Oasis InfoByte Task-2

UNEMPLOYMENT ANALYSIS WITH PYTHON

Topic : UNEMPLOYMENT ANALYSIS WITH PYTHON *Unemployment is measured by the unemployment rate which is the number of people who are unemployed as a percentage of the total labour force. We have seen a sharp increase in the unemployment rate during Covid-19, so analyzing the unemployment rate can be a good data science project.*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
## Supress warnings
import warnings
warnings.filterwarnings("ignore")
```

```
data = pd.read_csv("/unemployment.csv")
print("data has been successfully loaded")
```

```
    data has been successfully loaded
```

```
data
```

	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	79.740
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.740
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.740
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.740
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.740
...	...	...	...	...	...	...	...	...	...
262	West Bengal	30-06-2020	M	7.29	30726310	40.39	East	22.9868	87.855
263	West Bengal	31-07-2020	M	6.83	35372506	46.17	East	22.9868	87.855
264	West Bengal	31-08-2020	M	14.87	33298644	47.48	East	22.9868	87.855
265	West Bengal	30-09-2020	M	9.35	35707239	47.73	East	22.9868	87.855
266	West Bengal	31-10-2020	M	9.98	33962549	45.63	East	22.9868	87.855

267 rows × 9 columns

Data Analysis

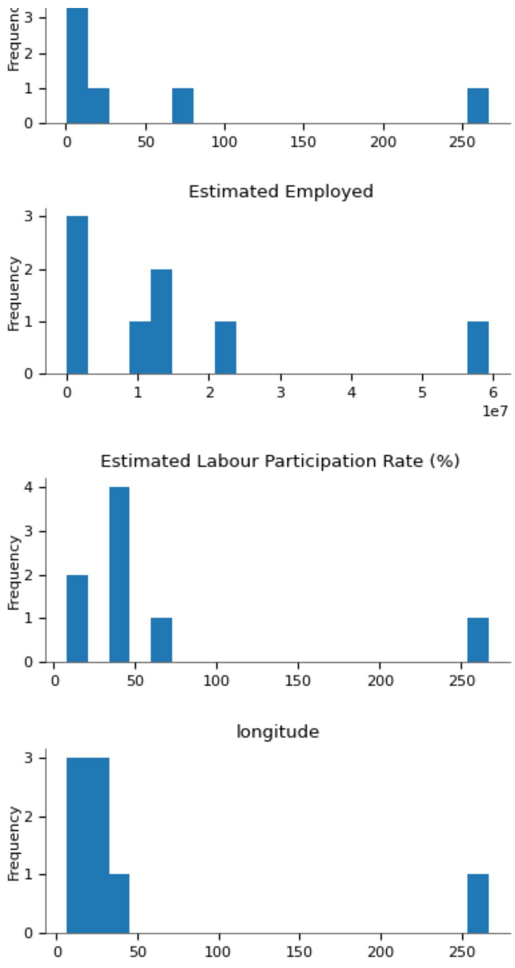
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 267 entries, 0 to 266
Data columns (total 9 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   Region                                    267 non-null    object 
 1    Date                                     267 non-null    object 
 2    Frequency                                267 non-null    object 
 3    Estimated Unemployment Rate (%)          267 non-null    float64
 4    Estimated Employed                       267 non-null    int64  
 5    Estimated Labour Participation Rate (%)  267 non-null    float64
 6   Region.1                                  267 non-null    object 
 7   longitude                                 267 non-null    float64
 8   latitude                                  267 non-null    float64
dtypes: float64(4), int64(1), object(4)
memory usage: 18.9+ KB
```

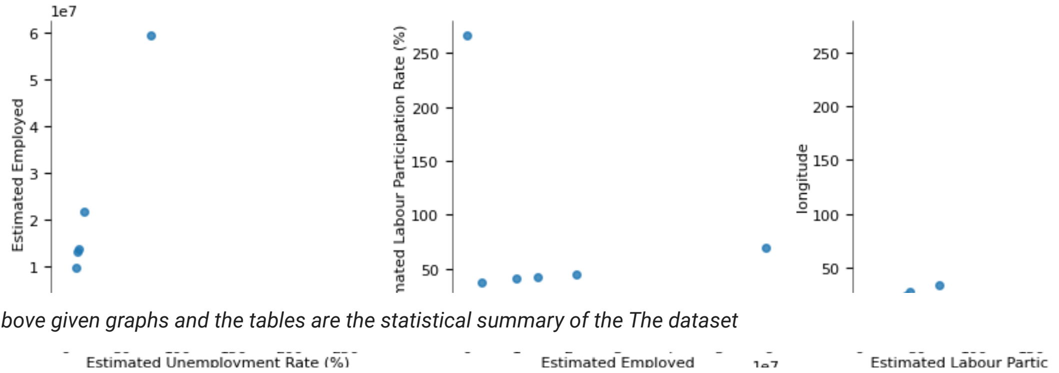
```
data.shape
```

```
(267, 9)
```

```
data.describe()
```



2-d distributions



The above given graphs and the tables are the statistical summary of the The dataset

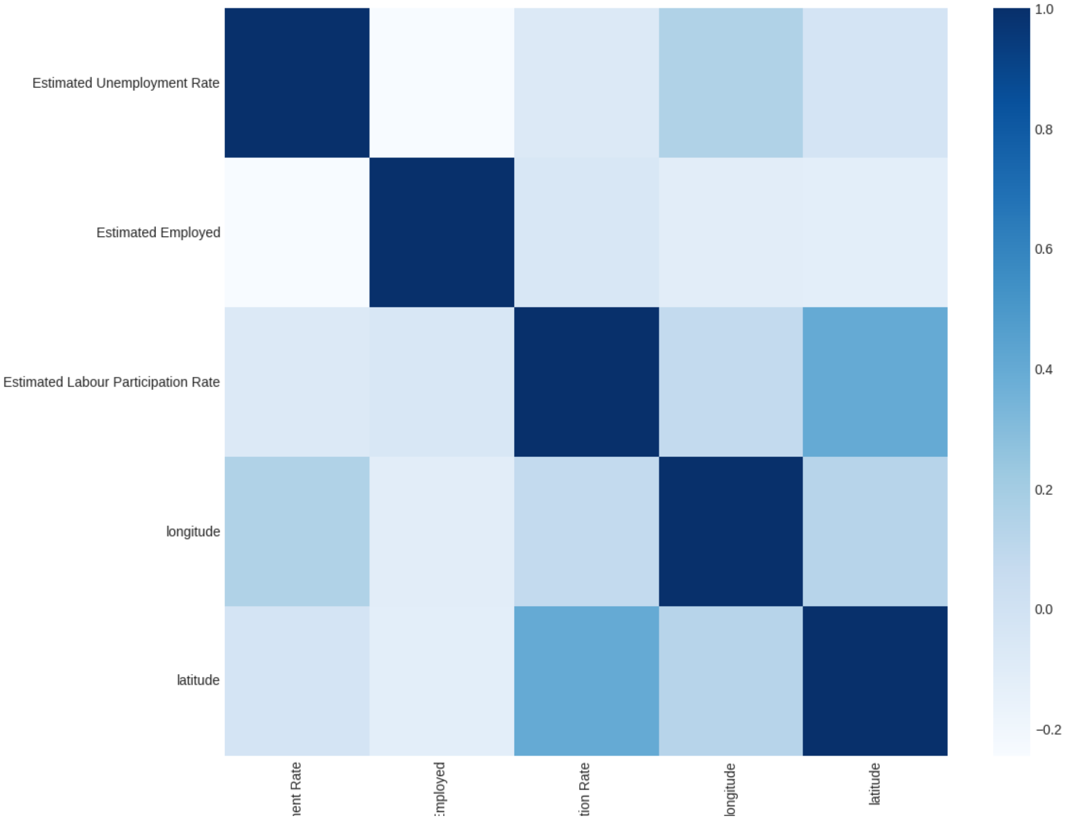
Let’s see if this dataset contains missing values or not:

```
print(data.isnull().sum())
```

```
Region                        0
Estimated Unemployment Rate   0
dtype: int64
```

Here ,I discovered that the column names are incorrect after looking into the missing values. In order to make this data easier to grasp, so have renamed all the columns as you can see above.

```
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr(), cmap='Blues')
plt.show()
```

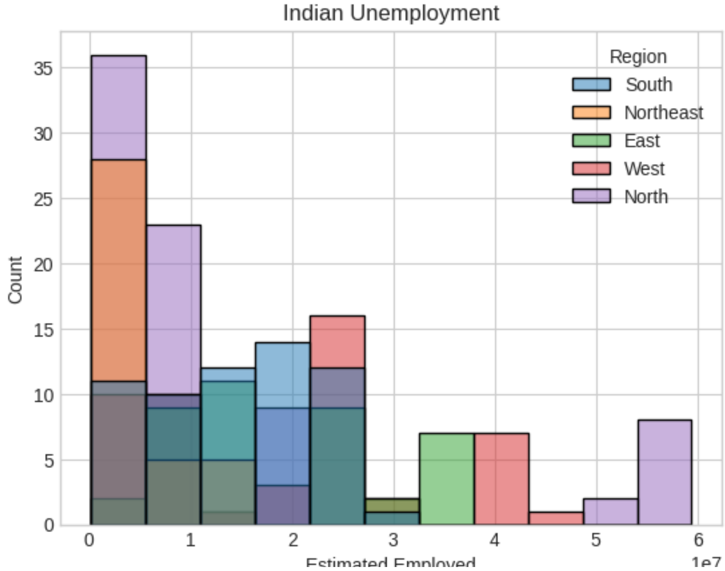


the correlation coefficient between two variables is 0.8, this means that there is a strong positive correlation between the two variables. If the correlation coefficient between two variables is -0.5, this means that there is a moderate negative correlation between the two variables.

Data visulization

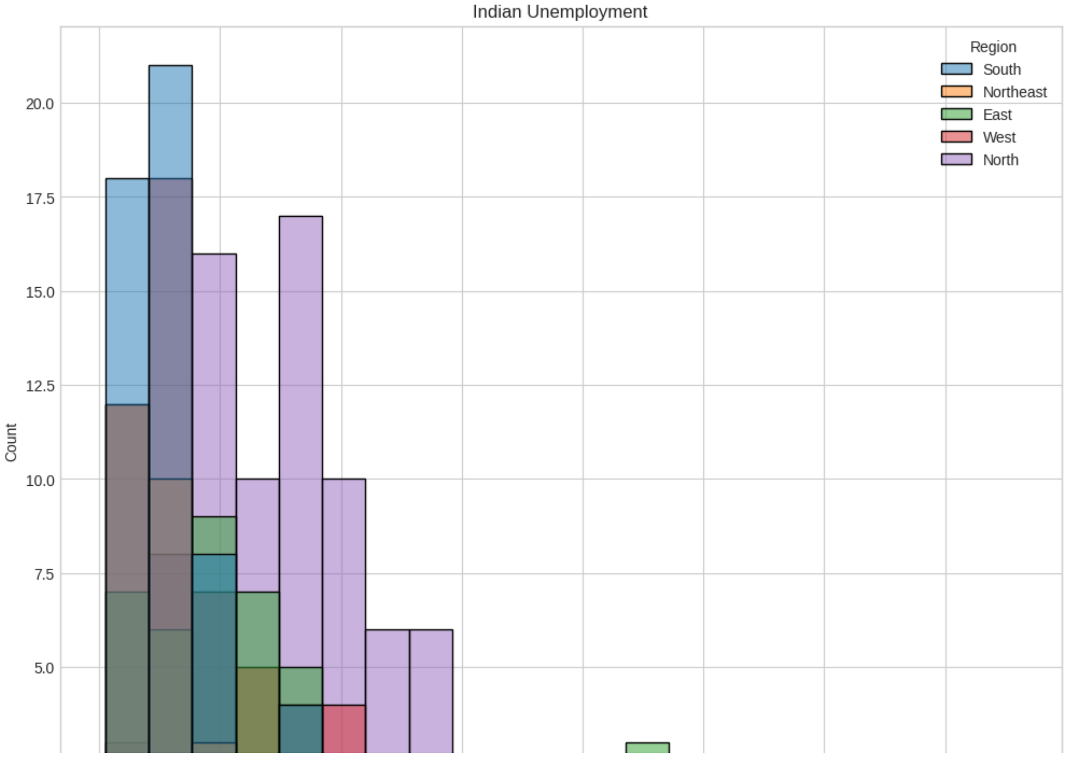
The estimated number of employees according to different regions of India:

```
data.columns= ["States","Date","Frequency",
 "Estimated Unemployment Rate","Estimated Employed",
 "Estimated Labour Participation Rate","Region",
 "longitude","latitude"]
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Employed", hue="Region", data=data)
plt.show()
```



the unemployment rate according to different regions of India

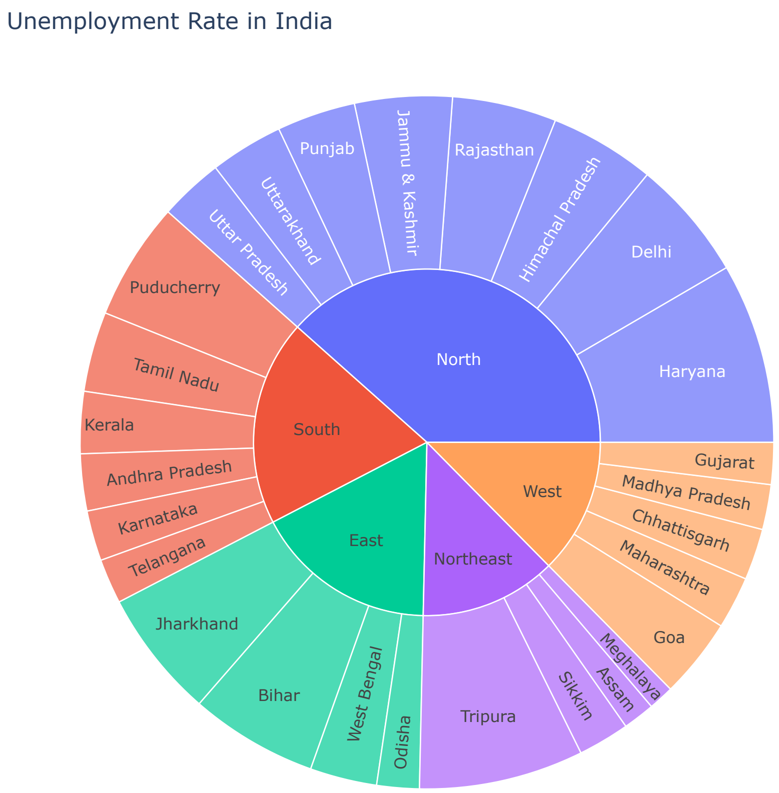
```
plt.figure(figsize=(12, 10))
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Unemployment Rate", hue="Region", data=data)
plt.show()
```



Double-click (or enter) to edit

A dashboard to analyze the unemployment rate of each Indian state by region

```
unemploment = data[["States", "Region", "Estimated Unemployment Rate"]]
figure = px.sunburst(unemploment, path=["Region", "States"],
values="Estimated Unemployment Rate",
width=700, height=700, color_continuous_scale="RdY1Gn",
title="Unemployment Rate in India")
figure.show()
```


Unemployment Rate in India

if you see a large arc for the state of Uttar Pradesh, this means that Uttar Pradesh has a high unemployment rate. If you see a small for the state of Kerala, this means that Kerala has a low unemployment rate.

You can also use the sunburst plot to compare the unemployment rates of different regions. For example, if you see that the northeastern region of India has a larger average arc size than the southern region of India, this means that the northeastern region has a higher average unemployment rate than the southern region.

Overall, the sunburst plot is a useful tool for visualizing and understanding the unemployment rate in India by state

Region possesssing most of data

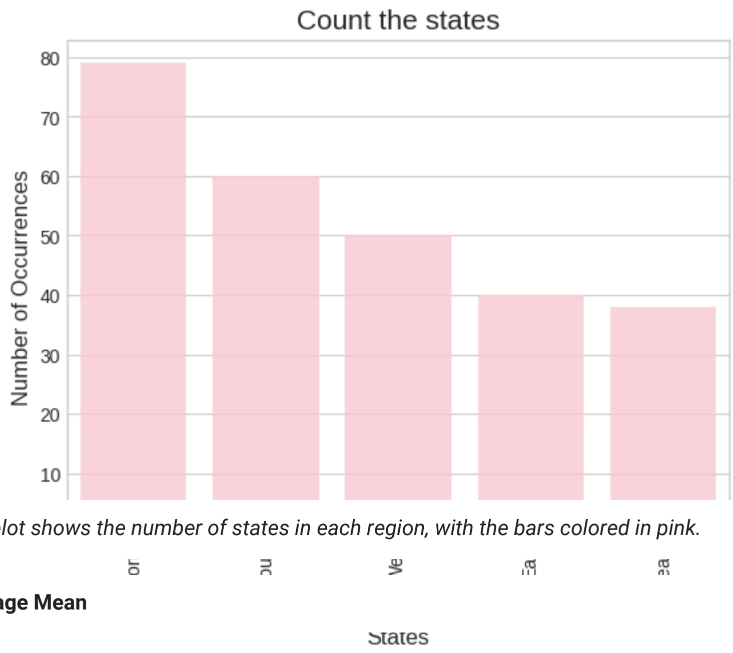
```
# Get the value counts of the Region column
cnt_srs = data.Region.value_counts()

# Create a bar plot
sns.barplot(x=cnt_srs.index, y=cnt_srs.values, alpha=0.8, color='pink')

# Set the axis labels and title
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('States', fontsize=12)
plt.title('Count the states', fontsize=15)

# Rotate the x-axis labels
plt.xticks(rotation='vertical')

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



The plot shows the number of states in each region, with the bars colored in pink.

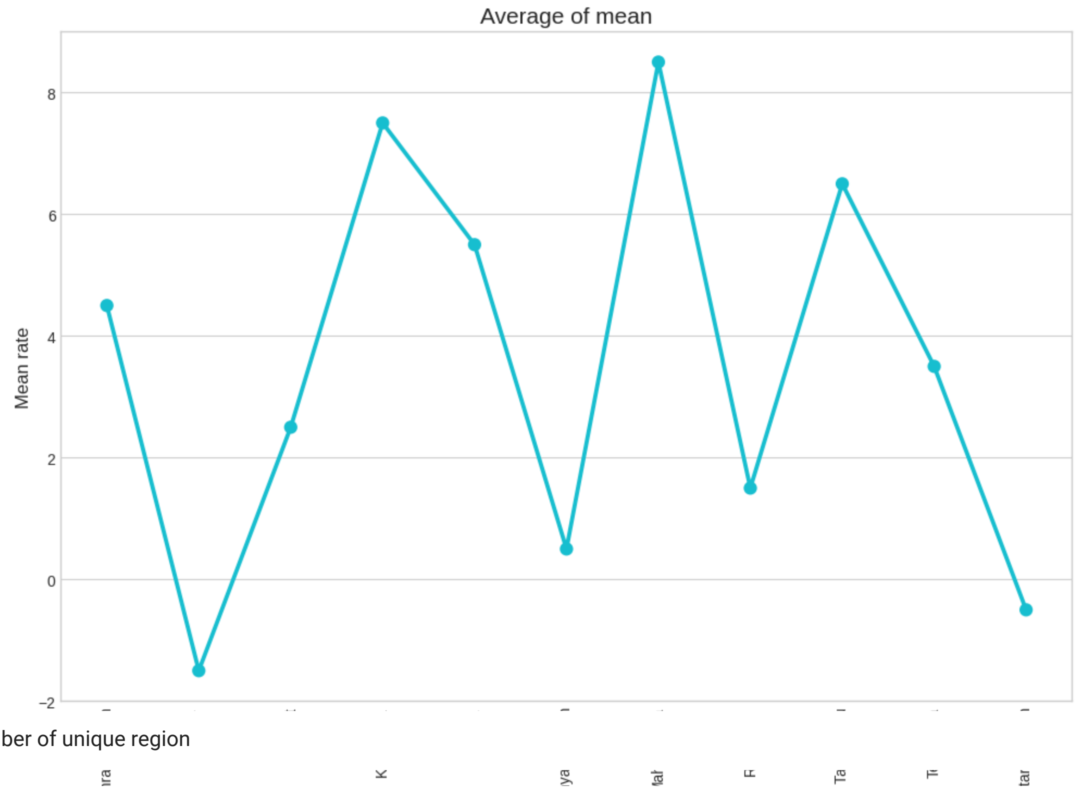
Average Mean

```
# Create a point plot
plt.figure(figsize=(12,8))
sns.pointplot(x=grouped_df['Region'].values, y=grouped_df['Estimated Unemployment Rate'].values, dodge=0.8, color=sns.color_palette()[9])

# Set the axis labels and title
plt.ylabel('Mean rate', fontsize=12)
plt.xlabel('States', fontsize=12)
plt.title("Average of mean", fontsize=15)

# Rotate the x-axis labels
plt.xticks(rotation='vertical')

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



Number of unique region

```
data.Region.nunique()
```

```
    11
```

Exact Numbers

```
make_total = data.pivot_table("Estimated Unemployment Rate",index=['Region'],aggfunc='mean')
topstate=make_total.sort_values(by='Estimated Unemployment Rate',ascending=False)[:47]
print(topstate)
```

```
                    Estimated Unemployment Rate
    Region
    Maharashtra                             8.5
    Karnataka                               7.5
    Tamil Nadu                              6.5
    Kerala                                  5.5
    Andhra Pradesh                          4.5
    Telangana                               3.5
    Gujarat                                 2.5
    Rajasthan                               1.5
    Madhya Pradesh                          0.5
    Uttar Pradesh                          -0.5
    Bihar                                  -1.5
```

Conclusion

Unemployment rates fluctuate from year to year, with all regions experiencing some degree of fluctuation. The East, North, Northeast, South, and West regions experienced the highest yearly fluctuations. This suggests that these regions are more sensitive to economic shocks and other factors that can impact employment.

Additional analysis

The data analyst may also want to investigate the reasons for the high yearly fluctuations in the East, North, Northeast, South, and West regions. For example, they may want to look at the economic structure of these regions, the types of industries that are present, and the demographic characteristics of the population. This analysis could help to identify the factors that are driving the fluctuations and develop policies to mitigate them.

Thank you

Oasis infoByte task-2 by Dhvani Naik