Name of the Intern : SHAHNAWAZ ALAM

Data Science Internship

Oasis Infobyte

Task 1: Unemployement Analysis with Machine Learning

Batch- April Phase 1 OIBSIP

Importing Libraries

In [1]:

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

Importing Dataset¶

In [2]:

```
print("Importing data...")
df=pd.read_csv(r"C:\Users\md naiyer azam\Desktop\OIBSIP_Internship\Data Science\Unemployment_Rate_upto_11_2020.csv
print("Sucessfully imported.")
```

```
Importing data...
Sucessfully imported.
```

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	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 [4]:

```
df.head(10)
```

Out[4]:

	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	79.74
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
5	Andhra Pradesh	30-06-2020	M	3.31	19805400	47.41	South	15.9129	79.74
6	Andhra Pradesh	31-07-2020	M	8.34	15431615	38.91	South	15.9129	79.74
7	Andhra Pradesh	31-08-2020	M	6.96	15251776	37.83	South	15.9129	79.74
8	Andhra Pradesh	30-09-2020	M	6.40	15220312	37.47	South	15.9129	79.74
9	Andhra Pradesh	31-10-2020	M	6.59	15157557	37.34	South	15.9129	79.74

In [5]:

```
df.tail()
```

Out[5]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latitude
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

In [6]:

```
df.shape   ##to get no. of rows and column(rows,column)
```

Out[6]:

```
(267, 9)
```

In [7]:

```
df.info()    #info of data
```

```
<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 [8]:

```
df.describe()  #description of data
```

Out[8]:

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	longitude	latitude
count	267.000000	2.670000e+02	267.000000	267.000000	267.000000
mean	12.236929	1.396211e+07	41.681573	22.826048	80.532425
std	10.803283	1.336632e+07	7.845419	6.270731	5.831738
min	0.500000	1.175420e+05	16.770000	10.850500	71.192400
25%	4.845000	2.838930e+06	37.265000	18.112400	76.085600
50%	9.650000	9.732417e+06	40.390000	23.610200	79.019300
75%	16.755000	2.187869e+07	44.055000	27.278400	85.279900
max	75.850000	5.943376e+07	69.690000	33.778200	92.937600

In [9]:

```
x = df['Region']
```

In [10]:

```
x
```

Out[10]:

```
0      Andhra Pradesh
1      Andhra Pradesh
2      Andhra Pradesh
3      Andhra Pradesh
4      Andhra Pradesh
           ...
262       West Bengal
263       West Bengal
264       West Bengal
265       West Bengal
266       West Bengal
Name: Region, Length: 267, dtype: object
```

In [11]:

```
y = df[' Estimated Unemployment Rate (%)']
```

In [12]:

```
y
```

Out[12]:

```
0       5.48
1       5.83
2       5.79
3      20.51
4      17.43
       ...
262     7.29
263     6.83
264    14.87
265     9.35
266     9.98
Name:  Estimated Unemployment Rate (%), Length: 267, dtype: float64
```

In [13]:

```
df1=df.iloc[:,3]
```

In [14]:

```
df1
```

Out[14]:

```
0       5.48
1       5.83
2       5.79
3      20.51
4      17.43
       ...
262     7.29
263     6.83
264    14.87
265     9.35
266     9.98
Name:  Estimated Unemployment Rate (%), Length: 267, dtype: float64
```

In [15]:

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

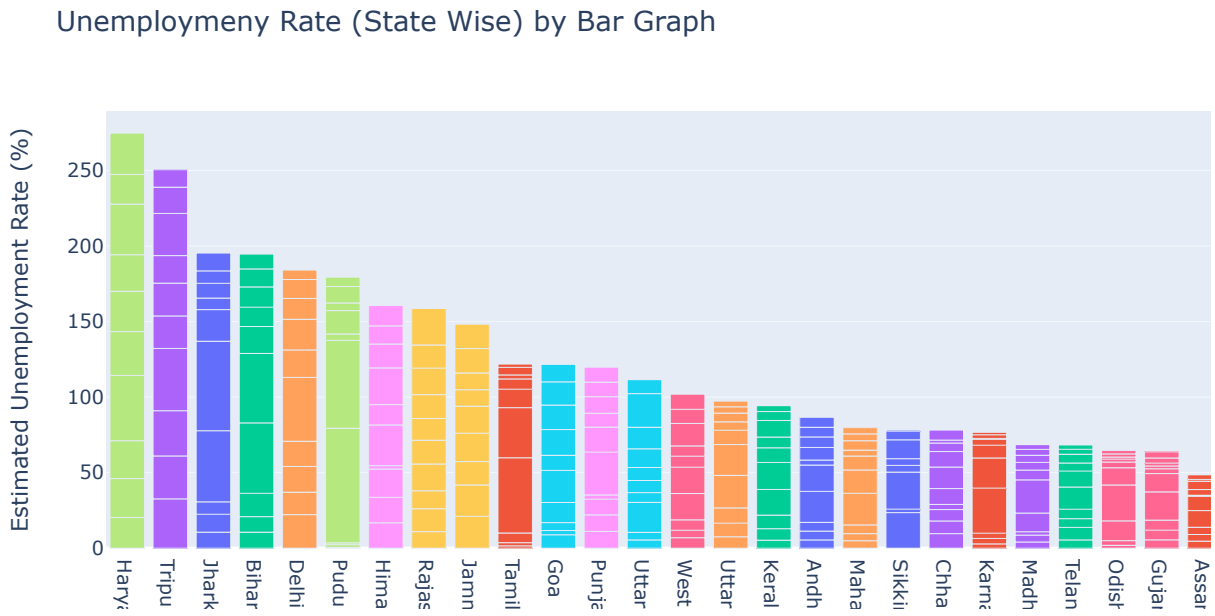
Out[15]:

```
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
```

Aanalyzing Data By Bargraph

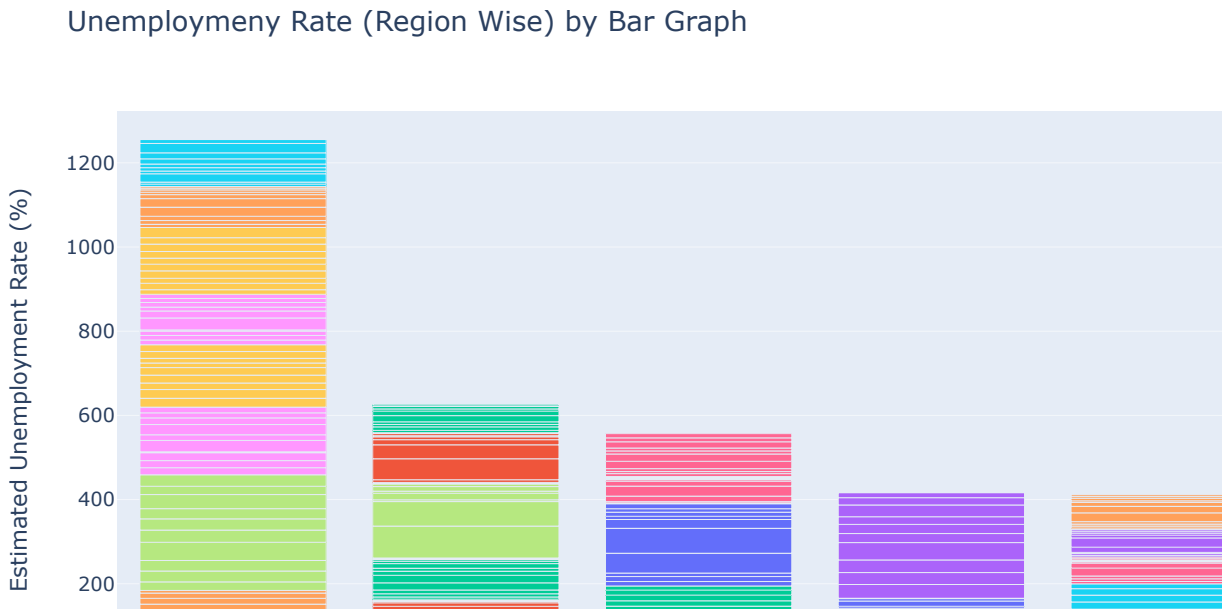
In [16]:

```
graph = px.bar(df,x='Region',y=' Estimated Unemployment Rate (%)',color='Region',
            title='Unemploymeny Rate (State Wise) by Bar Graph',template='plotly')
graph.update_layout(xaxis={'categoryorder':'total descending'})
graph.show()
```



In [18]:

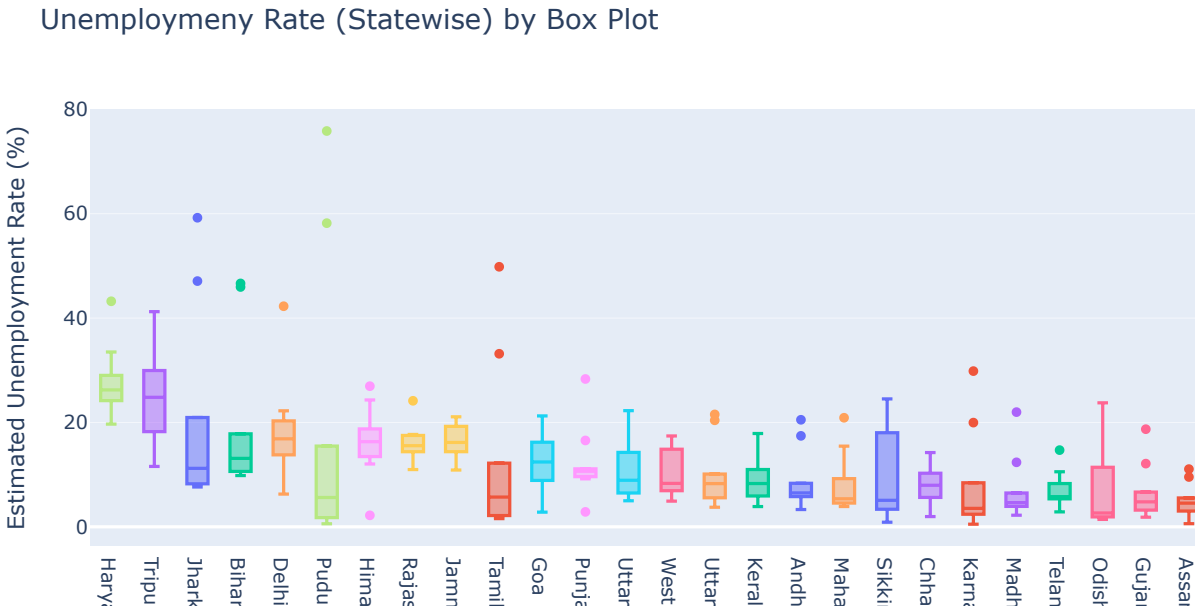
```
graph = px.bar(df,x='Region.1',y=' Estimated Unemployment Rate (%)',color='Region',
            title='Unemploymeny Rate (Region Wise) by Bar Graph',template='plotly')
graph.update_layout(xaxis={'categoryorder':'total descending'})
graph.show()
```



Aanalyzing Data By Boxplot

In [19]:

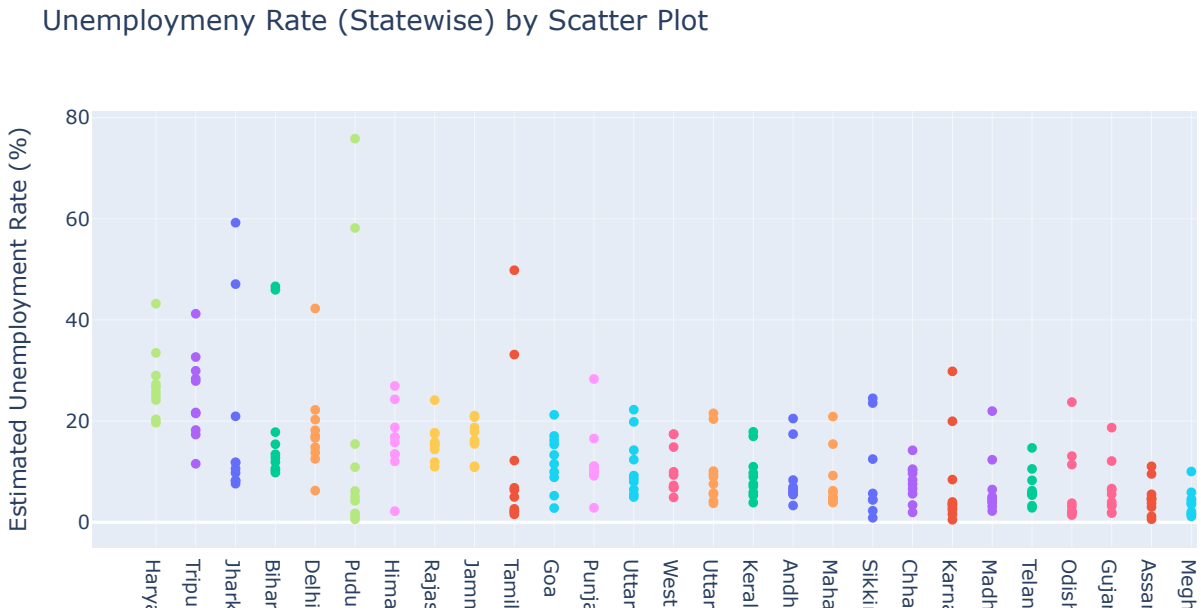
```
graph = px.box(df,x='Region',y=' Estimated Unemployment Rate (%)',color='Region',
           title='Unemploymeny Rate (Statewise) by Box Plot',template='plotly')
graph.update_layout(xaxis={'categoryorder':'total descending'})
graph.show()
```



Aanalyzing Data By Scatter Plot

In [20]:

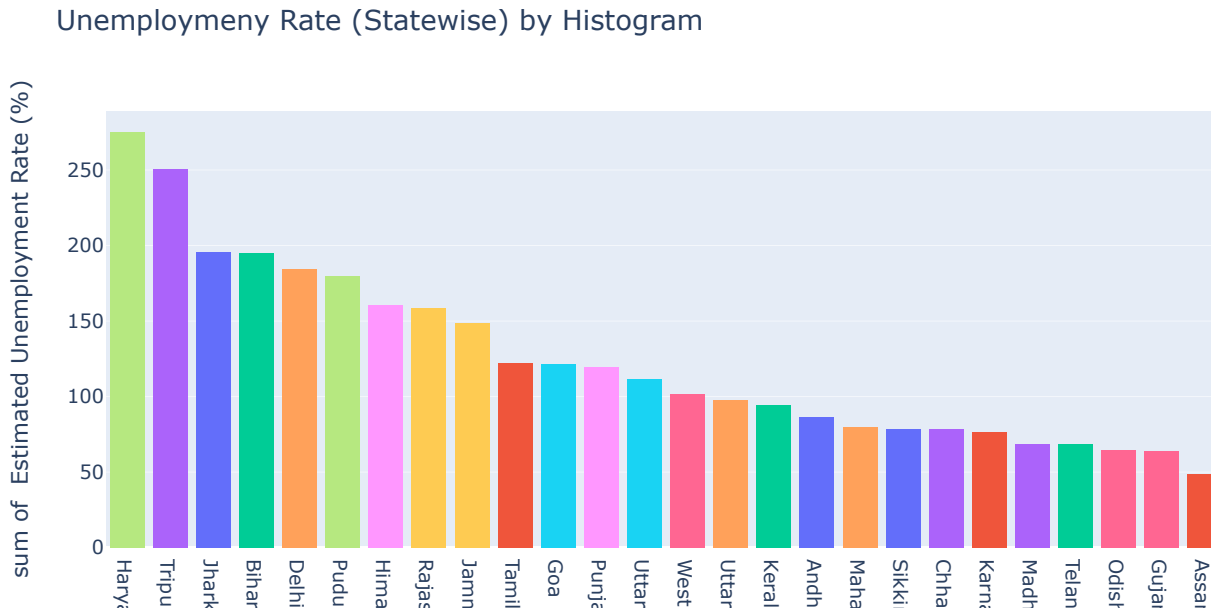
```
graph = px.scatter(df,x='Region',y=' Estimated Unemployment Rate (%)',color='Region',
                   title='Unemploymeny Rate (Statewise) by Scatter Plot',template='plotly')
graph.update_layout(xaxis={'categoryorder':'total descending'})
graph.show()
```



Aanalyzing Data By Histogram Plot

In [21]:

```
graph = px.histogram(df,x='Region',y=' Estimated Unemployment Rate (%)',color='Region',
                     title='Unemploymeny Rate (Statewise) by Histogram',template='plotly')
graph.update_layout(xaxis={'categoryorder':'total descending'})
graph.show()
```

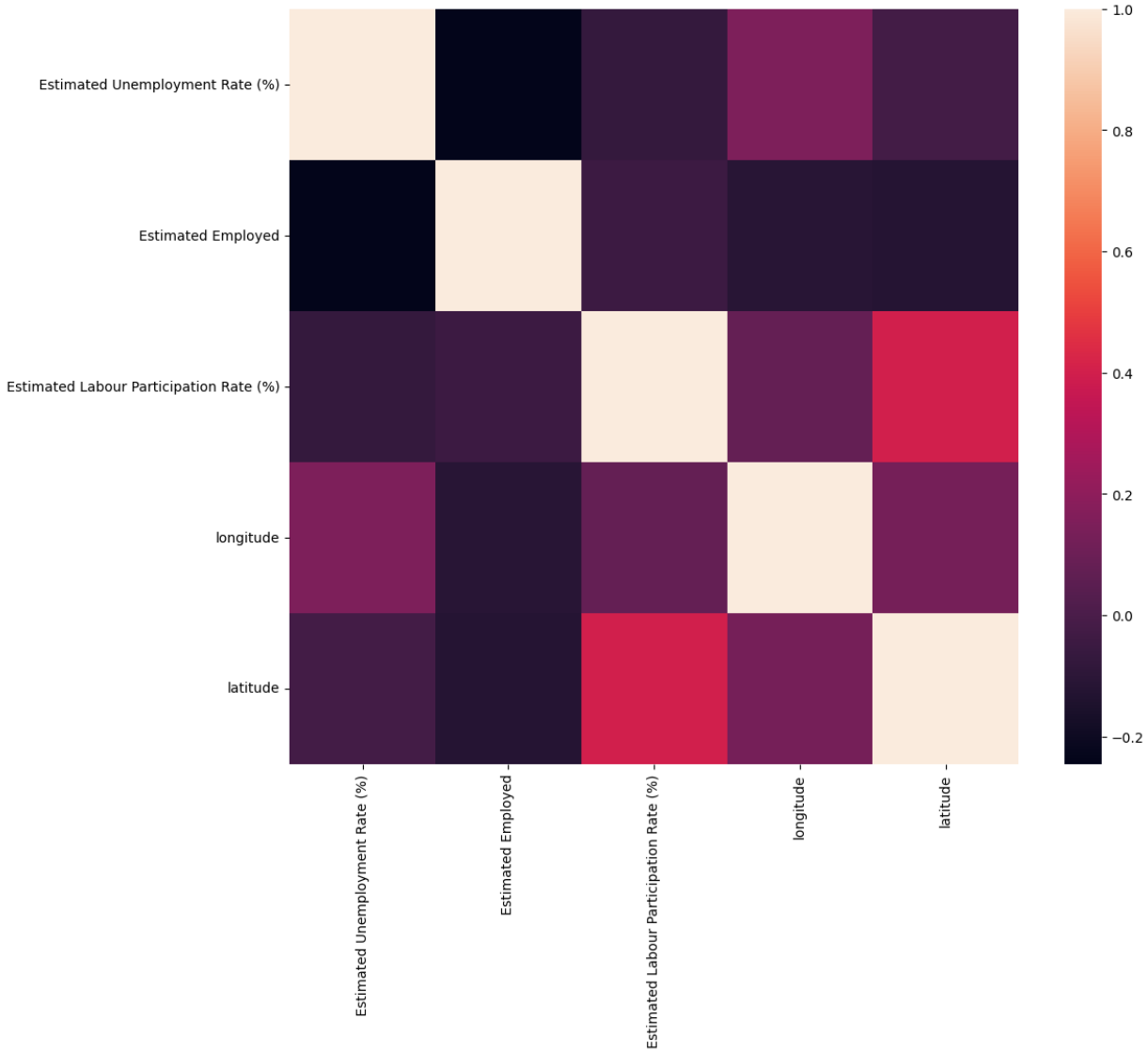


In [22]:

```
plt.figure(figsize=(12,10))
sns.heatmap(df.corr())
plt.show()
```

```
C:\Users\md naiyer azam\AppData\Local\Temp\ipykernel_31260\4186801998.py:2: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will def
ault to False. Select only valid columns or specify the value of numeric_only to silence this warni
ng.
```



Thank You...!😻