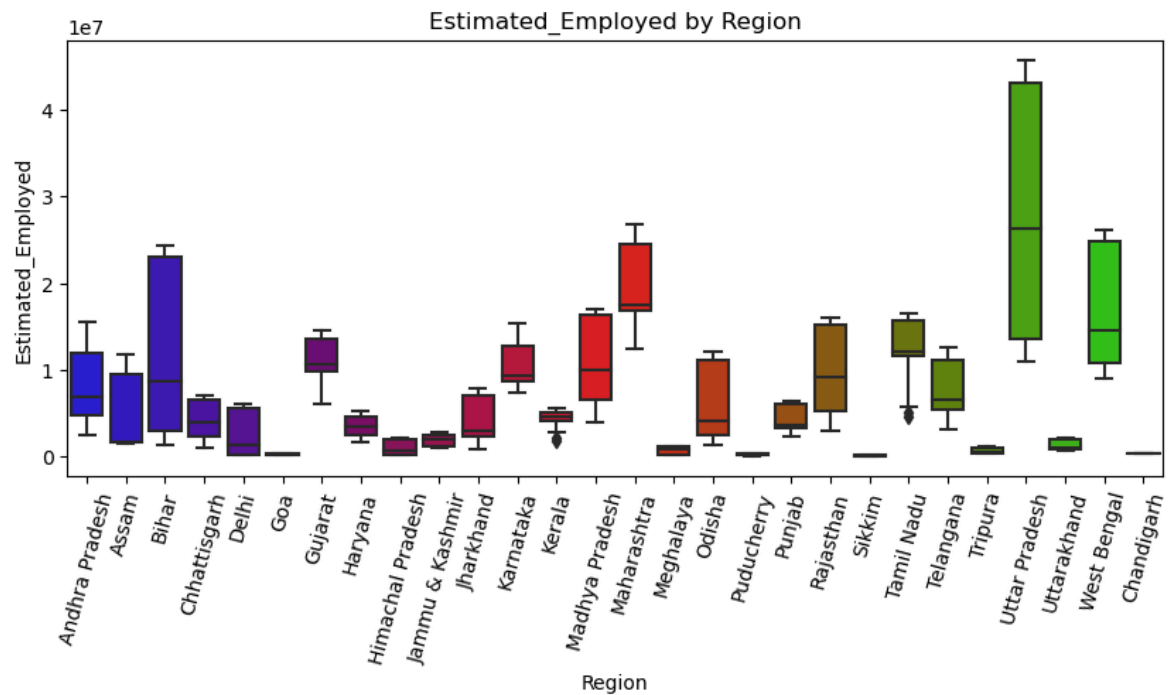


Observation

- This visualization shows the Labour Participation Rate by region
- Through this we have seen the Lowest & Highest Labour Participation Rate by region

Estimated_Employed by Region

```
In [430]: plt.figure(figsize=(10, 4))
sns.boxplot(x='Region', y='Employed', data = Unemployment, palette = 'brg')
plt.title('Estimated_Employed by Region')
plt.xlabel('Region')
plt.ylabel('Estimated_Employed')
plt.xticks(rotation = 75)
plt.show()
```



Observation

- The Estimated Employed by Region is shown in this plot

```
In [428]: sns.pairplot(Unemployment)
```

```
Out[428]: <seaborn.axisgrid.PairGrid at 0x1aa3b6c7fd0>
```

