```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
df = pd.read_csv("Unemployment in India.csv")
```

```
df
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139	43.24	Rural
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881	42.05	Rural
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707	43.50	Rural
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762	44.68	Rural
...	...	...	...	...	...	...	...
					71168	44.09	Urban
736	West Bengal	31-03-2020	Monthly	6.67	10806105	43.34	Urban

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
round(df.describe())
```

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
count	740.0	740.0	740.0

```
print(df.isnull().sum())
```

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

max			2014.0	45777509.0		73.0	

```
df.dropna()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139	43.24	Rural
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881	42.05	Rural
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707	43.50	Rural
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693	43.97	Rural
					56762	44.68	Rural
...	...	...	...	...	...	...	...
735	West Bengal	29-02-2020	Monthly	7.55	10871168	44.09	Urban
736	West Bengal	31-03-2020	Monthly	6.67	10806105	43.34	Urban

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
df.describe()
```

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
count	740.000000	7.400000e+02	740.000000
mean	14.577919	7.204460e+06	42.633514
std	74.365885	8.087988e+06	8.111493
50%	8.415000	4.744178e+06	41.190000

```
import warnings
warnings.filterwarnings("ignore")
```

```
sns.set_theme()
```

```
df.nunique()
```

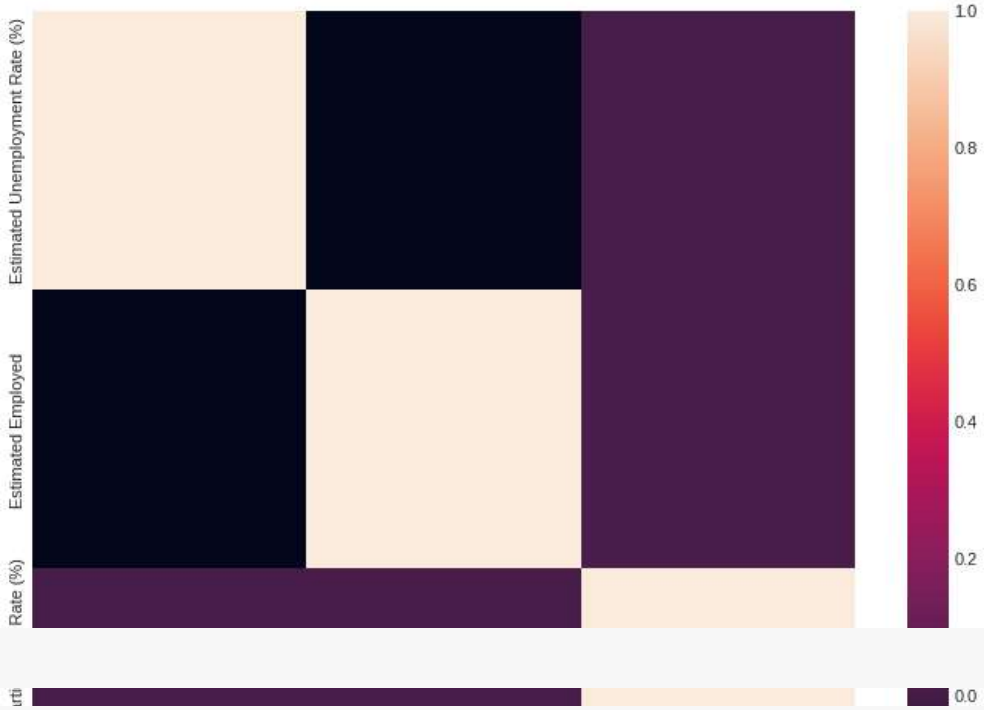
```
Region                                   28
 Date                                    14
 Frequency                                2
 Estimated Unemployment Rate (%)        629
 Estimated Employed                     740
 Estimated Labour Participation Rate (%)  626
Area                                      2
dtype: int64
```

```
df.isnull().sum()
```

```
Region                                   0
 Date                                    0
 Frequency                               0
 Estimated Unemployment Rate (%)         0
dtype: int64
```

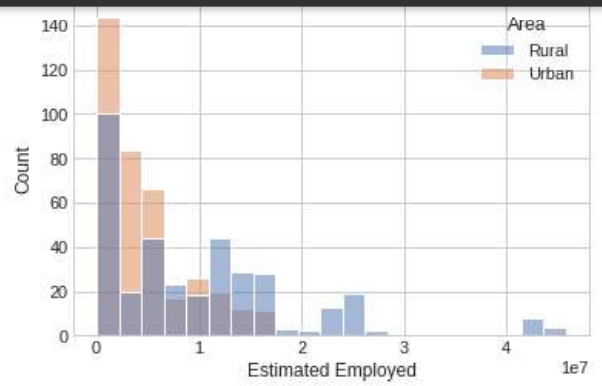
Automatic saving failed. This file was updated remotely or in another tab. Show diff

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



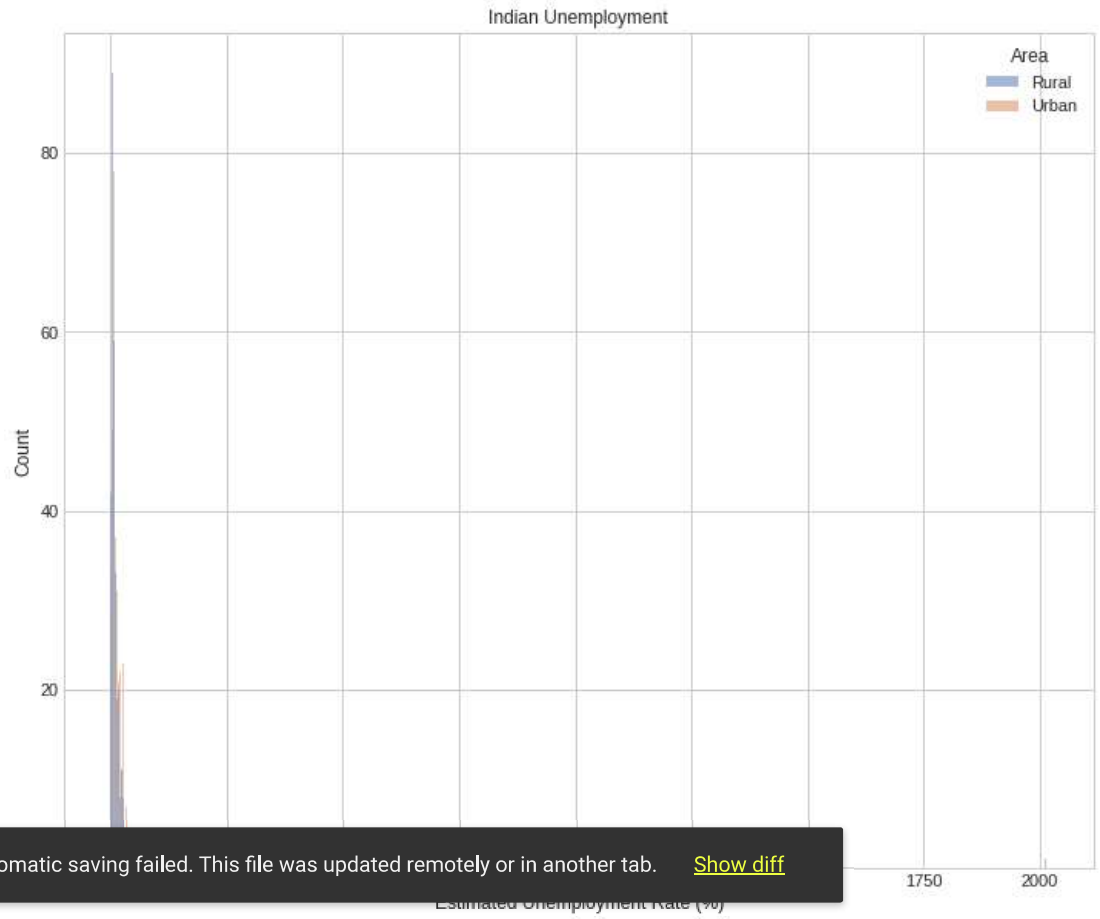

```
df.columns= ["Region","Date","Frequency",
               "Estimated Unemployment Rate (%)","Estimated Employed",
               "Estimated Labour Participation Rate (%)","Area"]
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Employed", hue="Area", data=df)
plt.show()
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff



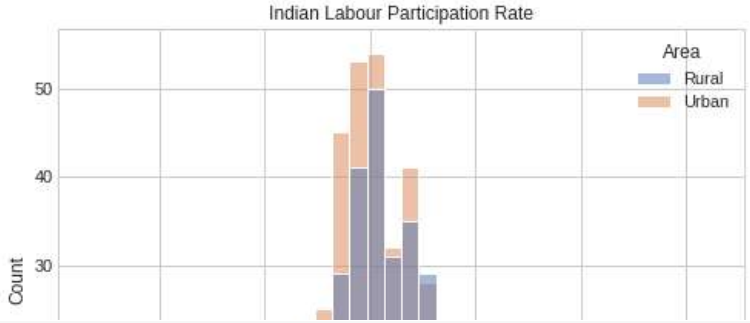

```
plt.figure(figsize=(12, 10))
plt.title("Indian Unemployment")
```

```
sns.histplot(x="Estimated Unemployment Rate (%)", hue="Area", data=df)
plt.show()
```



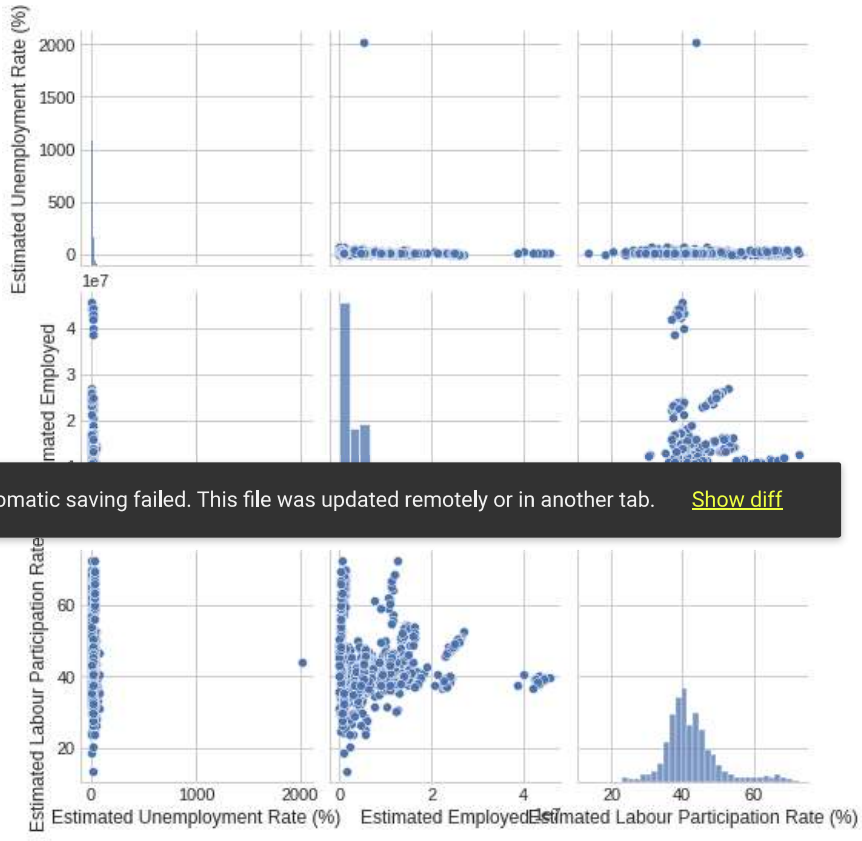
Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
plt.figure(figsize=(8,6))
plt.title("Indian Labour Participation Rate")
sns.histplot(x="Estimated Labour Participation Rate (%)",hue="Area",data=df)
plt.show()
```




```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f3aa004de80>
```




```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df["Region"] = le.fit_transform(df["Region"])
```

```
df["Frequency"] = le.fit_transform(df["Frequency"])
df["Area"] = le.fit_transform(df["Area"])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 740 entries, 0 to 739
Data columns (total 7 columns):
 #   Column                                   Non-Null Count  Dtype  
---  ------                                   --------------  -----  
 0   Region                                   740 non-null    int64  
 1   Date                                     740 non-null    object 
 2   Frequency                                740 non-null    int64  
 3   Estimated Unemployment Rate (%)          740 non-null    float64
 4   Estimated Employed                       740 non-null    int64  
 5   Estimated Labour Participation Rate (%)  740 non-null    float64
 6   Area                                     740 non-null    int64  
dtypes: float64(2), int64(4), object(1)
memory usage: 40.6+ KB
```

```
correlation_mat = df.corr()
```

```
print(correlation_mat)
```

```
                                          Region  Frequency  \
Region                                  1.000000  -0.013472   
Frequency                              -0.013472   1.000000   
Estimated Unemployment Rate (%)         0.016093  -0.019284   
Estimated Employed                      0.210371  -0.358902   
Estimated Labour Participation Rate (%)  0.169374  -0.220137   
Area                                   -0.013472   1.000000   

                                          Estimated Unemployment Rate (%)  \
Region                                                           0.016093   
Frequency                                                       -0.019284   
Estimated Unemployment Rate (%)                                  1.000000   
Estimated Employed                                              -0.041368   
Estimated Labour Participation Rate (%)                          0.005269   
Area                                                            -0.019284   

                                          Estimated Employed  \
Region                                              0.210371   
Frequency                                          -0.358902   
Estimated Unemployment Rate (%)                    -0.041368   
Estimated Employed                                  1.000000   
Estimated Labour Participation Rate (%)             0.011499   
Area                                               -0.358902   

                                          Estimated Labour Participation Rate (%)  \
Region                                                                   0.169374   
Frequency                                                               -0.220137   
Estimated Unemployment Rate (%)                                          0.005269   
Estimated Employed                                                       0.011499   
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
Estimated Labour Participation Rate (%)                                 1.000000
Area                                                                   -0.220137

                                        Area
Region                                -0.013472
Frequency                              1.000000
Estimated Unemployment Rate (%)       -0.019284
Estimated Employed                    -0.358902
Estimated Labour Participation Rate (%) -0.220137
Area                                   1.000000
```

```
sns.heatmap(correlation_mat,annot=True,linewidths=.5,cmap="YlGnBu")
```

```
<AxesSubplot:>
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
X = df.drop(columns=["Estimated Unemployment Rate (%)","Date","Estimated Employed"])
y = df["Estimated Unemployment Rate (%)"]
X_train , X_test , y_train , y_test = train_test_split(X,y,test_size=0.33,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler , MinMaxScaler

StSc = StandardScaler()
X_train  = StSc.fit_transform(X_train)
X_test  = StSc.fit_transform(X_test)
```

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(X_train, y_train)
y_pred = reg.predict(X_test)
```

```
from sklearn.metrics import mean_absolute_error as mae
mae(y_test,y_pred)
```

```
9.90115927950433
```

```
from sklearn.metrics import mean_squared_error as mse
mse(y_test, y_pred)
```

```
155.4251645722671
```

```
from sklearn.metrics import mean_absolute_percentage_error as mape
mape(y_test, y_pred)
```

```
2.0004977389891336
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

0s completed at 9:47 PM