

# UNEMPLOYMENT ANALYSIS WITH PYTHON

## Import the libraries

In [ ]:

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

## READ THE FILES

In [2]:

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

In [3]:

```
df.sample(5)
```

Out[3]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
489	Haryana	31-01-2020	Monthly	27.24	2630938.0	42.75	Urban
676	Tamil Nadu	30-11-2019	Monthly	2.65	11729952.0	37.38	Urban
116	Himachal Pradesh	31-01-2020	Monthly	16.67	2041035.0	45.11	Rural
9	Andhra Pradesh	29-02-2020	Monthly	5.91	11723617.0	42.83	Rural
314	Tripura	31-03-2020	Monthly	31.61	1060946.0	68.28	Rural

In [4]:

```
df2=pd.read_csv('Unemployment_Rate_upto_11_2020.csv')
```

## DATA CLEANING

In [7]:

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

Out[7]:

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

In [8]:

```
print(df.dtypes)
df.columns=['States', 'Date', 'Frequency', 'Estimated Unemployment Rate', 'Estimated Employed
```

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

In [35]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 6 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   States                               768 non-null    object
 1   Date                                 768 non-null    object
 2   Frequency                             768 non-null    object
 3   Estimated Unemployment Rate           768 non-null    float64
 4   Estimated Employed                     768 non-null    float64
 5   Estimated Labour Participation Rate    768 non-null    float64
dtypes: float64(3), object(3)
memory usage: 36.1+ KB
```

In [36]:

```
df.describe()
```

Out[36]:

	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate
count	768.000000	7.680000e+02	768.000000
mean	11.787946	7.204460e+06	42.630122
std	10.523784	7.938986e+06	7.961667
min	0.000000	4.942000e+04	13.330000
25%	4.720000	1.376532e+06	38.265000
50%	9.005000	5.064048e+06	41.430000
75%	15.472500	1.111390e+07	45.362500
max	76.740000	4.577751e+07	72.570000

In [10]:

```
df['Date'].fillna(df['Date'].mode().iloc[0],inplace=True)
df['Frequency'].fillna(df['Frequency'].mode().iloc[0],inplace=True)
df['States'].fillna(df['States'].mode().iloc[0],inplace=True)
df['Estimated Unemployment Rate'].fillna(df['Estimated Unemployment Rate'].mean(),inplace=True)
df['Estimated Employed'].fillna(df['Estimated Employed'].mean(),inplace=True)
df['Estimated Labour Participation Rate'].fillna(df['Estimated Labour Participation Rate'].mean(),inplace=True)
df.drop(['Area'],axis=1,inplace=True)
```

In [11]:

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

Out[11]:

```
States      0
Date        0
Frequency   0
Estimated Unemployment Rate  0
Estimated Employed  0
Estimated Labour Participation Rate  0
dtype: int64
```

In [14]:

```
df2.columns=['States','Date','Frequency','Estimated Unemployment Rate','Estimated Employed']
```

In [15]:

```
df2.isnull().sum()
```

Out[15]:

```
States          0
Date            0
Frequency       0
Estimated Unemployment Rate  0
Estimated Employed  0
Estimated Labour Participation Rate  0
Region          0
Longitude       0
Latitude        0
dtype: int64
```

In [16]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 6 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   States                               768 non-null    object
 1   Date                                768 non-null    object
 2   Frequency                           768 non-null    object
 3   Estimated Unemployment Rate          768 non-null    float64
 4   Estimated Employed                   768 non-null    float64
 5   Estimated Labour Participation Rate  768 non-null    float64
dtypes: float64(3), object(3)
memory usage: 36.1+ KB
```

In [17]:

```
df2.describe()
```

Out[17]:

	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate
count	768.000000	7.680000e+02	768.000000
mean	11.787946	7.204460e+06	42.630122
std	10.523784	7.938986e+06	7.961667
min	0.000000	4.942000e+04	13.330000
25%	4.720000	1.376532e+06	38.265000
50%	9.005000	5.064048e+06	41.430000
75%	15.472500	1.111390e+07	45.362500
max	76.740000	4.577751e+07	72.570000

## Analysis through Visualization

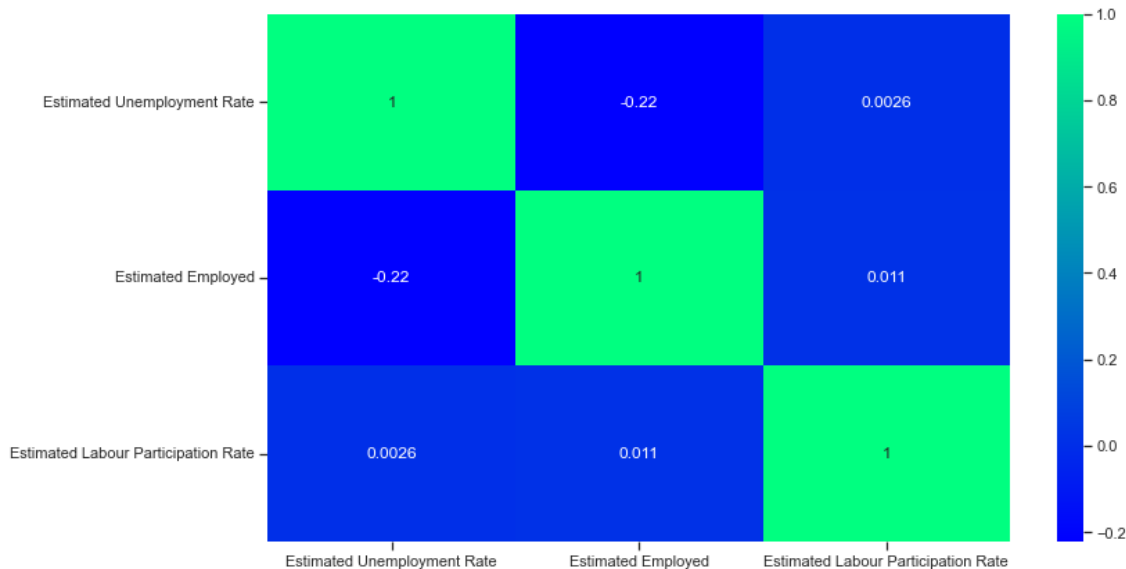
In [39]:

```

heat_maps = df[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Par
heat_maps = heat_maps.corr()

plt.figure(figsize=(12,7))
sns.set_context('notebook', font_scale=1)
sns.heatmap(heat_maps, annot=True, cmap='winter');

```



In [46]:

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

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

In [44]:

```
df.States.value_counts()
```

Out[44]:

```
States
Andhra Pradesh    56
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: count, dtype: int64
```

In [59]:

```
plt.figure(figsize=(35,10))
plt.title('Indian Unemployment')
sns.histplot(x="Estimated Unemployment Rate",hue="States",data=df)
plt.show()
```

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

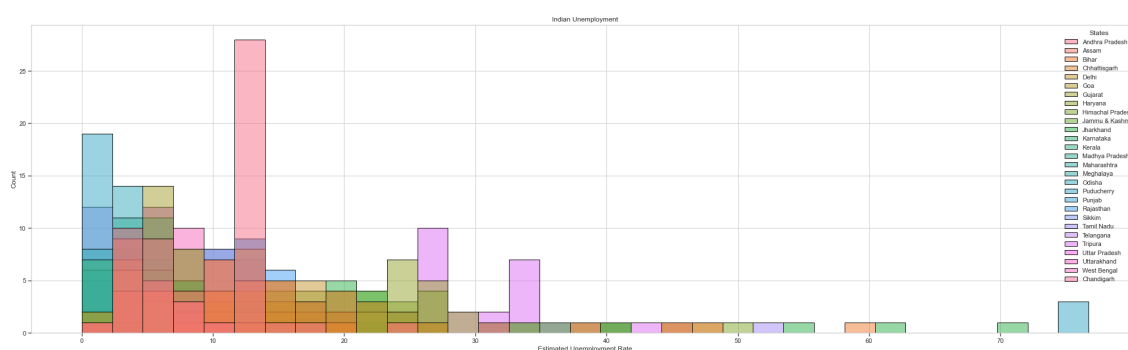
is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1498: FutureWarning:

is\_categorical\_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

C:\Users\sanja\anaconda3\lib\site-packages\seaborn\\_oldcore.py:1119: FutureWarning:

use\_inf\_as\_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.



In [53]:

```
import plotly.express as px
fg=px.histogram(df,x='States',y='Estimated Unemployment Rate',title='Indian unemployment
fg.update_layout(xaxis={'categoryorder':'total descending'})
fg.show()
```

In [61]:

```
df2.sample()
```

Out[61]:

	Region	Date	Frequency	Unemployment_rate	Employed	labour_participation_rate	area
25	Bihar	30-06-2020	M	17.82	23963933	36.86	East

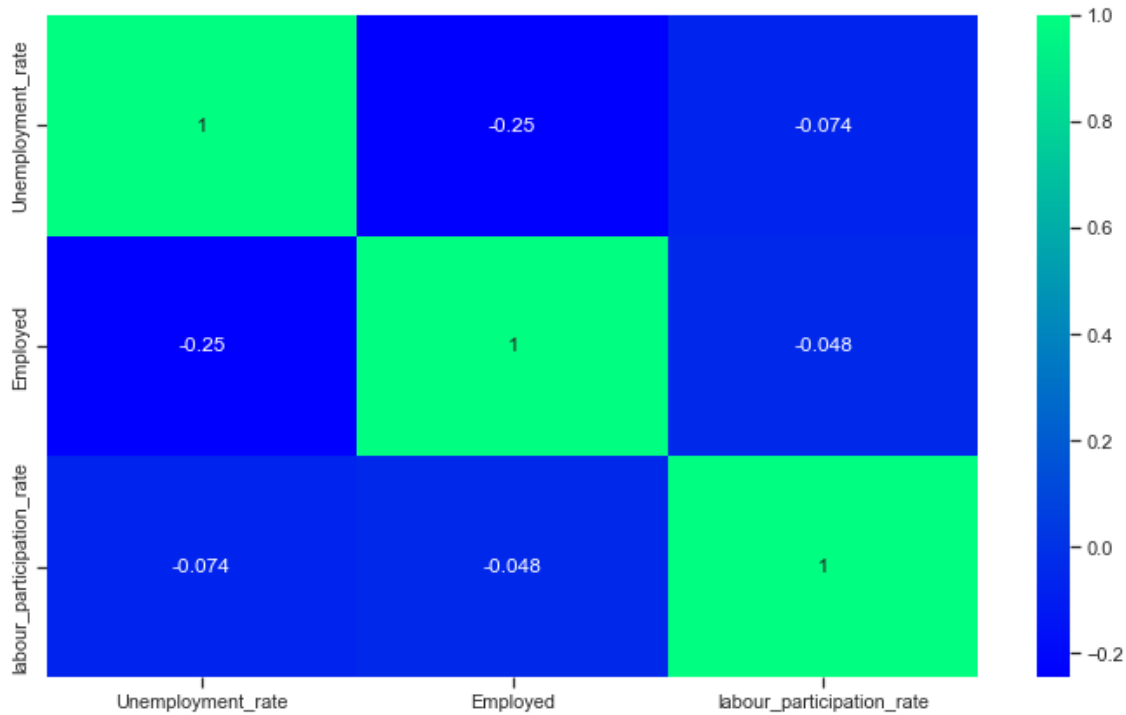


In [65]:

```
heat_maps = df2[['Unemployment_rate', 'Employed', 'labour_participation_rate']]  
heat_maps = heat_maps.corr()  
  
plt.figure(figsize=(12,7))  
sns.set_context('notebook', font_scale=1)  
sns.heatmap(heat_maps, annot=True, cmap='winter')
```

Out[65]:

<AxesSubplot:>



In [67]:

```
df2.Region.value_counts()
```

Out[67]:

Region	
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: count, dtype: int64

In [68]:

```
region = df2.groupby(["Region"])[['Unemployment_rate', "Employed", "labour_participation_rate"]].mean()
region = pd.DataFrame(region).reset_index()

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

In [69]:

```
region = df2.groupby(["Region"])[['Unemployment_rate', "Employed", "labour_participation_rate"]].mean()
region = pd.DataFrame(region).reset_index()

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

## Summary

So this is how you can analyze the unemployment rate by using the Python programming language. Unemployment is measured by the unemployment rate which is the number of people who are unemployed as a percentage of the total labour force. I hope you liked this article on unemployment rate analysis with Python. Feel free to ask your valuable questions in the comments section below.