Oasis Infobyte

Task 2 : Unemployment Analysis with Python

Context

The story behind this datasets is how lock-down affects employment opportunities and how the unemployment rate increases during the Covid-19.

Content

This dataset contains the unemployment rate of all the states in India

- Region = states in India
- Date = date which the unemployment rate observed
- Frequency = measuring frequency (Monthly)
- Estimated Unemployment Rate (%) = percentage of people unemployed in each States of India
- Estimated Employed = percentage of people employed
- Estimated Labour Participation Rate (%) = labour force participation rate by dividing the number of people actively participating in the labour force by the total number of people eligible to participate in the labor force

1. Import all necessary

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

2. Import dataframe

In [2]:
```
df1 = pd.read_csv("Unemployment in India.csv")
```

In [3]:
```
df1.head()
```

Out[3]:

	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

In [4]:
```
df1.tail()
```

Out[4]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
763	NaN	NaN	NaN	NaN	NaN	NaN	NaN
764	NaN	NaN	NaN	NaN	NaN	NaN	NaN
765	NaN	NaN	NaN	NaN	NaN	NaN	NaN
766	NaN	NaN	NaN	NaN	NaN	NaN	NaN
767	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [5]:
```
df2 = pd.read_csv("Unemployment_Rate_upto_11_2020.csv")
```

In [6]:
```
df2.head()
```

Out[6]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latitu
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.

In [7]:
```
df2.tail()
```

Out[7]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latit
262	West Bengal	30-06-2020	M	7.29	30726310	40.39	East	22.9868	87.
263	West Bengal	31-07-2020	M	6.83	35372506	46.17	East	22.9868	87.
264	West Bengal	31-08-2020	M	14.87	33298644	47.48	East	22.9868	87.
265	West Bengal	30-09-2020	M	9.35	35707239	47.73	East	22.9868	87.
266	West Bengal	31-10-2020	M	9.98	33962549	45.63	East	22.9868	87.

3. Check for Null Values

In [8]:
```
df2.isnull().sum()
```

Out[8]:
```
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
```

- Luckily, we dont have any null values in Unemployment_Rate_upto_11_2020 table

In [9]:
```
df1.isnull().sum()
```

Out[9]:
```
Region                                     28
 Date                                      28
 Frequency                                 28
 Estimated Unemployment Rate (%)           28
 Estimated Employed                        28
 Estimated Labour Participation Rate (%)   28
Area                                       28
dtype: int64
```

In [10]:
```
df1.dropna(inplace = True)
```

In [11]:
```
df1.isnull().sum()
```

Out[11]:
```
Region                                     0
 Date                                      0
 Frequency                                 0
 Estimated Unemployment Rate (%)           0
 Estimated Employed                        0
 Estimated Labour Participation Rate (%)   0
Area                                       0
dtype: int64
```

4. Check for Dublicate row

In [12]:
```
df1.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```
df2.duplicated().sum()
```

Out[13]:
```
0
```

so there is no dublicate row in our database

5. Summery of data

In [14]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 740 entries, 0 to 753
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: 46.2+ KB
```

In [15]:
```
df2.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
```

6. Check column name

In [16]:
```
df1.columns
```

Out[16]:
```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Area'],
      dtype='object')
```

In [17]:
```
df2.columns
```

Out[17]:
```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Region.1', 'longitude', 'latitude'],
      dtype='object')
```

7. Check the datatype

In [18]:
```
df1.dtypes
```

Out[18]:
```
Region                                     object
 Date                                      object
 Frequency                                 object
 Estimated Unemployment Rate (%)          float64
 Estimated Employed                       float64
 Estimated Labour Participation Rate (%)  float64
Area                                       object
dtype: object
```

In [19]:
```
df2.dtypes
```

Out[19]:
```
Region                                     object
 Date                                      object
 Frequency                                 object
 Estimated Unemployment Rate (%)          float64
 Estimated Employed                         int64
 Estimated Labour Participation Rate (%)  float64
Region.1                                   object
longitude                                 float64
latitude                                  float64
dtype: object
```

8.Shape of dataset

In [20]:
```
df1.shape
```

Out[20]:
```
(740, 7)
```

In [21]:
```
df2.shape
```

Out[21]:
```
(267, 9)
```

9. Find Corelation of data

In [22]:
```
df1.corr()
```

Out[22]:

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
Estimated Unemployment Rate (%)	1.000000	-0.222876	0.002558
Estimated Employed	-0.222876	1.000000	0.011300
Estimated Labour Participation Rate (%)	0.002558	0.011300	1.000000

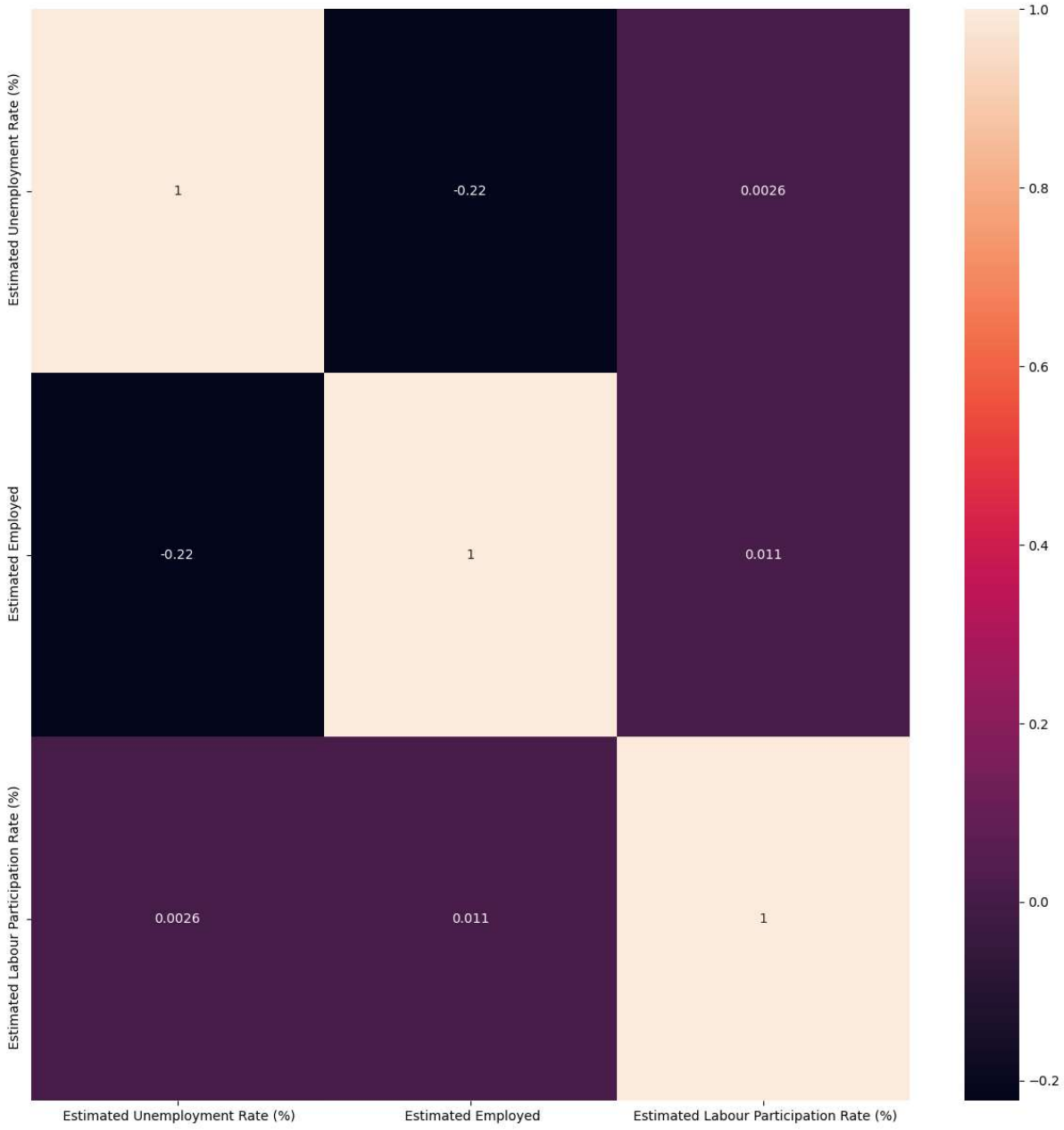
In [23]:
```
df2.corr()
```

Out[23]:

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	longitude	latitude
Estimated Unemployment Rate (%)	1.000000	-0.245176	-0.073540	0.149976	-0.023976
Estimated Employed	-0.245176	1.000000	-0.047948	-0.113664	-0.119321
Estimated Labour Participation Rate (%)	-0.073540	-0.047948	1.000000	0.080372	0.397836
longitude	0.149976	-0.113664	0.080372	1.000000	0.125895
latitude	-0.023976	-0.119321	0.397836	0.125895	1.000000

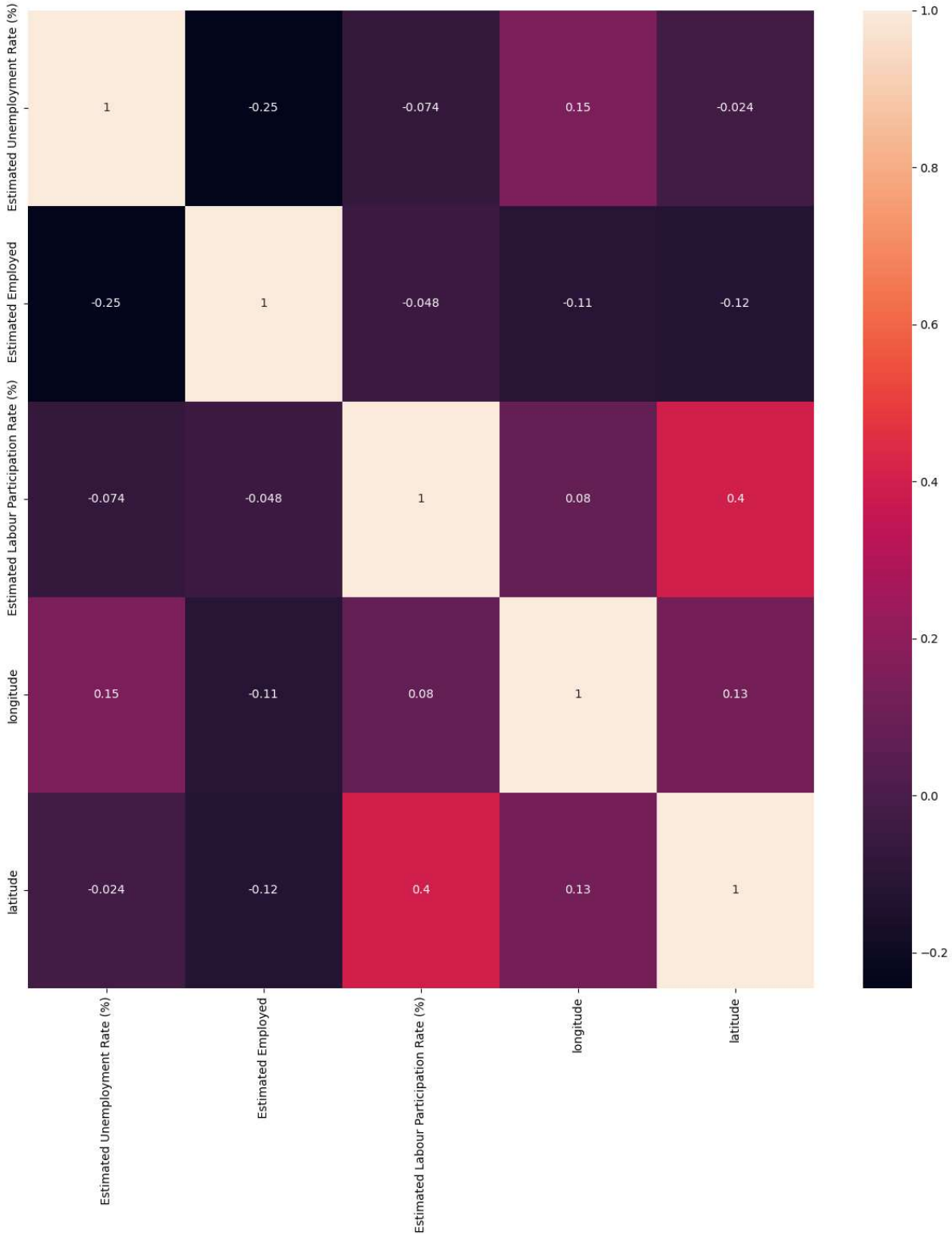
In [24]:
```
plt.figure(figsize=(15,15))
sns.heatmap(df1.corr() , annot=True)
```

Out[24]: <AxesSubplot:>



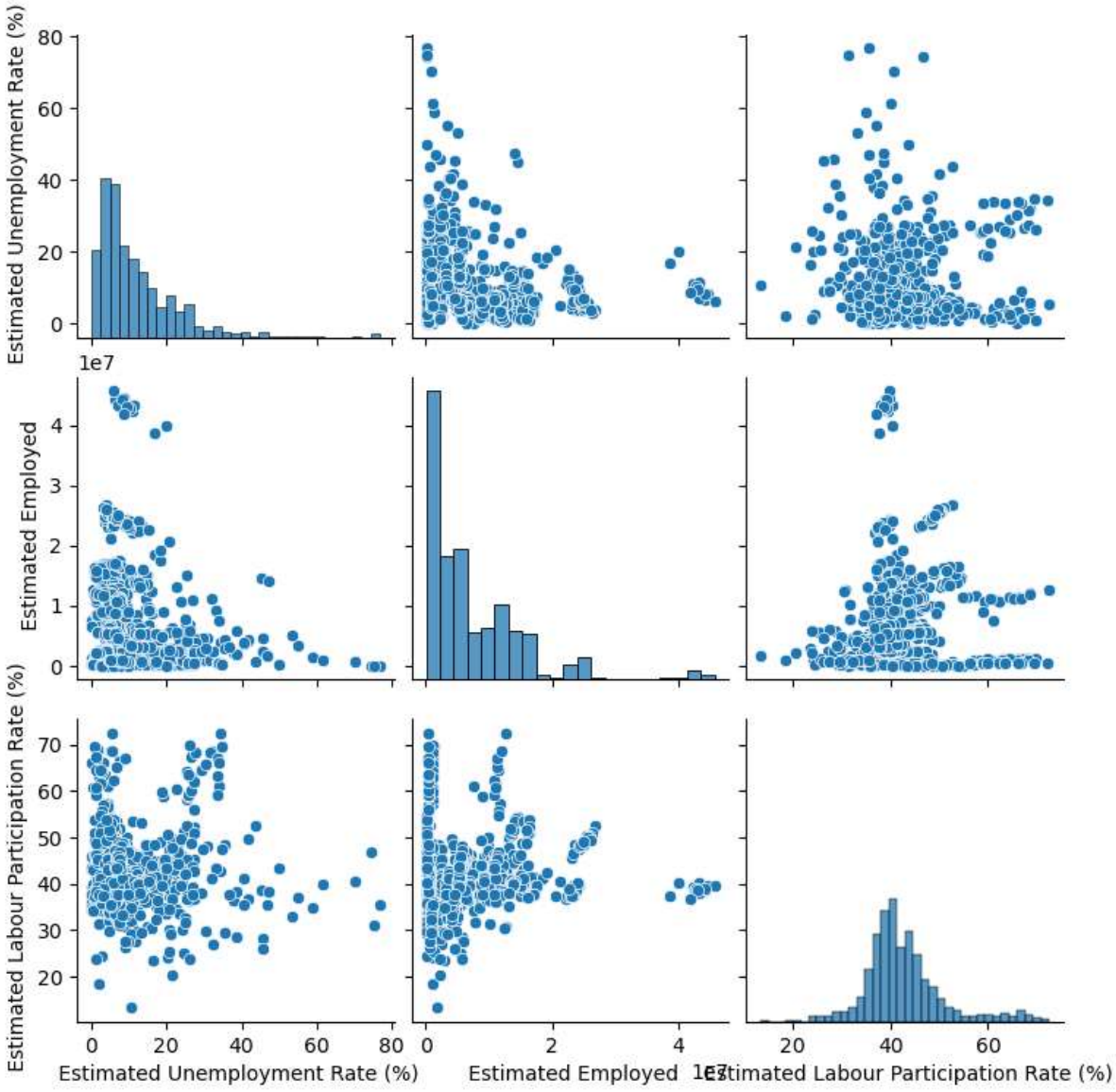
In [25]:
```
plt.figure(figsize=(15,15))
sns.heatmap(df2.corr() , annot=True)
```

Out[25]: <AxesSubplot:>



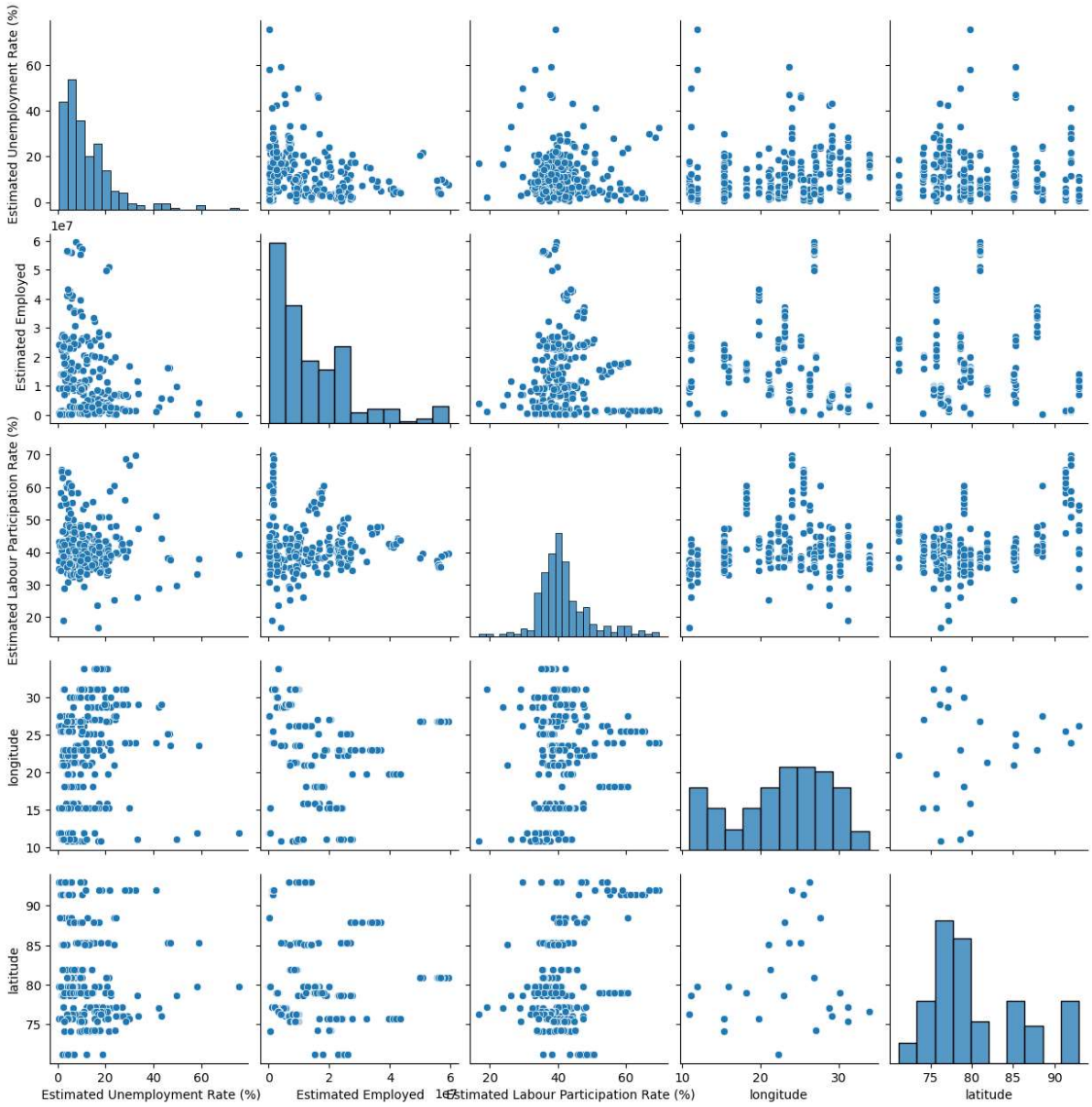
In [26]:
```
sns.pairplot(df1)
```

Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x24a0b4659a0>
```



In [27]:
```
sns.pairplot(df2)
```

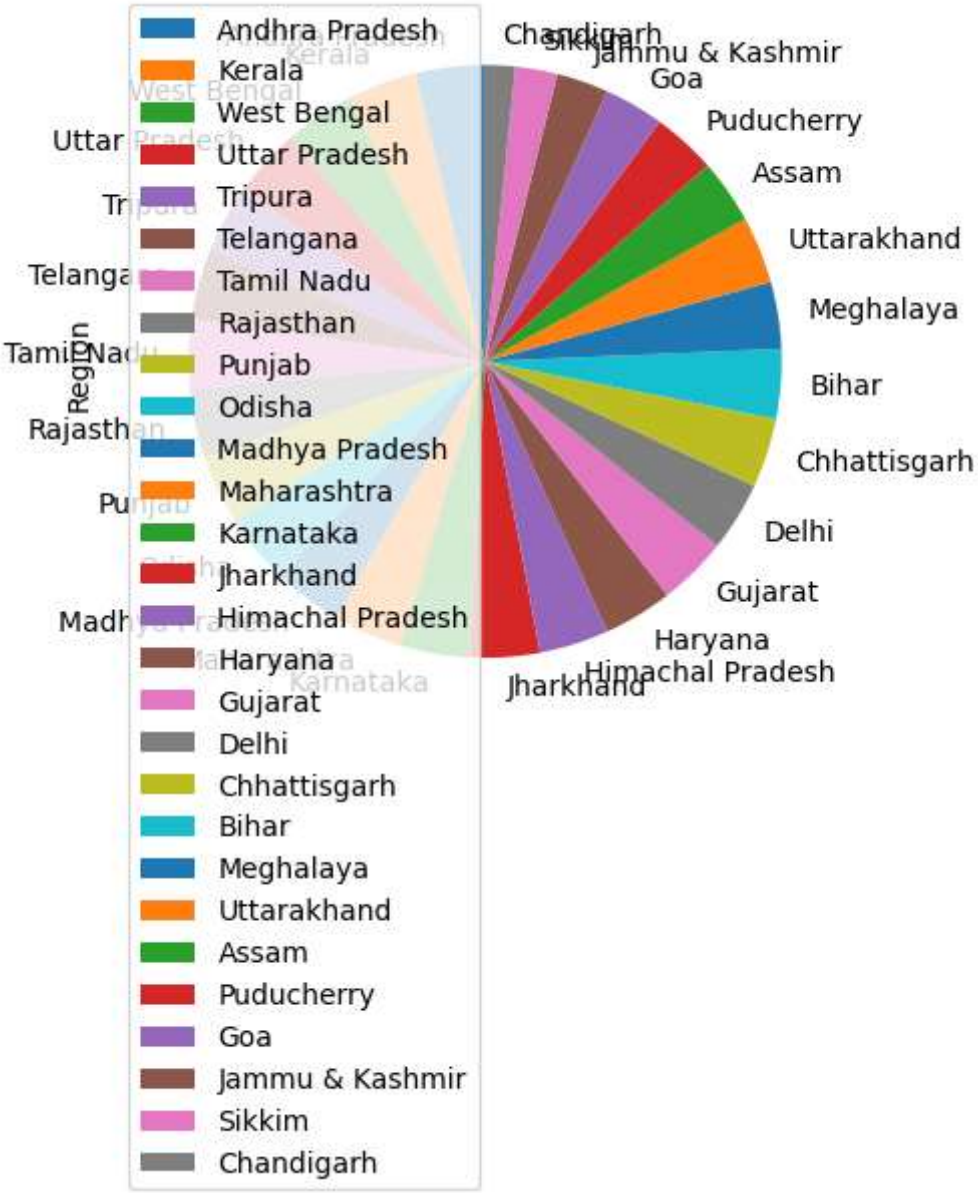
Out[27]:
```
<seaborn.axisgrid.PairGrid at 0x24a0c6f80a0>
```



In [28]:
```
df1['Region'].value_counts()
```

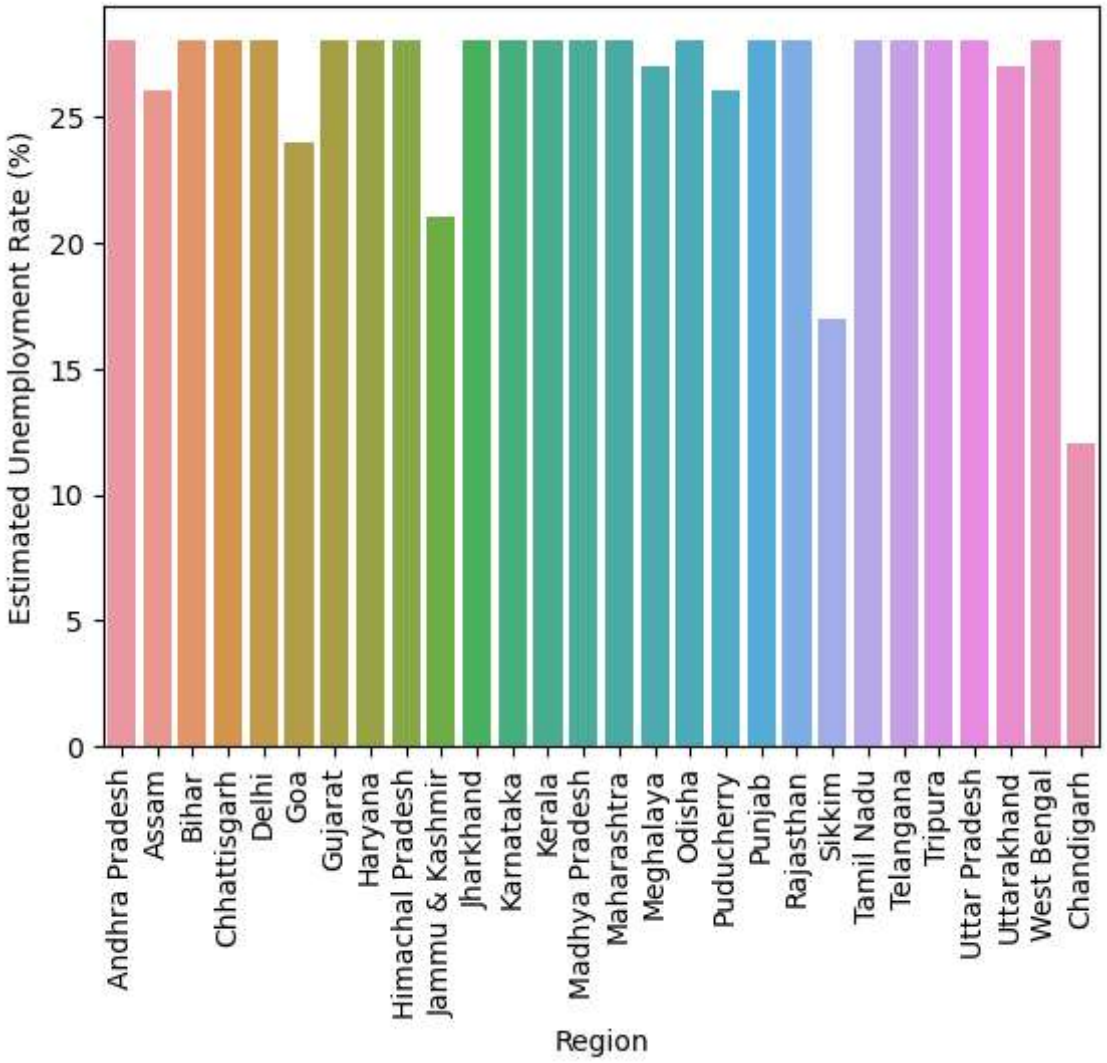
Out[28]:
```
Andhra Pradesh      28
Kerala              28
West Bengal         28
Uttar Pradesh       28
Tripura             28
Telangana           28
Tamil Nadu          28
Rajasthan           28
Punjab              28
Odisha              28
Madhya Pradesh      28
Maharashtra         28
Karnataka           28
Jharkhand           28
Himachal Pradesh    28
Haryana             28
Gujarat             28
Delhi               28
Chhattisgarh        28
Bihar               28
Meghalaya           27
Uttarakhand         27
Assam               26
Puducherry          26
Goa                 24
Jammu & Kashmir     21
Sikkim              17
Chandigarh          12
Name: Region, dtype: int64
```

In [29]:
```
(df1['Region'].value_counts()).plot(kind='pie',startangle=90)
plt.legend()
plt.show()
```



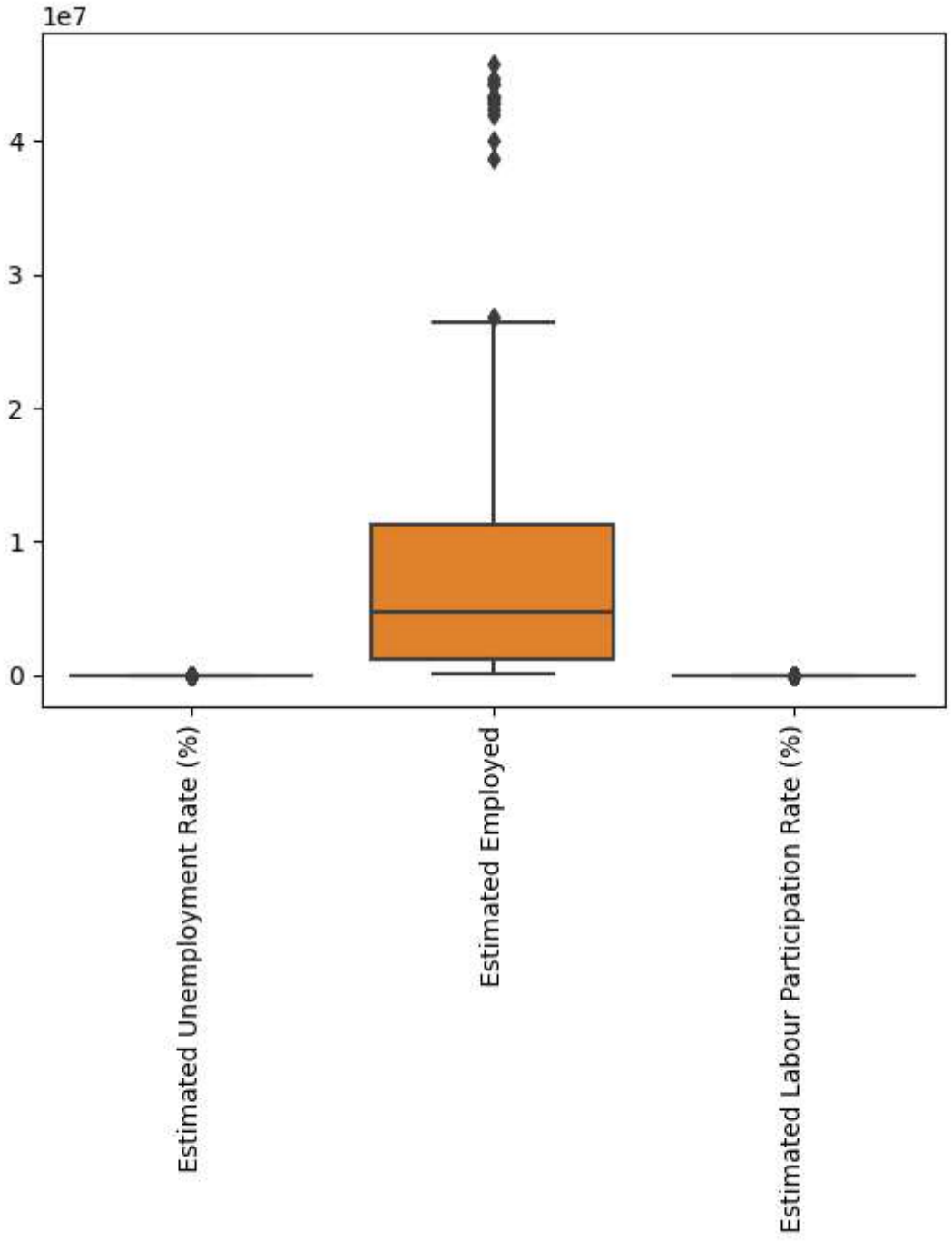
In [30]:
```
sns.countplot(x='Region',data=df1)
plt.xticks(rotation=90)
plt.ylabel('Estimated Unemployment Rate (%)')
```

Out[30]:
```
Text(0, 0.5, 'Estimated Unemployment Rate (%)')
```

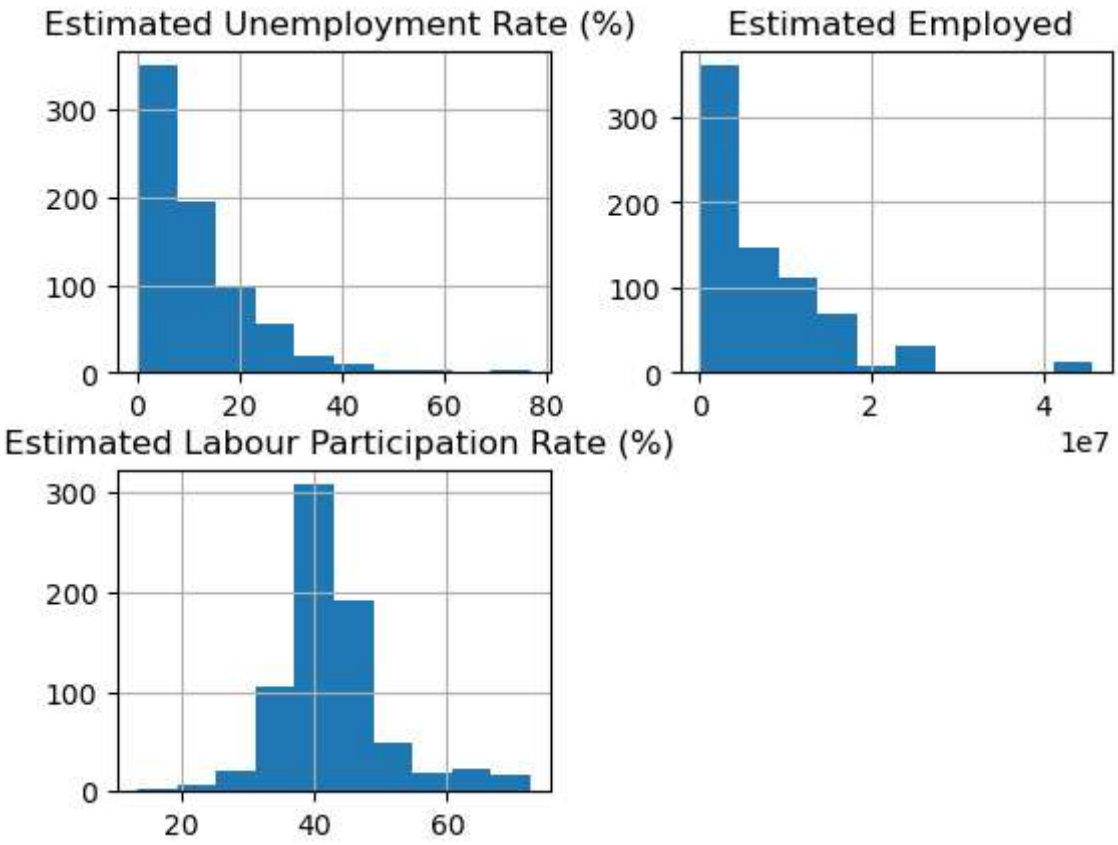


In [31]:
```
sns.boxplot(data = df1)
plt.xticks(rotation=90)
```

Out[31]:
```
(array([0, 1, 2]),
 [Text(0, 0, ' Estimated Unemployment Rate (%)'),
  Text(1, 0, ' Estimated Employed'),
  Text(2, 0, ' Estimated Labour Participation Rate (%)')])
```



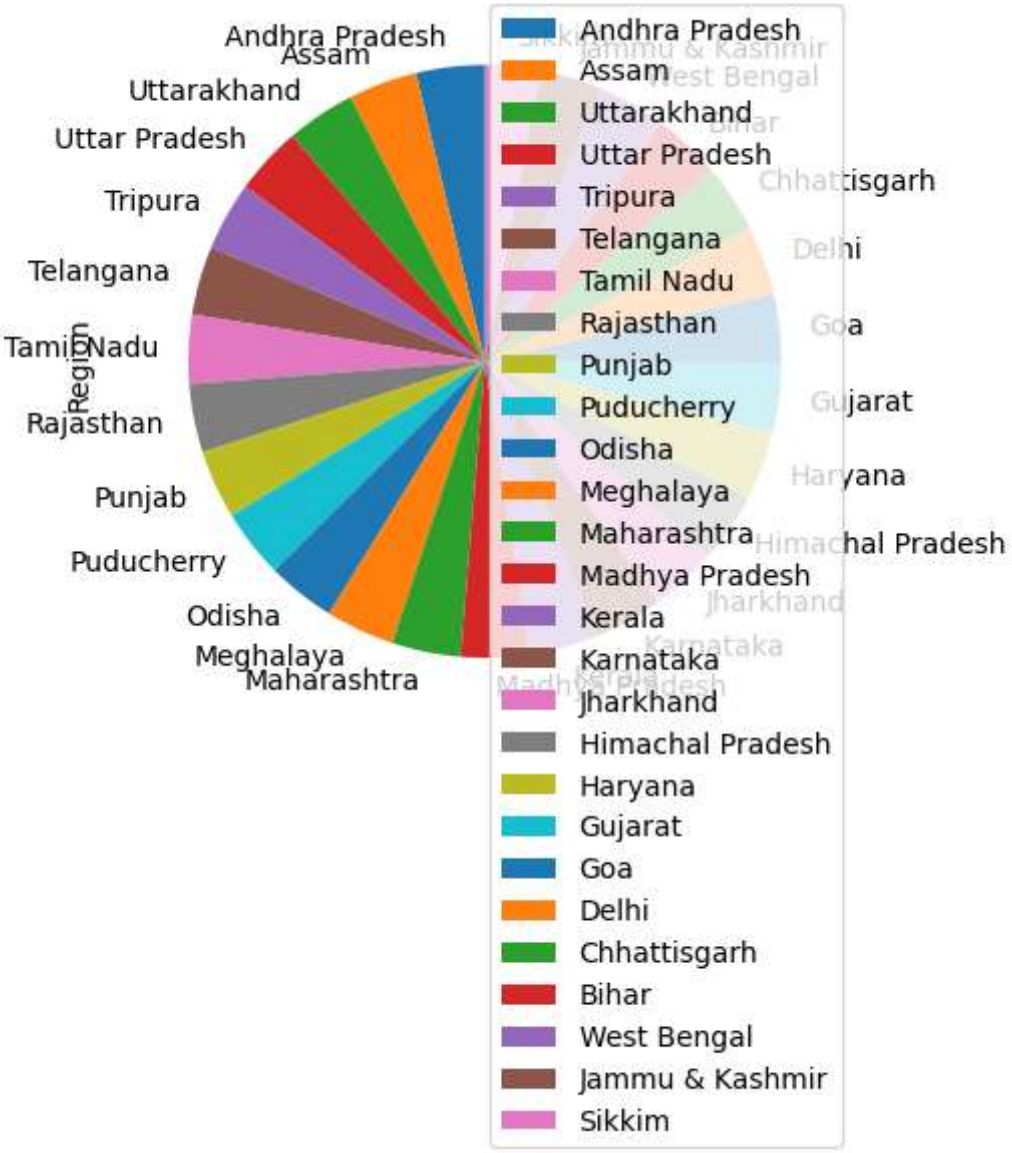
In [32]:
```
df1.hist()
plt.show()
```



In [33]:
```
df2['Region'].value_counts()
```

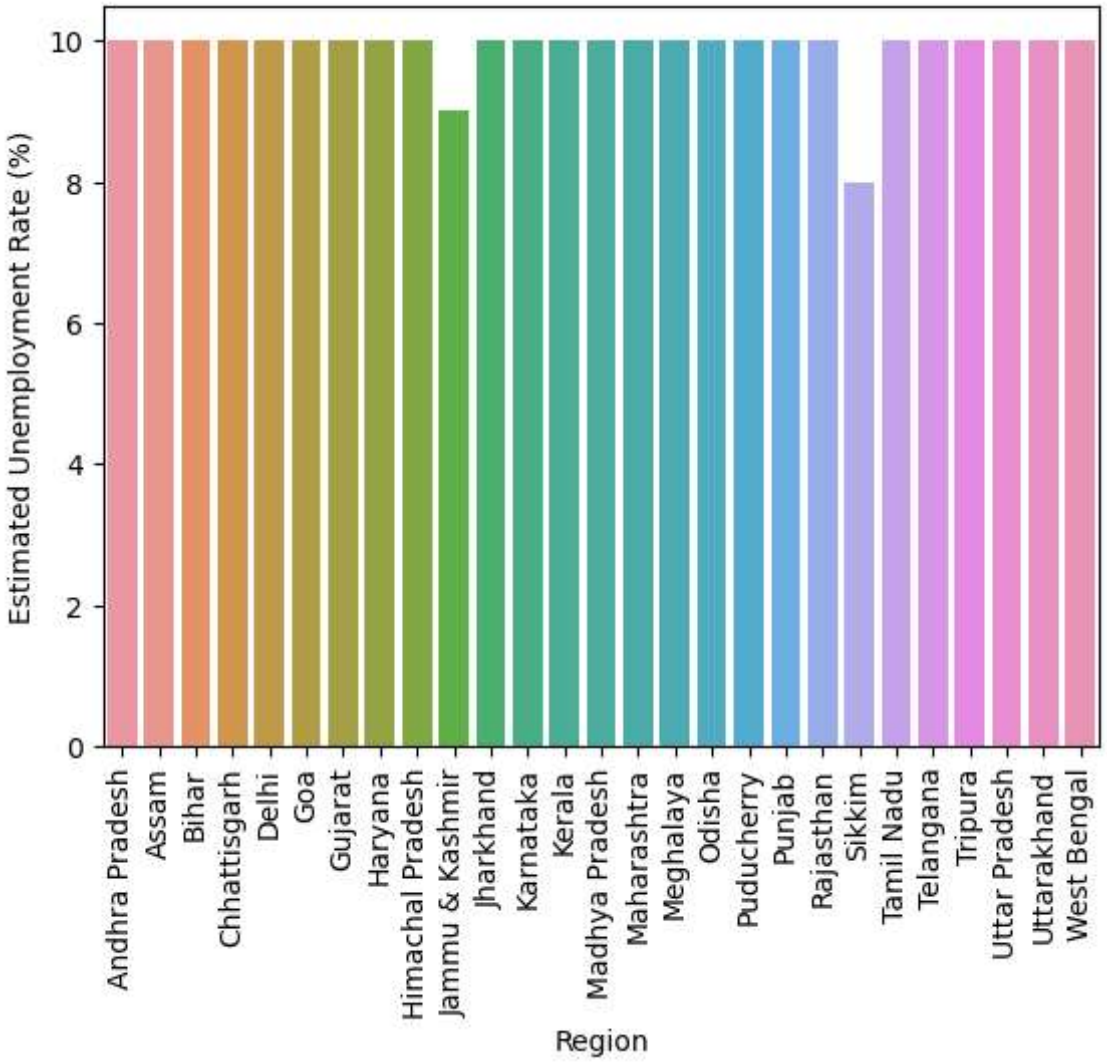
Out[33]:
```
Andhra Pradesh      10
Assam               10
Uttarakhand         10
Uttar Pradesh       10
Tripura             10
Telangana           10
Tamil Nadu          10
Rajasthan           10
Punjab              10
Puducherry          10
Odisha              10
Meghalaya           10
Maharashtra         10
Madhya Pradesh      10
Kerala              10
Karnataka           10
Jharkhand           10
Himachal Pradesh    10
Haryana             10
Gujarat             10
Goa                 10
Delhi               10
Chhattisgarh        10
Bihar               10
West Bengal         10
Jammu & Kashmir      9
Sikkim               8
Name: Region, dtype: int64
```

In [34]:
```
(df2['Region'].value_counts()).plot(kind='pie',startangle=90)
plt.legend()
plt.show()
```



In [35]:
```
sns.countplot(x='Region',data=df2)
plt.xticks(rotation=90)
plt.ylabel('Estimated Unemployment Rate (%)')
```

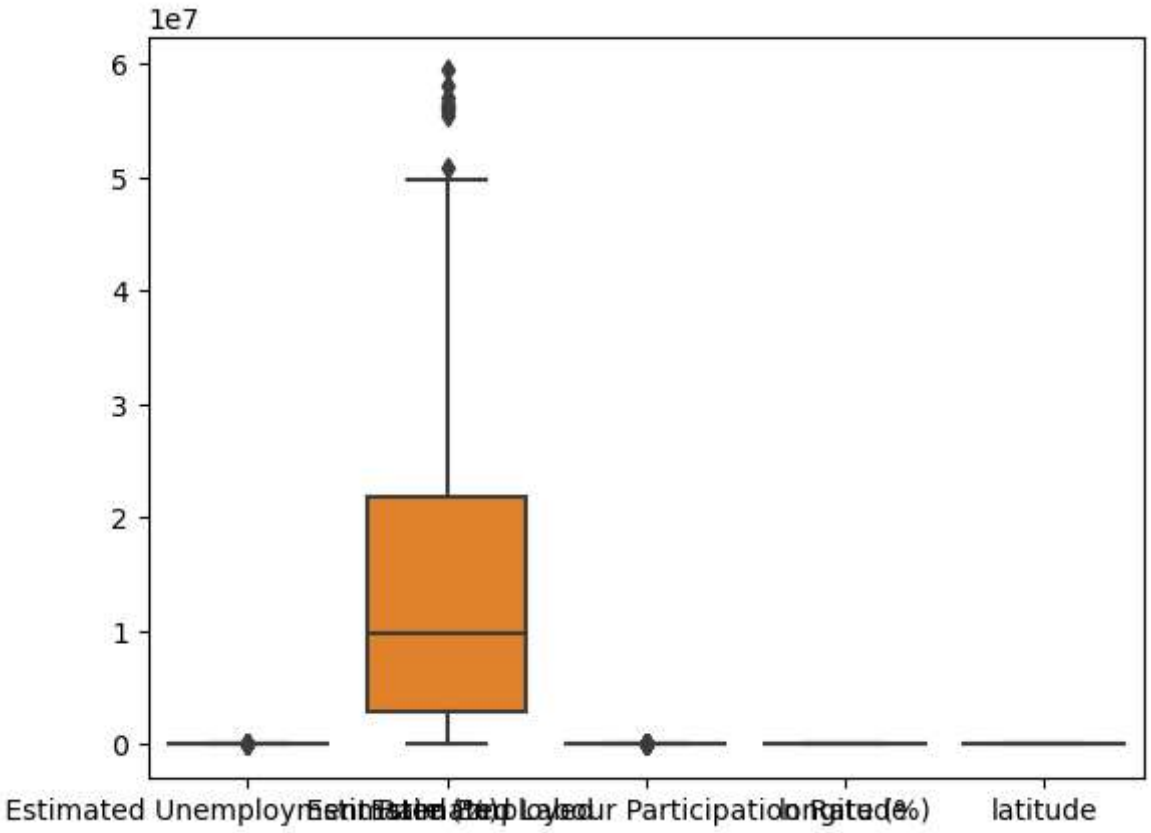
Out[35]:
```
Text(0, 0.5, 'Estimated Unemployment Rate (%)')
```



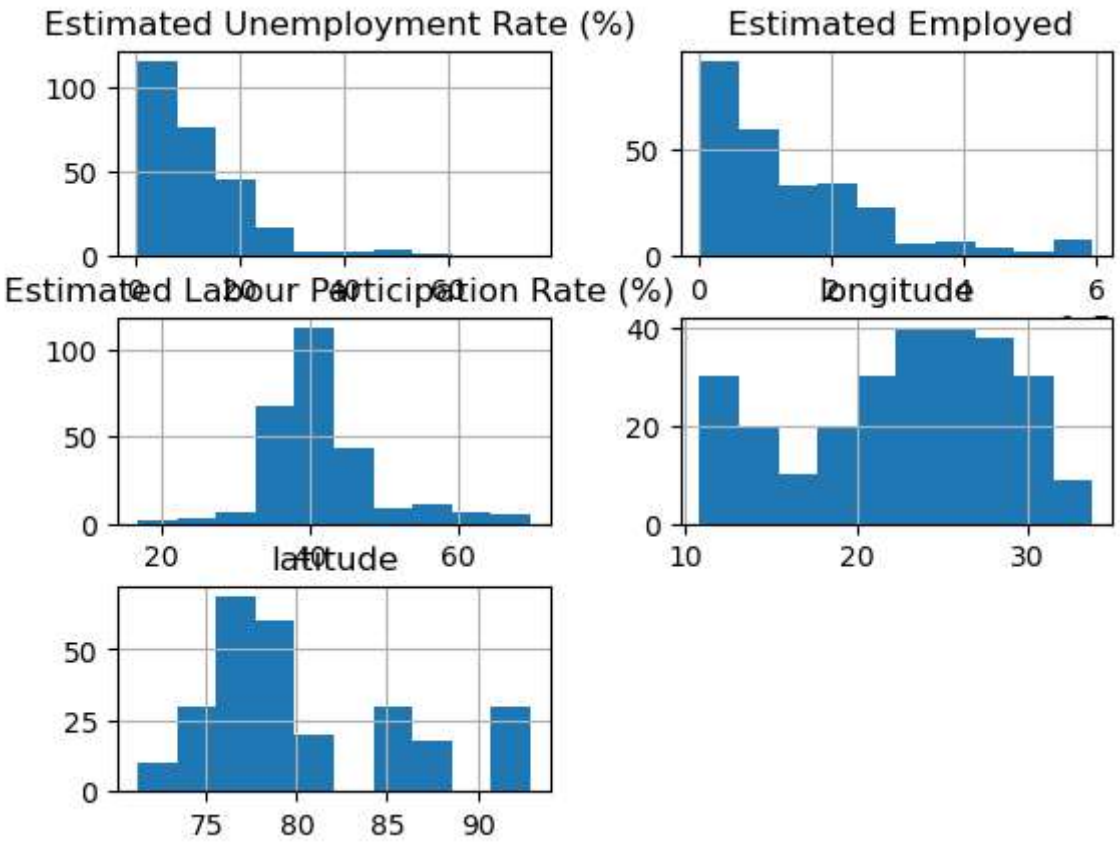
In [36]:
```
sns.boxplot(data = df2)
plt.xtickes(rotation = 90)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1292\2997186525.py in <module>
      1 sns.boxplot(data = df2)
----> 2 plt.xtickes(rotation = 90)

AttributeError: module 'matplotlib.pyplot' has no attribute 'xtickes'
```



In [37]:
```
df2.hist()
plt.show()
```



In [ ]:

In [ ]: