

UNEMPLOYMENT ANALYSIS WITH PYTHON



```
importing libraries
```

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
pd.options.display.max_rows=5000
```

In [2]:

```
#reading csv file
df=pd.read_csv("C:\\Users\\ayith\\OneDrive\\Documents\\data sets\\Unemployment_Rate_upto
```

In [3]:

df.head()

Out[3]:

	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	78.2501
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	78.2501
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	78.2501
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	78.2501
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	78.2501

In [4]:

df.info() *# data info checking*

```
<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
```

In [5]:

```
df.isnull().sum() # finding null values
```

Out[5]:

```
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
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['Region', 'Date', 'Frequency', 'Estimated Unemployment Rate (%)',
      'Estimated Employed', 'Estimated Labour Participation Rate (%)',
      'Region.1', 'longitude', 'latitude'],
      dtype='object')
```

```
changing the column names
```

In [7]:

```
df = df.rename(columns={df.columns[0]: 'State', df.columns[1]: 'Date', df.columns[2]: 'Freque
df.head()
```

Out[7]:

	State	Date	Frequency	EUR	EE	ELPR	Region	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

In [8]:

```
df.columns
```

Out[8]:

```
Index(['State', 'Date', 'Frequency', 'EUR', 'EE', 'ELPR', 'Region',  
      'longitude', 'latitude'],  
      dtype='object')
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 267 entries, 0 to 266  
Data columns (total 9 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   EUR         267 non-null    float64  
4   EE          267 non-null    int64  
5   ELPR        267 non-null    float64  
6   Region      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
```

In [10]:

```
# finding the unique values with each column  
df['State'].unique()
```

Out[10]:

```
array(['Andhra Pradesh', 'Assam', 'Bihar', 'Chhattisgarh', 'Delhi', 'Goa',  
      'Gujarat', 'Haryana', 'Himachal Pradesh', 'Jammu & Kashmir',  
      'Jharkhand', 'Karnataka', 'Kerala', 'Madhya Pradesh',  
      'Maharashtra', 'Meghalaya', 'Odisha', 'Puducherry', 'Punjab',  
      'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana', 'Tripura',  
      'Uttar Pradesh', 'Uttarakhand', 'West Bengal'], dtype=object)
```

In [11]:

```
df['Region'].unique()
```

Out[11]:

```
array(['South', 'Northeast', 'East', 'West', 'North'], dtype=object)
```

In [12]:

```
df.groupby('Region').size()
```

Out[12]:

```
Region
East      40
North     79
Northeast 38
South     60
West      50
dtype: int64
```

In [13]:

```
round(df.groupby(['Region'])[['EUR', 'EE', 'ELPR']].mean().reset_index(),2)
```

Out[13]:

	Region	EUR	EE	ELPR
0	East	13.92	19602366.90	40.11
1	North	15.89	13072487.92	38.70
2	Northeast	10.95	3617105.53	52.06
3	South	10.45	14040589.33	40.44
4	West	8.24	18623512.72	41.26

In [14]:

```
df.groupby('State').size()
```

Out[14]:

State	
Andhra Pradesh	10
Assam	10
Bihar	10
Chhattisgarh	10
Delhi	10
Goa	10
Gujarat	10
Haryana	10
Himachal Pradesh	10
Jammu & Kashmir	9
Jharkhand	10
Karnataka	10
Kerala	10
Madhya Pradesh	10
Maharashtra	10
Meghalaya	10
Odisha	10
Puducherry	10
Punjab	10
Rajasthan	10
Sikkim	8
Tamil Nadu	10
Telangana	10
Tripura	10
Uttar Pradesh	10
Uttarakhand	10
West Bengal	10

dtype: int64

In [15]:

```
df.groupby('State')[['EUR', 'EE', 'ELPR']].mean().reset_index()
```

Out[15]:

	State	EUR	EE	ELPR
0	Andhra Pradesh	8.664000	1.542548e+07	38.962000
1	Assam	4.856000	1.081028e+07	43.498000
2	Bihar	19.471000	2.360683e+07	37.173000
3	Chhattisgarh	7.819000	8.421349e+06	41.161000
4	Delhi	18.414000	4.632822e+06	35.857000
5	Goa	12.167000	4.423748e+05	39.242000
6	Gujarat	6.376000	2.273075e+07	45.490000
7	Haryana	27.477000	6.844059e+06	42.100000
8	Himachal Pradesh	16.065000	2.033885e+06	40.252000
9	Jammu & Kashmir	16.477778	3.310032e+06	37.894444
10	Jharkhand	19.539000	8.770642e+06	40.356000
11	Karnataka	7.668000	2.162402e+07	42.000000
12	Kerala	9.434000	8.596795e+06	33.382000
13	Madhya Pradesh	6.854000	2.231834e+07	38.926000
14	Maharashtra	7.979000	3.920476e+07	41.466000
15	Meghalaya	3.866000	1.349815e+06	59.859000
16	Odisha	6.462000	1.272683e+07	37.748000
17	Puducherry	17.942000	3.652629e+05	35.918000
18	Punjab	11.981000	8.783034e+06	39.979000
19	Rajasthan	15.868000	1.973175e+07	40.591000
20	Sikkim	9.792500	2.345232e+05	45.756250
21	Tamil Nadu	12.187000	2.198790e+07	37.383000
22	Telangana	6.833000	1.624408e+07	54.972000
23	Tripura	25.055000	1.397292e+06	57.848000
24	Uttar Pradesh	9.737000	5.552480e+07	37.608000
25	Uttarakhand	11.156000	2.743275e+06	35.259000
26	West Bengal	10.192000	3.330516e+07	45.158000

In [16]:

```
df.columns
```

Out[16]:

```
Index(['State', 'Date', 'Frequency', 'EUR', 'EE', 'ELPR', 'Region',
      'longitude', 'latitude'],
      dtype='object')
```

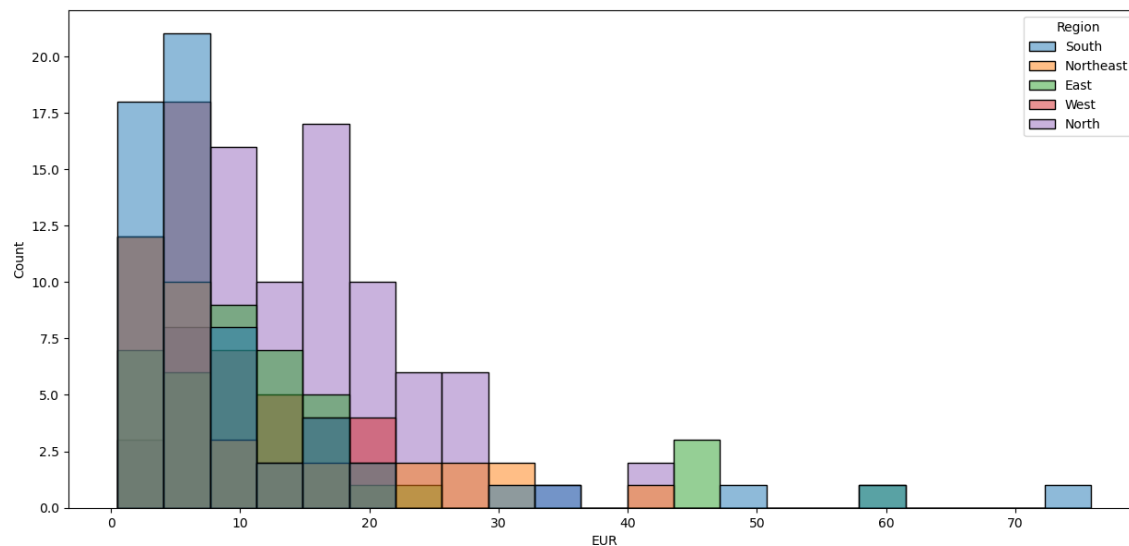
```
### checking the bar graphs of regional wise
```

In [17]:

```
plt.figure(figsize=(15,7))  
sns.histplot(x='EUR',hue='Region',data=df)
```

Out[17]:

<AxesSubplot:xlabel='EUR', ylabel='Count'>



In [18]:

```
unemployment = df[["State", "Region", "EUR"]]  
  
fig = px.sunburst(unemployment, path=['Region', 'State'], values='EUR',  
                  title= 'Unemployment rate in every State and Region', height=650)  
fig.show()
```

In [19]:

```
region = df.groupby(["Region"])[["EUR", "EE", "ELPR"]].mean()
region = pd.DataFrame(region).reset_index()

fig = px.bar(region, x="Region", y="EUR", color="Region", title="Average Unemployment Rat
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```

In [20]:

```
df['Frequency'].unique()
```

Out[20]:

```
array(['M'], dtype=object)
```

In [21]:

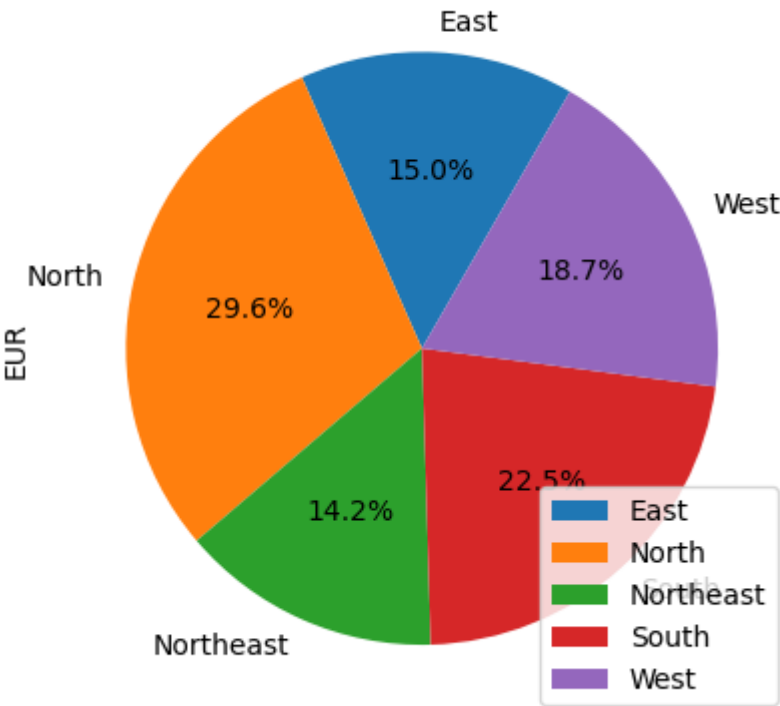
```
df['Date'].unique()
```

Out[21]:

```
array(['31-01-2020', '29-02-2020', '31-03-2020', '30-04-2020',
      '31-05-2020', '30-06-2020', '31-07-2020', '31-08-2020',
      '30-09-2020', '31-10-2020'], dtype=object)
```

In [22]:

```
df.groupby(['Region']).count().plot(kind='pie',y='EUR',autopct='%1.1f%%', startangle=60, plt.show())
```



In [23]:

```
df
```

Out[23]:

	State	Date	Frequency	EUR	EE	ELPR	Region	longitude	latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.7400
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.7400
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.7400
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.7400
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.7400

In [24]:

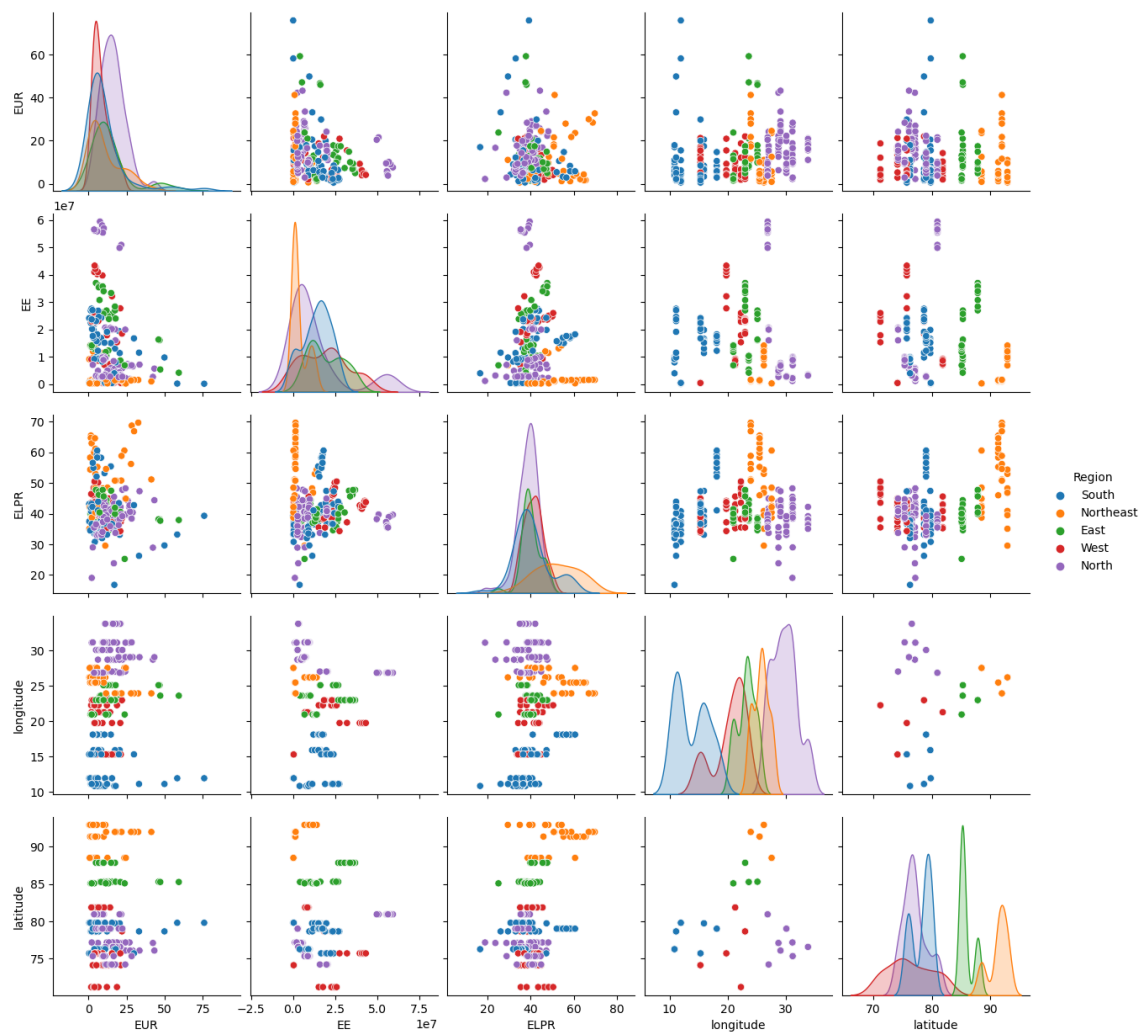
```
region = df.groupby(["State"])[["EUR"]].mean()
region = pd.DataFrame(region).reset_index()

fig = px.bar(region, x="State", y="EUR", title="Average Unemployment Rate by State")
fig.update_layout(xaxis={'categoryorder': 'total descending'})

fig.show()
```

In [25]:

```
sns.pairplot(df,hue='Region')  
plt.show()
```



In []: