

```
In [1]: import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

import calendar
```

```
In [2]: import datetime as dt

import plotly.io as pio
pio.templates
```

```
Out[2]: Templates configuration
-----
Default template: 'plotly'
Available templates:
['ggplot2', 'seaborn', 'simple_white', 'plotly',
'plotly_white', 'plotly_dark', 'presentation', 'xgridoff',
'ygridoff', 'gridon', 'none']
```

```
In [3]: import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from IPython.display import HTML
```

```
In [5]: df = pd.read_csv('Unemployment_Rate_upto_11_2020.csv')
```

```
In [6]: df.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

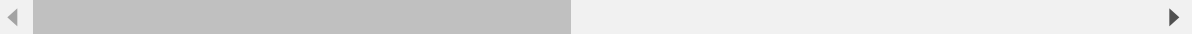
In [7]: df.info()

```
<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
```

In [8]: df.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
```

In [9]: df.columns = ['States', 'Date', 'Frequency', 'Estimated Unemployment Rate', 'Estima



In [10]: df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)

In [11]: df['Frequency'] = df['Frequency'].astype('category')

In [12]: df['Month'] = df['Date'].dt.month

In [14]: df['Month_int'] = df['Month'].apply(lambda x : int(x))

In [15]: df['Month_name'] = df['Month_int'].apply(lambda x: calendar.month_abbr[x])

In [16]: df['Region'] = df['Region'].astype('category')

```
In [17]: df.drop(columns='Month',inplace=True)
df.head(3)
```

Out[17]:

	States	Date	Frequency	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Region	longitude	latitude
0	Andhra Pradesh	2020-01-31	M	5.48	16635535	41.02	South	15.9129	79.1
1	Andhra Pradesh	2020-02-29	M	5.83	16545652	40.90	South	15.9129	79.1
2	Andhra Pradesh	2020-03-31	M	5.79	15881197	39.18	South	15.9129	79.1

```
In [18]: df_stats = df[['Estimated Unemployment Rate',
                        'Estimated Employed', 'Estimated Labour Participation Rate']]

round(df_stats.describe().T,2)
```

Out[18]:

	count	mean	std	min	25%	50%	75%
Estimated Unemployment Rate	267.0	12.24	10.80	0.50	4.84	9.65	16.76
Estimated Employed	267.0	13962105.72	13366318.36	117542.00	2838930.50	9732417.00	21878686.00
Estimated Labour Participation Rate	267.0	41.68	7.85	16.77	37.26	40.39	44.06

```
In [19]: region_stats = df.groupby(['Region'])[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate']]

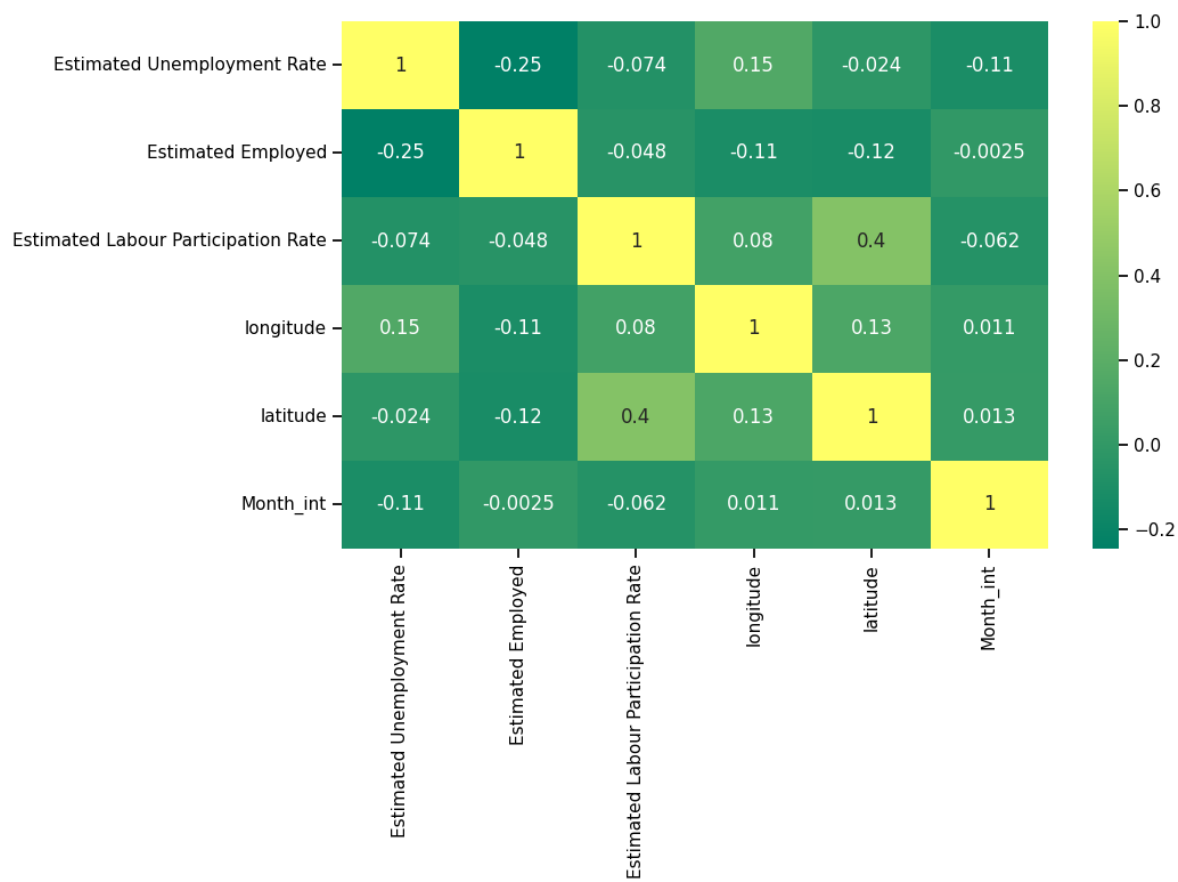
region_stats = round(region_stats,2)

region_stats
```

Out[19]:

	Region	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate
0	East	13.92	19602366.90	40.11
1	North	15.89	13072487.92	38.70
2	Northeast	10.95	3617105.53	52.06
3	South	10.45	14040589.33	40.44
4	West	8.24	18623512.72	41.26

```
In [20]: heat_maps = df[['Estimated Unemployment Rate',  
                        'Estimated Employed', 'Estimated Labour Participation Rate',  
                        'longitude', 'latitude', 'Month_int']]  
  
heat_maps = heat_maps.corr()  
  
plt.figure(figsize=(10,6))  
sns.set_context('notebook',font_scale=1)  
sns.heatmap(heat_maps, annot=True,cmap='summer');
```



```
In [21]: fig = px.box(df,x='States',y='Estimated Unemployment Rate',color='States',title='Estimated Unemployment Rate by State')
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```

```
In [22]: fig = px.scatter_matrix(df,template='plotly',  
    dimensions=['Estimated Unemployment Rate','Estimated Employed',  
    'Estimated Labour Participation Rate'],  
    color='Region')  
fig.show()
```

```
In [23]: plot_ump = df[['Estimated Unemployment Rate', 'States']]

df_unemp = plot_ump.groupby('States').mean().reset_index()

df_unemp = df_unemp.sort_values('Estimated Unemployment Rate')

fig = px.bar(df_unemp, x='States', y='Estimated Unemployment Rate', color='State',
              title='Average Unemployment Rate in each state', template='plotly')

fig.show()
```

```
In [24]: fig = px.bar(df, x='Region',y='Estimated Unemployment Rate',animation_frame =  
          title='Unemployment rate across region from Jan.2020 to Oct.2020',  
  
          fig.update_layout(xaxis={'categoryorder':'total descending'})  
  
          fig.layout.updatemenus[0].buttons[0].args[1]["frame"]["duration"] = 2000  
  
          fig.show()
```



```
In [25]: unemplo_df = df[['States', 'Region', 'Estimated Unemployment Rate', 'Estimated Em  
unemplo = unemplo_df.groupby(['Region', 'States'])['Estimated Unemployment Rate
```

```
In [26]: fig = px.sunburst(unemplo, path=['Region', 'States'], values='Estimated Unemplo  
        color_continuous_scale='Plasma', title= 'unemployment rate in  
        height=650, template='ggplot2')  
  
fig.show()
```

```
In [27]: fig = px.scatter_geo(df, 'longitude', 'latitude', color="Region",
                             hover_name="States", size="Estimated Unemployment Rate",
                             animation_frame="Month_name", scope='asia', template='plot1

fig.layout.updatemenus[0].buttons[0].args[1]["frame"]["duration"] = 2000

fig.update_geos(lataxis_range=[5,35], lonaxis_range=[65, 100], oceancolor="#666666",
                showocean=True)

fig.show()
```

```
In [28]: lock = df[(df['Month_int'] >= 4) & (df['Month_int'] <=7)]

bf_lock = df[(df['Month_int'] >= 1) & (df['Month_int'] <=4)]
```

```
In [29]: g_lock = lock.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
g_bf_lock = bf_lock.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()

g_lock['Unemployment Rate before lockdown'] = g_bf_lock['Estimated Unemployment Rate before lockdown']
g_lock.columns = ['States', 'Unemployment Rate after lockdown', 'Unemployment Rate before lockdown']
g_lock.head(2)
```

Out[29]:

	States	Unemployment Rate after lockdown	Unemployment Rate before lockdown
0	Andhra Pradesh	12.3975	9.4025
1	Assam	6.2450	6.2250

```
In [30]: g_lock['percentage change in unemployment'] = round(g_lock['Unemployment Rate after lockdown'] - g_lock['Unemployment Rate before lockdown'], 2)
```

```
In [31]: plot_per = g_lock.sort_values('percentage change in unemployment')
```

```
In [32]: fig = px.bar(plot_per, x='States',y='percentage change in unemployment',color=
          title='percentage change in Unemployment in each state after lockd
fig.show()
```

```
In [33]: def sort_impact(x):
          if x <= 10:
              return 'impacted States'
          elif x <= 20:
              return 'hard impacted States'
          elif x <= 30:
              return 'harder impacted States'
          elif x <= 40:
              return 'hardest impacted States'
          return x
```

```
In [34]: plot_per['impact status'] = plot_per['percentage change in unemployment'].appl
```

```
In [35]: fig = px.bar(plot_per, y='States',x='percentage change in unemployment',color=
          title='Impact of lockdown on employment across states',template='g

fig.show()
```

```
In [ ]:
```