

# oyment-analysis-with-python-oasis

March 4, 2024

## 1 UNEMPLOYMENT ANALYSIS WITH PYTHON (OASIS)

Problem Statement:

Unemployment is measured by the unemployment rate which is the number of people who are unemployed as a percentage of the total labour force. We have seen a sharp increase in the unemployment rate during Covid-19, so analyzing the unemployment rate can be a good data science project.

```
[1]: pip install folium
```

```
Requirement already satisfied: folium in
c:\users\student_0002\anaconda3\lib\site-packages (0.15.1)
Requirement already satisfied: requests in
c:\users\student_0002\anaconda3\lib\site-packages (from folium) (2.28.1)
Requirement already satisfied: Jinja2>=2.9 in
c:\users\student_0002\anaconda3\lib\site-packages (from folium) (3.1.2)
Requirement already satisfied: branca>=0.6.0 in
c:\users\student_0002\anaconda3\lib\site-packages (from folium) (0.7.1)
Requirement already satisfied: xyzservices in
c:\users\student_0002\anaconda3\lib\site-packages (from folium) (2023.10.1)
Requirement already satisfied: numpy in
c:\users\student_0002\anaconda3\lib\site-packages (from folium) (1.23.5)
Requirement already satisfied: MarkupSafe>=2.0 in
c:\users\student_0002\anaconda3\lib\site-packages (from Jinja2>=2.9->folium)
(2.1.1)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\student_0002\anaconda3\lib\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
c:\users\student_0002\anaconda3\lib\site-packages (from requests->folium)
(1.26.14)
Requirement already satisfied: charset-normalizer<3,>=2 in
c:\users\student_0002\anaconda3\lib\site-packages (from requests->folium)
(2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\student_0002\anaconda3\lib\site-packages (from requests->folium)
(2022.12.7)
```

Note: you may need to restart the kernel to use updated packages.

```
[2]: pip install geopy
```

```
Requirement already satisfied: geopy in  
c:\users\student_0002\anaconda3\lib\site-packages (2.4.1)  
Requirement already satisfied: geographiclib<3,>=1.52 in  
c:\users\student_0002\anaconda3\lib\site-packages (from geopy) (2.0)  
Note: you may need to restart the kernel to use updated packages.
```

```
[3]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import plotly.express as px  
  
import folium  
from folium import Circle  
from geopy import Nominatim  
  
from sklearn.preprocessing import MinMaxScaler  
from IPython.display import display
```

```
[4]: Data = pd.read_csv('Unemployment_Rate_upto_11_2020.csv')
```

```
[5]: Data.head(10)
```

```
[5]:
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	\
0	Andhra Pradesh	31-01-2020	M	5.48	
1	Andhra Pradesh	29-02-2020	M	5.83	
2	Andhra Pradesh	31-03-2020	M	5.79	
3	Andhra Pradesh	30-04-2020	M	20.51	
4	Andhra Pradesh	31-05-2020	M	17.43	
5	Andhra Pradesh	30-06-2020	M	3.31	
6	Andhra Pradesh	31-07-2020	M	8.34	
7	Andhra Pradesh	31-08-2020	M	6.96	
8	Andhra Pradesh	30-09-2020	M	6.40	
9	Andhra Pradesh	31-10-2020	M	6.59	

	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	\
0	16635535	41.02	South	
1	16545652	40.90	South	
2	15881197	39.18	South	
3	11336911	33.10	South	
4	12988845	36.46	South	
5	19805400	47.41	South	
6	15431615	38.91	South	
7	15251776	37.83	South	
8	15220312	37.47	South	

9 15157557 37.34 South

	longitude	latitude
0	15.9129	79.74
1	15.9129	79.74
2	15.9129	79.74
3	15.9129	79.74
4	15.9129	79.74
5	15.9129	79.74
6	15.9129	79.74
7	15.9129	79.74
8	15.9129	79.74
9	15.9129	79.74

[6]: Data.info

[6]: <bound method DataFrame.info of

			Region	Date	Frequency
0	Andhra Pradesh	31-01-2020	M		5.48
1	Andhra Pradesh	29-02-2020	M		5.83
2	Andhra Pradesh	31-03-2020	M		5.79
3	Andhra Pradesh	30-04-2020	M		20.51
4	Andhra Pradesh	31-05-2020	M		17.43
..	...	...	...	...	...
262	West Bengal	30-06-2020	M		7.29
263	West Bengal	31-07-2020	M		6.83
264	West Bengal	31-08-2020	M		14.87
265	West Bengal	30-09-2020	M		9.35
266	West Bengal	31-10-2020	M		9.98

	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1
0	16635535	41.02	South
1	16545652	40.90	South
2	15881197	39.18	South
3	11336911	33.10	South
4	12988845	36.46	South
..	...	...	...
262	30726310	40.39	East
263	35372506	46.17	East
264	33298644	47.48	East
265	35707239	47.73	East
266	33962549	45.63	East

	longitude	latitude
0	15.9129	79.740
1	15.9129	79.740
2	15.9129	79.740

```

3      15.9129    79.740
4      15.9129    79.740
..      ...      ...
262    22.9868    87.855
263    22.9868    87.855
264    22.9868    87.855
265    22.9868    87.855
266    22.9868    87.855

```

[267 rows x 9 columns]>

```
[7]: Data.columns
```

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

```
[8]: Data.describe()
```

```
[8]:
```

	Estimated Unemployment Rate (%)	Estimated Employed \		
count	267.000000	2.670000e+02		
mean	12.236929	1.396211e+07		
std	10.803283	1.336632e+07		
min	0.500000	1.175420e+05		
25%	4.845000	2.838930e+06		
50%	9.650000	9.732417e+06		
75%	16.755000	2.187869e+07		
max	75.850000	5.943376e+07		

	Estimated Labour Participation Rate (%)	longitude	latitude
count	267.000000	267.000000	267.000000
mean	41.681573	22.826048	80.532425
std	7.845419	6.270731	5.831738
min	16.770000	10.850500	71.192400
25%	37.265000	18.112400	76.085600
50%	40.390000	23.610200	79.019300
75%	44.055000	27.278400	85.279900
max	69.690000	33.778200	92.937600

```
[9]: Data.isnull().sum
```

```
[9]: <bound method NDFrame._add_numeric_operations.<locals>.sum of
```

	Region	Date
Frequency	Estimated Unemployment Rate (%) \	
0	False False False	False
1	False False False	False
2	False False False	False

3	False	False	False	False
4	False	False	False	False
..	...	...	...	...
262	False	False	False	False
263	False	False	False	False
264	False	False	False	False
265	False	False	False	False
266	False	False	False	False

	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1 \
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
..	...	...	...
262	False	False	False
263	False	False	False
264	False	False	False
265	False	False	False
266	False	False	False

	longitude	latitude
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
..	...	...
262	False	False
263	False	False
264	False	False
265	False	False
266	False	False

[267 rows x 9 columns]>

```
[10]: Data[Data.duplicated()]
```

```
[10]: Empty DataFrame
Columns: [Region, Date, Frequency, Estimated Unemployment Rate (%),
Estimated Employed, Estimated Labour Participation Rate (%), Region.1,
longitude, latitude]
Index: []
```

```
[11]: Data.drop_duplicates(inplace=True)
```

```
[12]: Data.isnull().sum().sum()
```

```
[12]: 0
```

```
[13]: Data.isnull().sum()
```

```
[13]: 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
```

```
[14]: sns.pairplot(Data)
```

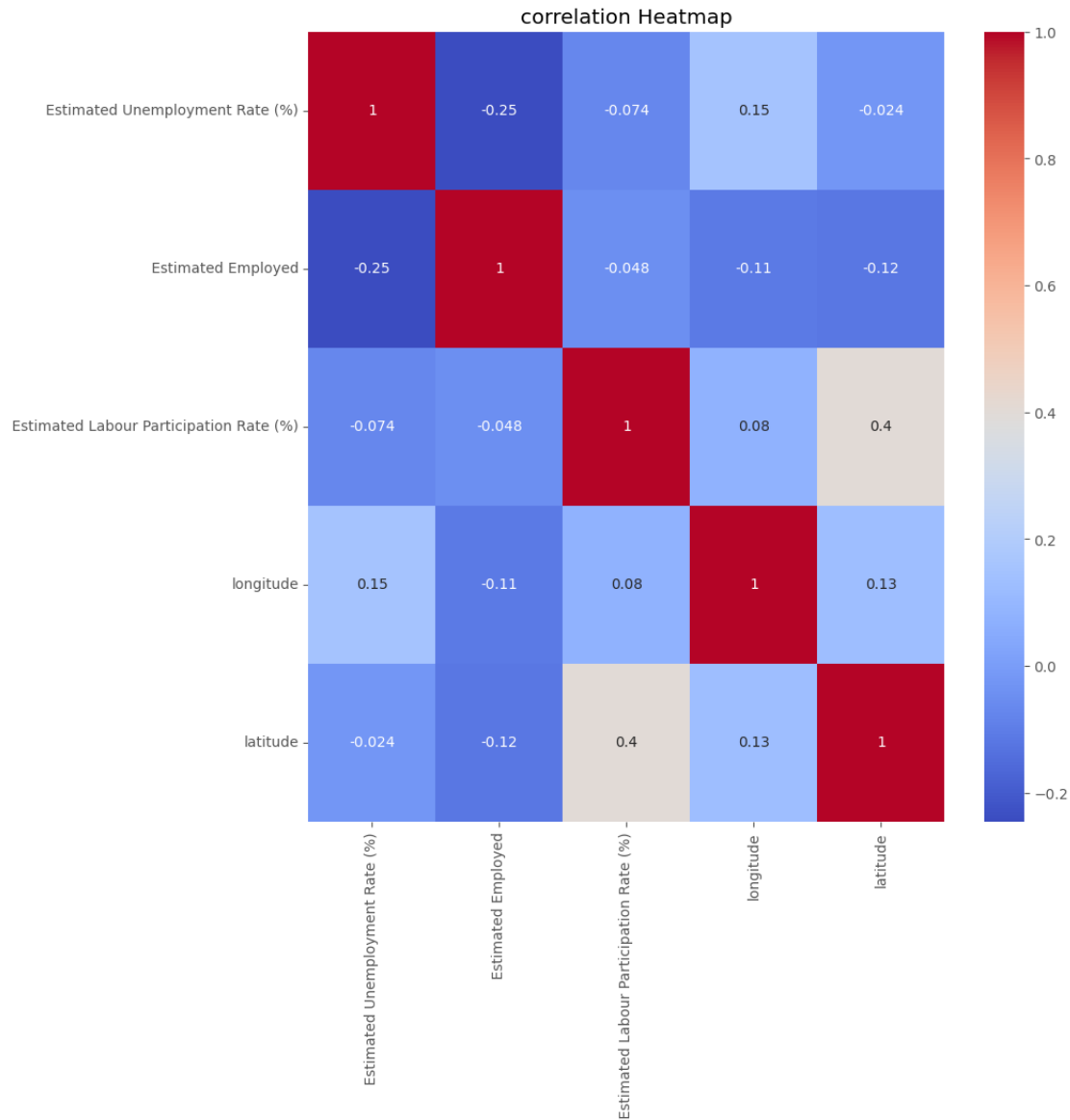
```
[14]: <seaborn.axisgrid.PairGrid at 0x22e449a3f10>
```



```
[15]: numeric_data = Data.select_dtypes(include='number')

plt.style.use("ggplot")

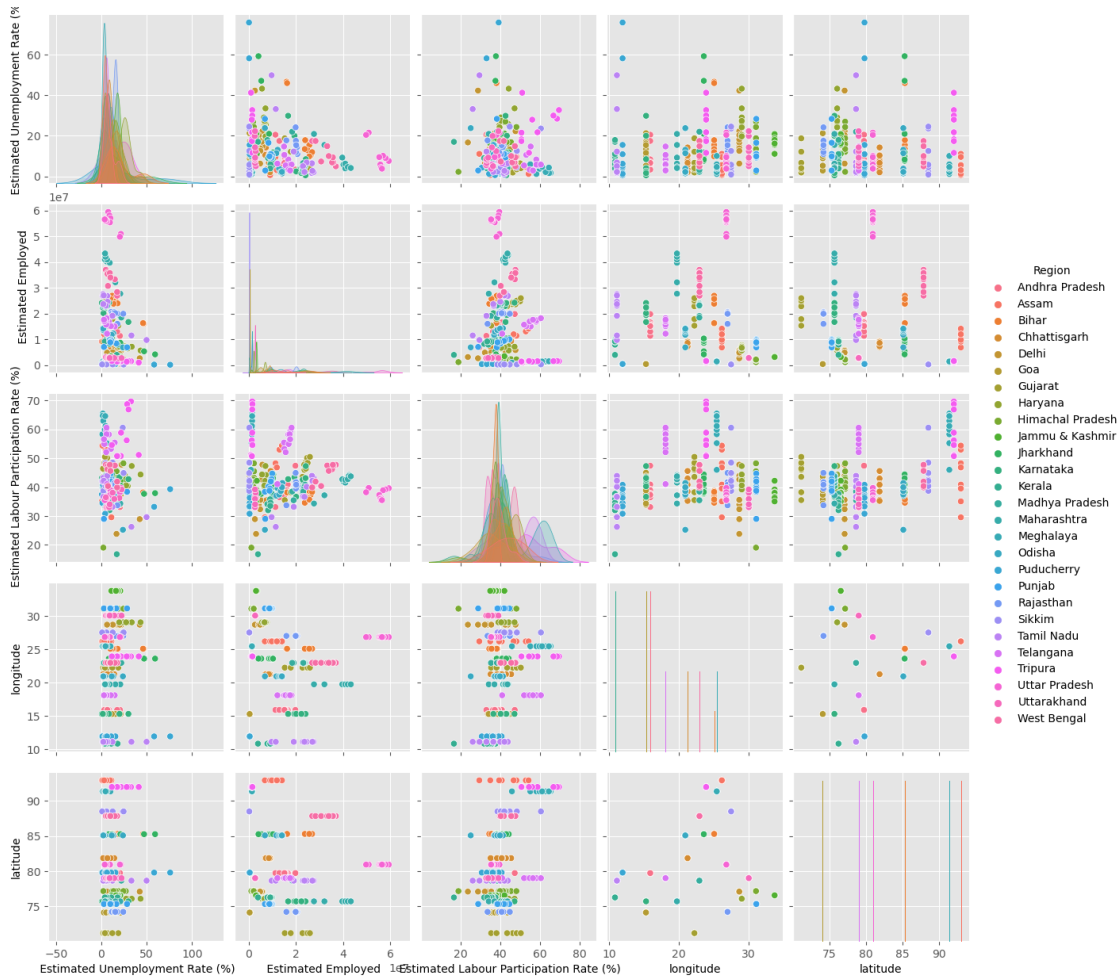
plt.figure(figsize=(10, 10))
sns.heatmap(numeric_data.corr(), annot=True, cmap="coolwarm")
plt.title("correlation Heatmap")
plt.show()
```



```
[16]: sns.pairplot(data = Data , hue = 'Region')
```

```
[16]: <seaborn.axisgrid.PairGrid at 0x22e463c7d00>
```



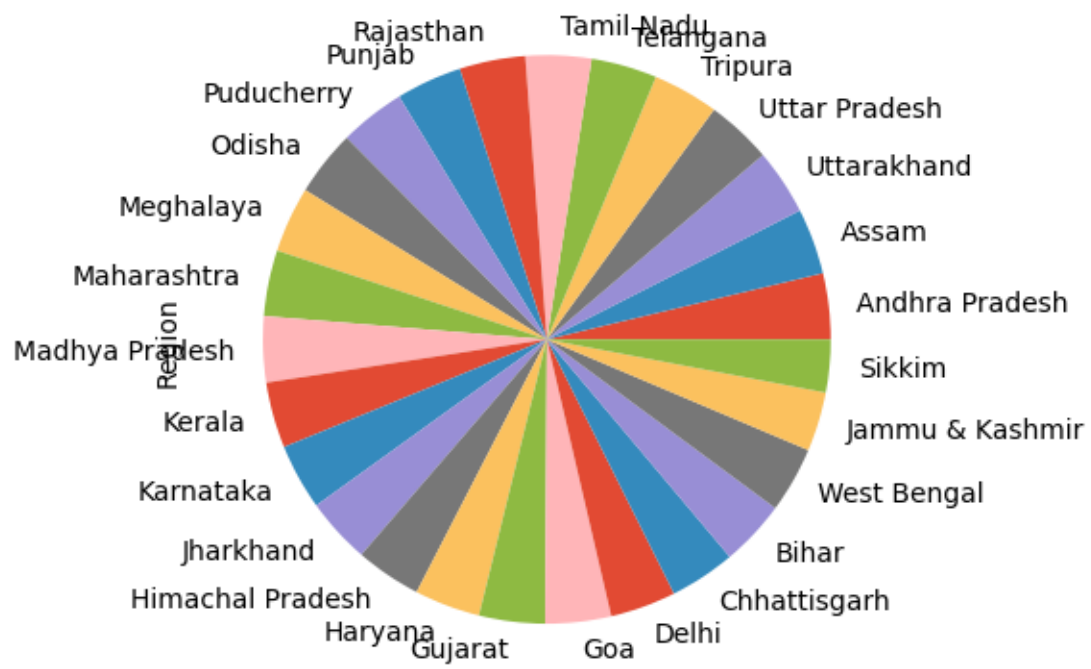


```
[17]: Data.Region.unique()
```

```
[17]: 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)
```

```
[18]: Data['Region'].value_counts().plot.pie()
```

```
[18]: <Axes: ylabel='Region'>
```



```
[21]: Data['Region'].value_counts().plot.bar()
```

```
[21]: <Axes: >
```

