In [4]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

import datetime as dt
import calendar
```

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

In [6]:
```
ds
```

Out[6]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	la
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	7
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	7
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	7
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	7
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	7
...	...	...	...	...	...	...	...	...	
262	West Bengal	30-06-2020	M	7.29	30726310	40.39	East	22.9868	8
263	West Bengal	31-07-2020	M	6.83	35372506	46.17	East	22.9868	8
264	West Bengal	31-08-2020	M	14.87	33298644	47.48	East	22.9868	8
265	West Bengal	30-09-2020	M	9.35	35707239	47.73	East	22.9868	8
266	West Bengal	31-10-2020	M	9.98	33962549	45.63	East	22.9868	8

267 rows × 9 columns

In [7]:
```
ds.columns=["state","date","frequency","estimated unemployment rate","estimate
```

In [8]:
```
ds.head()
```

Out[8]:

	state	date	frequency	estimated unemployment rate	estimated employed	estimated labour participation rate	region	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 [9]:
```
ds.shape
```

Out[9]:
```
(267, 9)
```

In [10]:
```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 267 entries, 0 to 266
Data columns (total 9 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   state                                 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                                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 [11]:
```
round(ds.describe().T)
```

Out[11]:

	count	mean	std	min	25%	50%	75%	
estimated unemployment rate	267.0	12.0	11.0	0.0	5.0	10.0	17.0	
estimated employed	267.0	13962106.0	13366318.0	117542.0	2838930.0	9732417.0	21878686.0	5943
estimated labour participation rate	267.0	42.0	8.0	17.0	37.0	40.0	44.0	
longitude	267.0	23.0	6.0	11.0	18.0	24.0	27.0	
latitude	267.0	81.0	6.0	71.0	76.0	79.0	85.0	

In [12]:
```
ds.isnull().sum()
```

Out[12]:
```
state                                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 [13]:
```
ds.state.value_counts()
```

Out[13]:
```
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: state, dtype: int64
```

In [14]:
```
ds['date'] = pd.to_datetime(ds['date'], dayfirst=True)
ds['month_int'] =  ds['date'].dt.month
ds['month'] =  ds['month_int'].apply(lambda x: calendar.month_abbr[x])
ds.head()
```

Out[14]:

	state	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	79.74
1	Andhra Pradesh	2020-02-29	M	5.83	16545652	40.90	South	15.9129	79.74
2	Andhra Pradesh	2020-03-31	M	5.79	15881197	39.18	South	15.9129	79.74
3	Andhra Pradesh	2020-04-30	M	20.51	11336911	33.10	South	15.9129	79.74
4	Andhra Pradesh	2020-05-31	M	17.43	12988845	36.46	South	15.9129	79.74

In [16]:
```
IND =  ds.groupby(["month"])[['estimated unemployment rate', "estimated employ
IND = pd.DataFrame(IND).reset_index()
```

In [17]:
```
month = IND.month
unemployment_rate = IND["estimated unemployment rate"]
labour_participation_rate = IND["estimated labour participation rate"]

fig = go.Figure()

fig.add_trace(go.Bar(x = month, y = unemployment_rate, name= "Unemployment Rat
fig.add_trace(go.Bar(x = month, y = labour_participation_rate, name= "Labour P

fig.update_layout(title="Uneployment Rate and Labour Participation Rate",
                  xaxis={"categoryorder":"array", "categoryarray":["Jan", "Feb

fig.show()
```

In [18]:
```
fig = px.bar(IND, x='month',y='estimated employed', color='month',
             category_orders = {"month": ["Jan", "Feb", "Mar", "Apr", "May", "
             title='estimated employed people from Jan 2020 to Oct 2020')

fig.show()
```

In [19]:
```
state = ds.groupby(["state"])[["estimated unemployment rate", "estimated emplo
state = pd.DataFrame(state).reset_index()
```

In [21]:
```
fig = px.box(ds,x='state',y='estimated unemployment rate',color='state',title=
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```

In [22]:

```
fig = px.bar(ds, x='state',y='estimated unemployment rate', animation_frame =
            title='Unemployment rate from Jan 2020 to Oct 2020 (State)')

fig.update_layout(xaxis={'categoryorder':'total descending'})

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

fig.show()
```

In [24]:
```
fig = px.scatter_geo(ds,'longitude', 'latitude', color="state",
                     hover_name="state", size="estimated unemployment rate",
                     animation_frame="month",scope='asia',title='Impact of loc

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

fig.update_geos(lataxis_range=[5,40], lonaxis_range=[65, 100],oceancolor="ligh
    showocean=True)

fig.show()
```

In [25]:
```
ds.region.unique()
```

Out[25]:
```
array(['South', 'Northeast', 'East', 'West', 'North'], dtype=object)
```

In [26]:
```
region = ds.groupby(["region"])[['estimated unemployment rate', "estimated emp
region = pd.DataFrame(region).reset_index()
```

In [27]:
```
fig = px.scatter_matrix(ds, dimensions=['estimated unemployment rate','estimat
fig.show()
```

```
C:\Users\zua20\anaconda3\lib\site-packages\plotly\express\_core.py:279: Futur
eWarning:

iteritems is deprecated and will be removed in a future version. Use .items i
nstead.
```

In [28]:
```
fig = px.bar(region, x="region", y="estimated unemployment rate", color="regic
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```

In [29]:
```
fig = px.bar(ds, x='region',y='estimated unemployment rate', animation_frame =
            title='Unemployment rate 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 [30]:
```
unemployment = ds.groupby(['region','state'])['estimated unemployment rate'].m

unemployment.head()
```

Out[30]:

	region	state	estimated unemployment rate
0	East	Bihar	19.471
1	East	Jharkhand	19.539
2	East	Odisha	6.462
3	East	West Bengal	10.192
4	North	Delhi	18.414

In [31]:
```
fig = px.sunburst(unemployment, path=['region','state'], values='estimated une
                  title= 'Unemployment rate in every State and Region', height
fig.show()
```

In [32]:
```
before_lockdown = ds[(ds['month_int'] >= 1) & (ds['month_int'] <4)]
after_lockdown = ds[(ds['month_int'] >= 4) & (ds['month_int'] <=6)]
```

In [33]:
```
af_lockdown = after_lockdown.groupby('state')['estimated unemployment rate'].m

lockdown = before_lockdown.groupby('state')['estimated unemployment rate'].mea
lockdown['unemployment rate before lockdown'] = af_lockdown['estimated unemplo

lockdown.columns = ['state','unemployment rate before lockdown','unemployment
lockdown.head()
```

Out[33]:

	state	unemployment rate before lockdown	unemployment rate after lockdown
0	Andhra Pradesh	5.700000	13.750000
1	Assam	4.613333	7.070000
2	Bihar	12.110000	36.806667
3	Chhattisgarh	8.523333	9.380000
4	Delhi	18.036667	25.713333

In [34]:
```
lockdown['rate change in unemployment'] = round(lockdown['unemployment rate be
                                                /lockdown['unemployment rate a
```

In [36]:
```
fig = px.bar(lockdown, x='state',y='rate change in unemployment',color='rate c
             title='Percentage change in Unemployment rate in each state after
fig.update_layout(xaxis={'categoryorder':'total ascending'})
fig.show()
```