

# Unemployment Analysis Using Python



## Importing Necessary Libraries

```
In [2]: 1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 import plotly.express as px
5 import matplotlib.pyplot as plt
6
7 %matplotlib inline
```

## Setting Graph Style

```
In [3]: 1 sns.set_style('darkgrid')
2 plt.rcParams['font.size'] = 10
3 plt.rcParams['figure.figsize'] = (8, 4)
4 # plt.rcParams['figure.facecolor'] = '#00000000'
```

## Loading Dataset from csv file

```
In [4]: 1 df = pd.read_csv("Unemployment_Rate_upto_11_2020.csv")
2 df.head()
```

```
Out[4]:
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.74
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.74
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.74
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.74

## Rename the columns

```
In [5]: 1 new_column_names = {
2     'Region': "State",
3     'Date': "Date",
4     'Frequency': "Frequency",
5     'Estimated Unemployment Rate (%)': "Unemployment_Rate",
6     'Estimated Employed': "Estimated_Employed",
7     'Estimated Labour Participation Rate (%)': "Labour_Participation_Rate",
8     'Region.1': "Area"
9 }
10 df.rename(columns=new_column_names, inplace=True)
11
12 df.head()
```

```
Out[5]:
```

	State	Date	Frequency	Unemployment_Rate	Estimated_Employed	Labour_Participation_Rate	Area	longitude	latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.74
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.74
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.74
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.74

## Creating month Column from Date

```
In [6]: 1 df['month'] = pd.DatetimeIndex(df['Date']).month
        2 df.head()
```

```
Out[6]:
```

	State	Date	Frequency	Unemployment_Rate	Estimated_Employed	Labour_Participation_Rate	Area	longitude	latitude	month
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74	1
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.74	2
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.74	3
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.74	4
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.74	5

```
In [7]: 1 df.isnull().sum()
```

```
Out[7]: State          0
Date          0
Frequency     0
Unemployment_Rate  0
Estimated_Employed  0
Labour_Participation_Rate  0
Area          0
longitude     0
latitude      0
month         0
dtype: int64
```

```
In [8]: 1 # Handling missing values: fill NaN with mean for numeric columns
        2 numeric_columns = df.select_dtypes(include=[float, int]).columns
        3 df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())
```

```
In [9]: 1 # Handling missing values: fill NaN with mode for categorical columns
        2 categorical_columns = df.select_dtypes(include=[object]).columns
        3 df[categorical_columns] = df[categorical_columns].fillna(df[categorical_columns].mode().iloc[0])
```

```
In [10]: 1 df.isnull().sum()
```

```
Out[10]: State          0
Date          0
Frequency     0
Unemployment_Rate  0
Estimated_Employed  0
Labour_Participation_Rate  0
Area          0
longitude     0
latitude      0
month         0
dtype: int64
```

#### Summary statistics of the dataset

```
In [11]: 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 267 entries, 0 to 266
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   State                 267 non-null   object
1   Date                 267 non-null   object
2   Frequency            267 non-null   object
3   Unemployment_Rate    267 non-null   float64
4   Estimated_Employed   267 non-null   int64
5   Labour_Participation_Rate  267 non-null   float64
6   Area                 267 non-null   object
7   longitude            267 non-null   float64
8   latitude             267 non-null   float64
9   month               267 non-null   int64
dtypes: float64(4), int64(2), object(4)
memory usage: 21.0+ KB
```

```
In [12]: 1 df.describe()
```

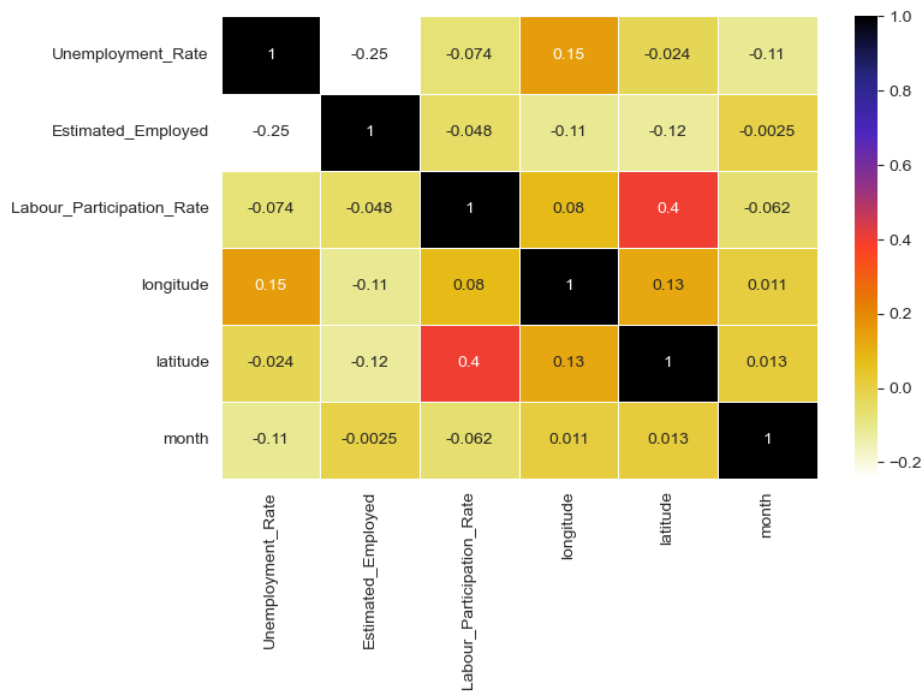
```
Out[12]:
```

	Unemployment_Rate	Estimated_Employed	Labour_Participation_Rate	longitude	latitude	month
count	267.000000	2.670000e+02	267.000000	267.000000	267.000000	267.000000
mean	12.236929	1.396211e+07	41.681573	22.826048	80.532425	5.535581
std	10.803283	1.336632e+07	7.845419	6.270731	5.831738	2.870915
min	0.500000	1.175420e+05	16.770000	10.850500	71.192400	1.000000
25%	4.845000	2.838930e+06	37.265000	18.112400	76.085600	3.000000
50%	9.650000	9.732417e+06	40.390000	23.610200	79.019300	6.000000
75%	16.755000	2.187869e+07	44.055000	27.278400	85.279900	8.000000
max	75.850000	5.943376e+07	69.690000	33.778200	92.937600	10.000000

#### Correlation between the Features of the Dataset

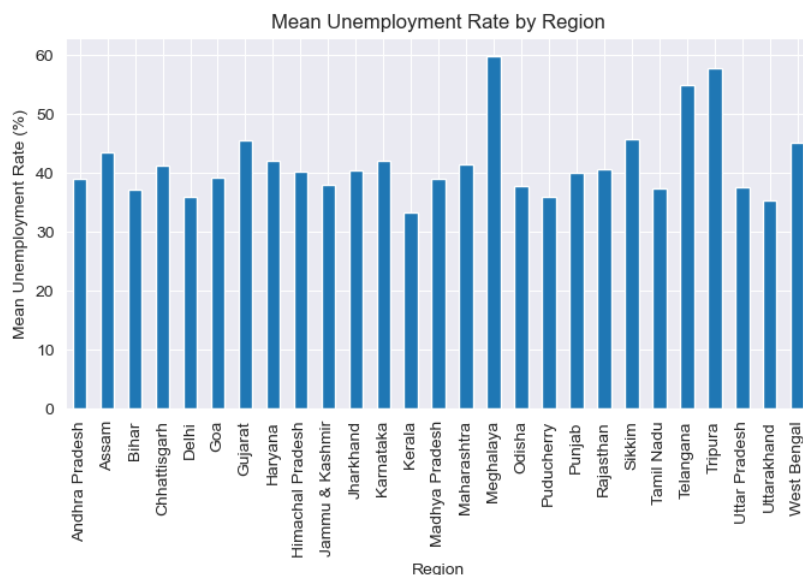
```
In [13]: 1 plt.figure(figsize=(8, 5))
2 sns.heatmap(df.corr(), cmap="CMRmap_r", annot=True, lw=0.5)
```

Out[13]: <AxesSubplot:>



### Average Estimated Unemployment Rate in Each State

```
In [14]: 1 # Grouping the data by Region and calculating mean unemployment rate
2 mean_unemployment_by_region = df.groupby("State")["Labour_Participation_Rate"].mean()
3 pd.DataFrame(mean_unemployment_by_region)
4
5 # Plotting mean unemployment rate by region
6 mean_unemployment_by_region.plot(kind="bar")
7
8 plt.title("Mean Unemployment Rate by Region")
9 plt.xlabel("Region")
10 plt.ylabel("Mean Unemployment Rate (%)")
11 plt.show()
```



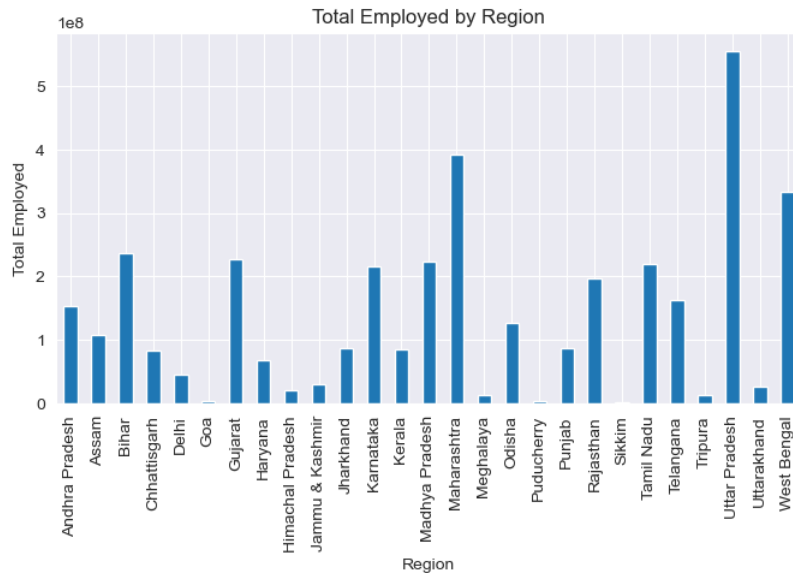
```
In [15]: 1 l = pd.DataFrame(mean_unemployment_by_region)
2 l = l.sort_values(by=["Labour_Participation_Rate"])
3 l = l.reset_index()
4 pd.DataFrame( {"State": [l.min()[0],l.max()[0]], "Mean Unemployment Rate": [l.min()[1],l.max()[1]] }, index=["Min", "Max"])
```

Out[15]:

	State	Mean Unemployment Rate
Min	Andhra Pradesh	33.382
Max	West Bengal	59.859

## Total Employed in Each State

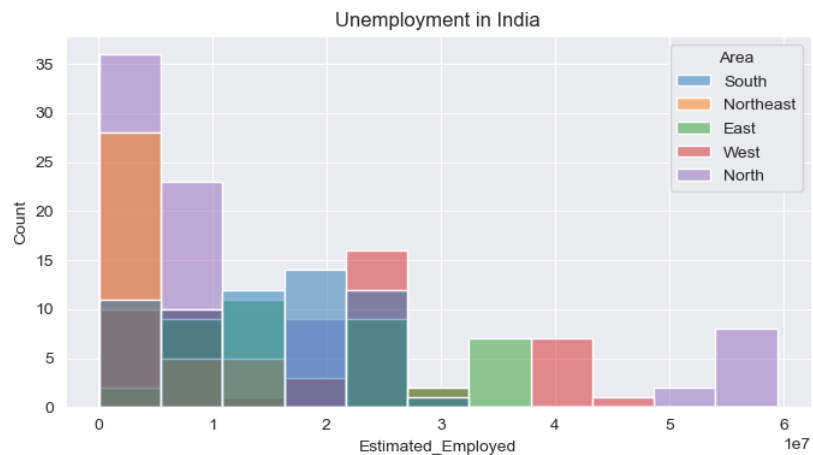
```
In [16]: 1 # Grouping the data by Region and calculating total employed
2 total_employed_by_region = df.groupby("State")["Estimated_Employed"].sum()
3 pd.DataFrame(total_employed_by_region).reset_index()
4
5 # Plotting total employed by region
6 total_employed_by_region.plot(kind="bar")
7 plt.title("Total Employed by Region")
8 plt.xlabel("Region")
9 plt.ylabel("Total Employed")
10 plt.show()
```



## Estimated Employed in Each Regions

```
In [17]: 1 plt.title("Unemployment in India")
2 sns.histplot(x="Estimated_Employed", hue="Area", data=df)
```

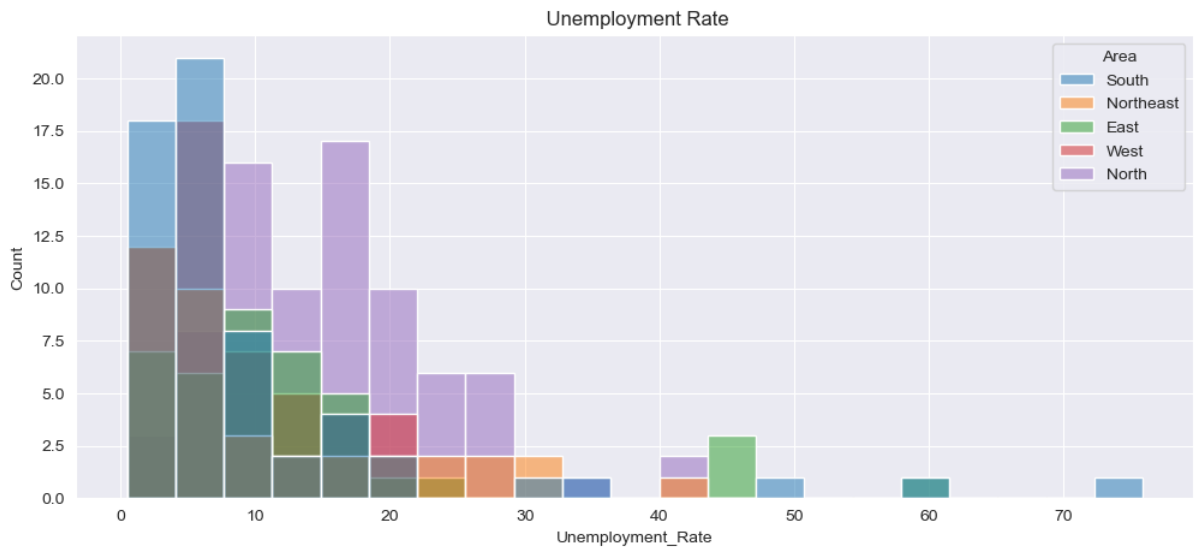
```
Out[17]: <AxesSubplot:title={'center':'Unemployment in India'}, xlabel='Estimated_Employed', ylabel='Count'>
```



## Unemployment Rate according to the different Region

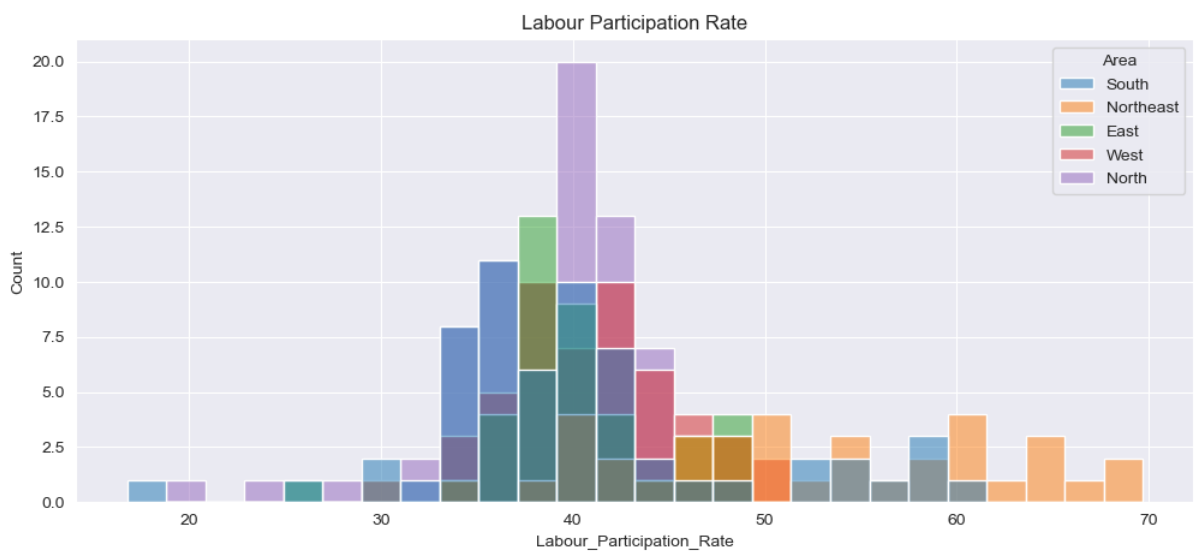
```
In [18]: 1 plt.figure(figsize=(12, 5))
2         plt.title(" Unemployment Rate")
3         sns.histplot(x="Unemployment_Rate", hue="Area", data=df)
```

```
Out[18]: <AxesSubplot:title={'center': ' Unemployment Rate'}, xlabel='Unemployment_Rate', ylabel='Count'>
```



```
In [19]: 1 plt.figure(figsize=(12, 5))
2         plt.title("Labour Participation Rate")
3         sns.histplot(x="Labour_Participation_Rate", hue="Area", data=df)
```

```
Out[19]: <AxesSubplot:title={'center': 'Labour Participation Rate'}, xlabel='Labour_Participation_Rate', ylabel='Count'>
```



## Sunburst to Analysis the Estimated Unemployment Rate

```
In [20]: 1 data = df[["State", "Area", "Unemployment_Rate"]]
2
3 fig = px.sunburst(data,
4                   path=["Area", "State"],
5                   values="Unemployment_Rate",
6                   width=600, height=600,
7                   title="Unemployment Rate in India"
8                   ).show()
```

Unemployment Rate in India



```
In [21]: 1 #mean_unemployment_by_region = df.groupby("State")["Labour_Participation_Rate"].mean()
2
3 monthly_unemployment_rate = df.groupby("month")["Unemployment_Rate"].mean()
4 monthly_unemployment_rate = pd.DataFrame(monthly_unemployment_rate).reset_index()
5
6 monthly_unemployment_rate
```

```
Out[21]:
```

	month	Unemployment_Rate
0	1	9.196538
1	2	9.266154
2	3	10.782593
3	4	22.236154
4	5	23.244444
5	6	10.911111
6	7	9.834444
7	8	10.313333
8	9	8.705926
9	10	8.026296

```
In [22]: 1 sns.barplot(data=monthly_unemployment_rate, x='month', y="Unemployment_Rate")
```

```
Out[22]: <AxesSubplot: xlabel='month', ylabel='Unemployment_Rate'>
```

