```
#Importing necessary libraries
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

import warnings

# Ignore all warnings
warnings.filterwarnings("ignore")


#Importing and Reading the Unemployment in India dataset
df = pd.read_csv("/content/Unemployment in India.csv")
df.head(5)
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139.0	43.24	Rural
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881.0	42.05	Rural

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   Region                                   740 non-null    object
 1    Date                                    740 non-null    object
 2    Frequency                               740 non-null    object
 3    Estimated Unemployment Rate (%)         740 non-null    float64
 4    Estimated Employed                      740 non-null    float64
 5    Estimated Labour Participation Rate (%) 740 non-null    float64
 6   Area                                     740 non-null    object
dtypes: float64(3), object(4)
memory usage: 42.1+ KB
```

```
#Printing the shape of the dataset
df.shape
```

```
(768, 7)
```

```
df.describe()
```

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
count	740.000000	7.400000e+02	740.000000
mean	11.787946	7.204460e+06	42.630122
std	10.721298	8.087988e+06	8.111094
min	0.000000	4.942000e+04	13.330000
25%	4.657500	1.190404e+06	38.062500
50%	8.350000	4.744178e+06	41.160000
75%	15.887500	1.127549e+07	45.505000
max	76.740000	4.577751e+07	72.570000

```
# Check for duplicate rows
duplicates = df.duplicated()
sum_duplicates = duplicates.sum()
print(f"Number of duplicate rows: {sum_duplicates}")

duplicate_rows = df[duplicates]
print("\nDuplicate Rows:")
print(duplicate_rows)

df.drop_duplicates(inplace=True)
```

```
Duplicate Rows:
```

```
361   NaN   NaN        NaN                                   NaN
362   NaN   NaN        NaN                                   NaN
363   NaN   NaN        NaN                                   NaN
364   NaN   NaN        NaN                                   NaN
365   NaN   NaN        NaN                                   NaN
366   NaN   NaN        NaN                                   NaN
367   NaN   NaN        NaN                                   NaN
368   NaN   NaN        NaN                                   NaN
369   NaN   NaN        NaN                                   NaN
370   NaN   NaN        NaN                                   NaN
371   NaN   NaN        NaN                                   NaN
372   NaN   NaN        NaN                                   NaN
754   NaN   NaN        NaN                                   NaN
755   NaN   NaN        NaN                                   NaN
756   NaN   NaN        NaN                                   NaN
757   NaN   NaN        NaN                                   NaN
758   NaN   NaN        NaN                                   NaN
759   NaN   NaN        NaN                                   NaN
760   NaN   NaN        NaN                                   NaN
761   NaN   NaN        NaN                                   NaN
762   NaN   NaN        NaN                                   NaN
763   NaN   NaN        NaN                                   NaN
764   NaN   NaN        NaN                                   NaN
765   NaN   NaN        NaN                                   NaN
766   NaN   NaN        NaN                                   NaN
767   NaN   NaN        NaN                                   NaN

      Estimated Employed   Estimated Labour Participation Rate (%) Area
360                  NaN                                       NaN  NaN
361                  NaN                                       NaN  NaN
362                  NaN                                       NaN  NaN
363                  NaN                                       NaN  NaN
364                  NaN                                       NaN  NaN
365                  NaN                                       NaN  NaN
366                  NaN                                       NaN  NaN
367                  NaN                                       NaN  NaN
368                  NaN                                       NaN  NaN
369                  NaN                                       NaN  NaN
370                  NaN                                       NaN  NaN
371                  NaN                                       NaN  NaN
372                  NaN                                       NaN  NaN
754                  NaN                                       NaN  NaN
755                  NaN                                       NaN  NaN
756                  NaN                                       NaN  NaN
757                  NaN                                       NaN  NaN
758                  NaN                                       NaN  NaN
759                  NaN                                       NaN  NaN
760                  NaN                                       NaN  NaN
761                  NaN                                       NaN  NaN
762                  NaN                                       NaN  NaN
763                  NaN                                       NaN  NaN
764                  NaN                                       NaN  NaN
765                  NaN                                       NaN  NaN
766                  NaN                                       NaN  NaN
767                  NaN                                       NaN  NaN
```

```
#Checking for null values
null_values = df.isnull()
null_counts = null_values.sum()

print("Null Value Counts:")
print(null_counts)
```

```
Null Value Counts:
Region                                    1
 Date                                     1
 Frequency                                1
 Estimated Unemployment Rate (%)          1
 Estimated Employed                       1
 Estimated Labour Participation Rate (%)  1
Area                                      1
dtype: int64
```

```
#Dropping the null values
df = df.dropna()
```

```
df
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139.0	43.24	Rural
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881.0	42.05	Rural
	Andhra	31-					

```
#Printing the columns of the dataset
print(df.columns)
```

```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Area'],
      dtype='object')
```

Pradesh 2019

```
df.rename(columns={' Date': 'Date', ' Frequency': 'Frequency', ' Estimated Unemployment Rate (%)': 'Estimated Unemployment Rate (%)'}, inplace=Tru
```

```
df.rename(columns={' Estimated Employed': 'Estimated Employed', ' Estimated Labour Participation Rate (%)': 'Estimated Labour Participation Rate (
```

```
#Counting the number of records in each region of the dataset
df.Region.value_counts()
```

```
Andhra Pradesh      28
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: Region, dtype: int64
```

```
df['Date'] = pd.to_datetime(df['Date'],dayfirst = True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 740 entries, 0 to 753
Data columns (total 7 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   Region                                   740 non-null    object
 1   Date                                     740 non-null    datetime64[ns]
 2   Frequency                                740 non-null    object
 3   Estimated Unemployment Rate (%)          740 non-null    float64
 4   Estimated Employed                       740 non-null    float64
 5   Estimated Labour Participation Rate (%)  740 non-null    float64
 6   Area                                     740 non-null    object
dtypes: datetime64[ns](1), float64(3), object(3)
memory usage: 46.2+ KB
```

```
df.head()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
	Andhra	2019-					

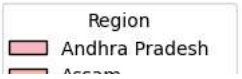
```
#Printing the month number corresponding to each date
df['Month_int'] = df['Date'].dt.month
df.head()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	Month_int
0	Andhra Pradesh	2019-05-31	Monthly	3.65	11999139.0	43.24	Rural	5
1	Andhra Pradesh	2019-06-30	Monthly	3.05	11755881.0	42.05	Rural	6

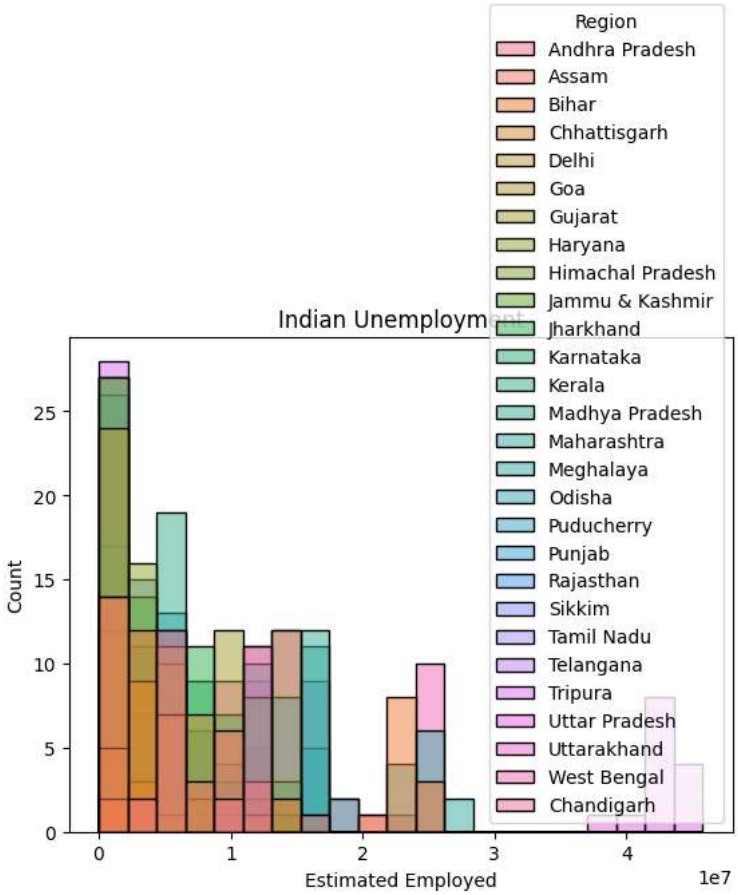
```
#Displaying the month
import calendar

df['Month'] = df['Month_int'].apply(lambda x: calendar.month_abbr[x])
df.head()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	Month_int
0	Andhra Pradesh	2019-05-31	Monthly	3.65	11999139.0	43.24	Rural	5
1	Andhra Pradesh	2019-06-30	Monthly	3.05	11755881.0	42.05	Rural	6
2	Andhra Pradesh	2019-07-31	Monthly	3.75	12086707.0	43.50	Rural	7
3	Andhra Pradesh	2019-08-31	Monthly	3.32	12285693.0	43.97	Rural	8

```
#Plotting Estimated Unemployment Rate (%) corresponding to each region
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Unemployment Rate (%)", hue="Region", data=df)
plt.show()
```




```
#Plotting Estimated Employed corresponding to each region
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Employed", hue="Region", data=df)
plt.show()
```




```
#Plotting Estimated Labour Participation Rate (%) corresponding to each region
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Labour Participation Rate (%)", hue="Region", data=df)
plt.show()
```


```
#Encoding the Frequency feature
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
df['Frequency']=label.fit_transform(df['Frequency'])
```

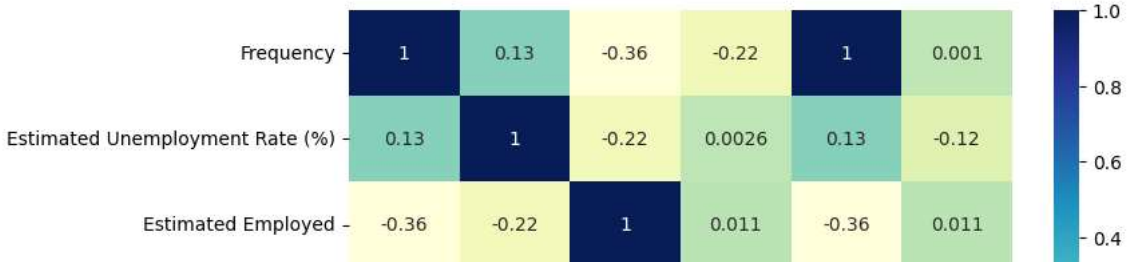
```
#Encoding the Area faeture
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
df['Area']=label.fit_transform(df['Area'])
```

```
df
```

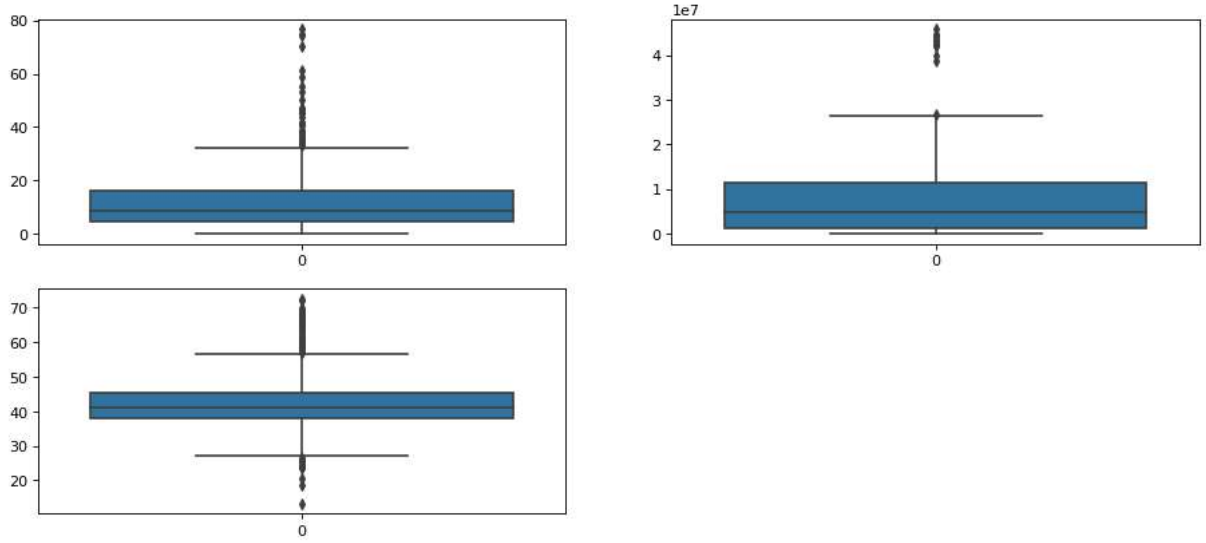
	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	Month_i
0	Andhra Pradesh	2019-05-31	0	3.65	11999139.0	43.24	0	
1	Andhra Pradesh	2019-06-30	0	3.05	11755881.0	42.05	0	
2	Andhra Pradesh	2019-07-31	0	3.75	12086707.0	43.50	0	
3	Andhra Pradesh	2019-08-31	0	3.32	12285693.0	43.97	0	
4	Andhra Pradesh	2019-09-30	0	5.17	12256762.0	44.68	0	
...	...	...	...	...	...	...	...	
749	West Bengal	2020-02-29	1	7.55	10871168.0	44.09	1	
750	West Bengal	2020-03-31	1	6.67	10806105.0	43.34	1	
751	West Bengal	2020-04-30	1	15.63	9299466.0	41.20	1	
752	West Bengal	2020-05-31	1	15.22	9240903.0	40.67	1	
753	West Bengal	2020-06-30	1	9.86	9088931.0	37.57	1	

```
#Showing the correlation matrix between all the numeric features
plt.figure(figsize = (8,5))
sns.heatmap(df.corr() , annot = True , cmap = "YlGnBu")
```

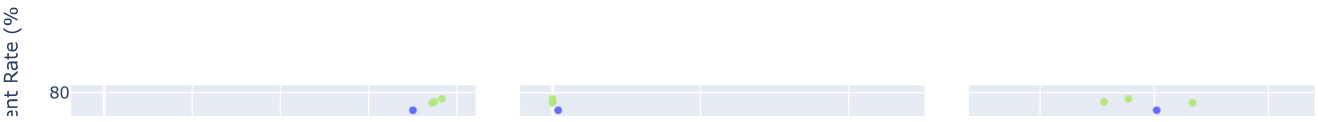
```
<Axes: >
```



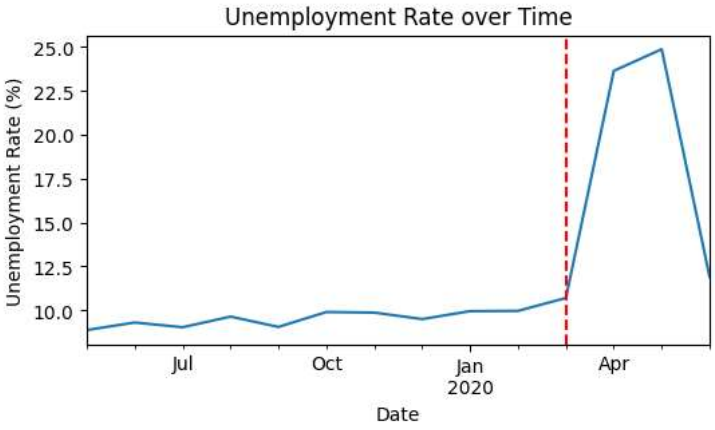

```
#Creating box-plots
plt.figure(figsize=(14, 6), dpi=80)

plt.subplot(221)
sns.boxplot(df['Estimated Unemployment Rate (%)'])

plt.subplot(222)
sns.boxplot(df['Estimated Employed'])

plt.subplot(223)
sns.boxplot(df['Estimated Labour Participation Rate (%)'])

plt.show()
```



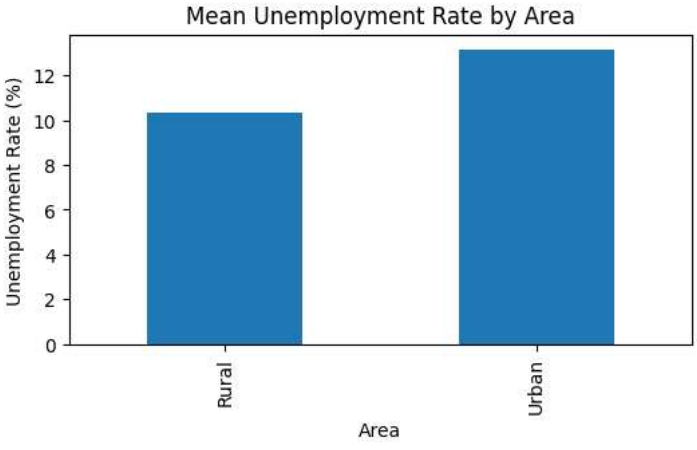

```
#Scatter matrix
plt.figure(figsize=(10, 5), dpi=80)
import plotly.express as px

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



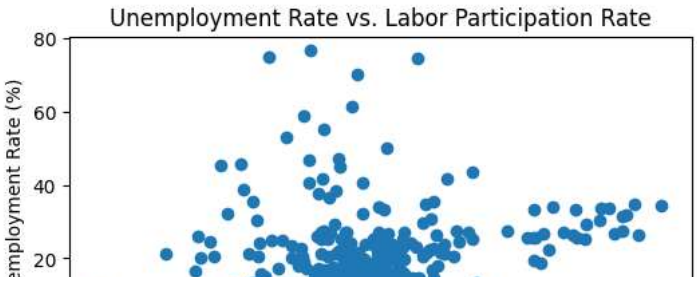

```
#Defining the start date of the lockdown period
lockdown_start = pd.to_datetime('2020-03-01')

#Creating a new column to indicate whether the date is before or after the lockdown
df['Lockdown'] = df['Date'] >= lockdown_start

#Grouping the dataframe by date and calculate the mean unemployment rate for each date
unemployment_by_date = df.groupby('Date')['Estimated Unemployment Rate (%)'].mean()

#Creating the line plot with a vertical line at the lockdown start date
fig, ax = plt.subplots(figsize=(6, 3))
unemployment_by_date.plot(ax=ax)
ax.axvline(lockdown_start, color='red', linestyle='--')
ax.set(title='Unemployment Rate over Time', xlabel='Date', ylabel='Unemployment Rate (%)')
plt.show()
```



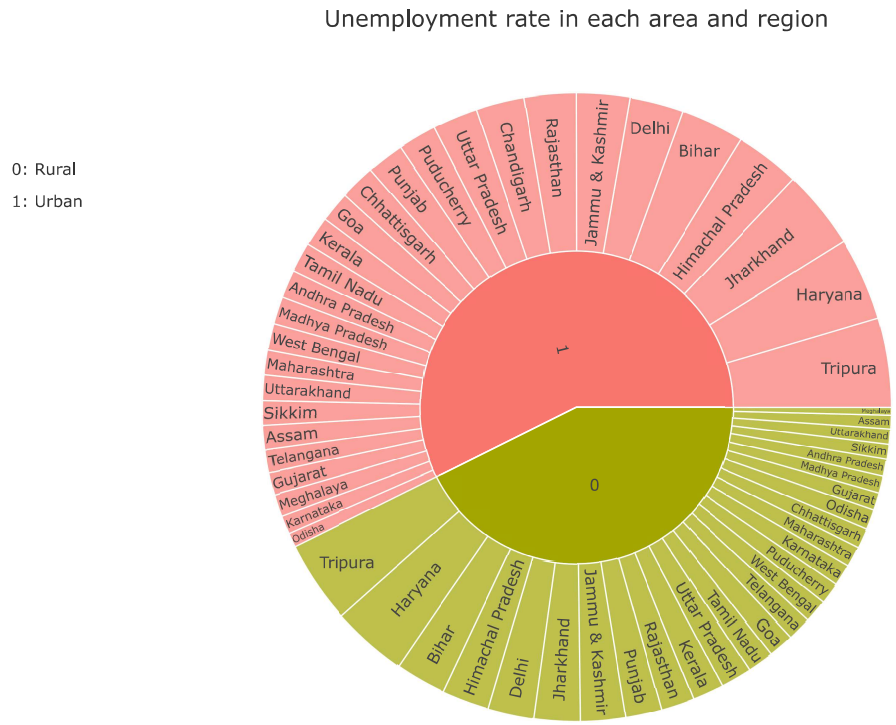

```
#Grouping the dataframe by area and calculate the mean unemployment rate for each area
unemployment_by_region = df.groupby('Area')['Estimated Unemployment Rate (%)'].mean()

#Creating a bar chart of the mean unemployment rate for each region
fig, ax = plt.subplots(figsize=(6, 3))
unemployment_by_region.plot(kind='bar', ax=ax)
ax.set(title='Mean Unemployment Rate by Area', xlabel='Area', ylabel='Unemployment Rate (%)')
#Adding custom labels to the bars
labels = {0: 'Rural', 1: 'Urban'}
ax.set_xticklabels([labels[int(x.get_text())] for x in ax.get_xticklabels()])
plt.show()
```




```
#Creating a scatter plot of the relationship between the unemployment rate and the estimated labor participation rate
fig, ax = plt.subplots(figsize=(6, 3))
ax.scatter(df['Estimated Labour Participation Rate (%)'], df['Estimated Unemployment Rate (%)'])
ax.set(title='Unemployment Rate vs. Labor Participation Rate', xlabel='Labor Participation Rate (%)', ylabel='Unemployment Rate (%)')
plt.show()
```




```
#Plotting Sunburst chart showing unemployment rate in each area and region

unemplo_df = df[['Region', 'Area', 'Estimated Unemployment Rate (%)', 'Estimated Employed', 'Estimated Labour Participation Rate (%)']]
unemplo = unemplo_df.groupby(['Area', 'Region'])['Estimated Unemployment Rate (%)'].mean().reset_index()

#Mapping numeric labels to custom labels
labels = {0: 'Rural', 1: 'Urban'}
unemplo['Area_Label'] = unemplo['Area'].map(labels)

fig = px.sunburst(unemplo, path=['Area', 'Region'], values='Estimated Unemployment Rate (%)',
                  color_continuous_scale='Plasma', title='Unemployment rate in each area and region',
                  height=650, template='ggplot2')
hover_data={'Area_Label': True, 'Region': True, 'Estimated Unemployment Rate (%)': ':.2f'}

#Adding a legend or side box
legend_text = [{'x': 0, 'y': 0.9, 'text': '0: Rural', 'showarrow': False},
               {'x': 0, 'y': 0.85, 'text': '1: Urban', 'showarrow': False}]

fig.update_layout(annotations=legend_text)
fig.show()
```



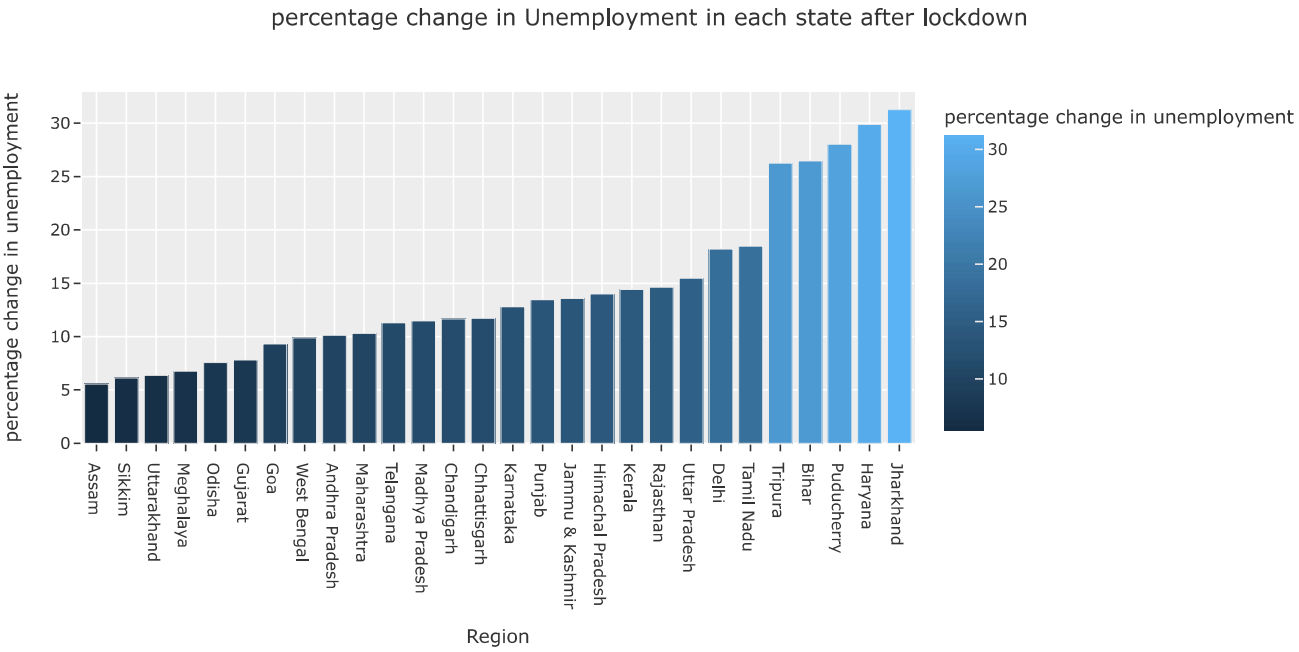

```
#data representation before and after lockdown

before_lockdown = df[(df['Month_int']>=1) &(df['Month_int'] <4)]
after_lockdown = df[(df['Month_int'] >=4) & (df['Month_int'] <=6)]


af_lockdown = after_lockdown.groupby('Region')['Estimated Unemployment Rate (%)'].mean().reset_index()

lockdown = before_lockdown.groupby('Region')['Estimated Unemployment Rate (%)'].mean().reset_index()
lockdown['unemployment rate before lockdown (%)'] = af_lockdown['Estimated Unemployment Rate (%)']

lockdown.columns = ['Region','unemployment rate before lockdown','unemployment rate after lockdown']
lockdown.head()
```

	Region	unemployment rate before lockdown	unemployment rate after lockdown
0	Andhra Pradesh	6.243333	11.126000
1	Assam	6.480000	6.563333
2	Bihar	14.276667	27.459000
3	Chandigarh	19.366667	12.656667

```
#percentage change in unemployment rate

lockdown['percentage change in unemployment'] = round(lockdown['unemployment rate after lockdown'] - lockdown['unemployment rate before lockdown']
plot = lockdown.sort_values('percentage change in unemployment')

print(plot)
```

```
             Region  unemployment rate before lockdown  \
1             Assam                           6.480000
21           Sikkim                          20.450000
26      Uttarakhand                           6.274000
16        Meghalaya                           3.323333
17           Odisha                           4.773333
7            Gujarat                          6.138333
6                Goa                          5.074000
27      West Bengal                           6.513333
0     Andhra Pradesh                          6.243333
15       Maharashtra                          5.295000
23         Telangana                          6.500000
14    Madhya Pradesh                          4.033333
3         Chandigarh                         19.366667
4        Chhattisgarh                         8.683333
12          Karnataka                          3.363333
19             Punjab                         11.041667
10    Jammu & Kashmir                         18.685000
9    Himachal Pradesh                         20.283333
13            Kerala                          7.290000
20         Rajasthan                         14.388333
25     Uttar Pradesh                          9.976667
5              Delhi                         16.145000
22        Tamil Nadu                          3.421667
24           Tripura                         30.613333
2              Bihar                         14.276667
18        Puducherry                          1.266667
8            Haryana                         24.165000
11          Jharkhand                         13.210000

    unemployment rate after lockdown  percentage change in unemployment
1                           6.563333                                5.56
21                          7.131429                                6.13
26                          7.370000                                6.37
16                          7.764444                                6.76
17                          8.566000                                7.57
7                           8.814000                                7.81
6                          10.301429                                9.30
27                         10.890000                                9.89
0                          11.126000                               10.13
15                         11.304000                               10.30
23                         12.291000                               11.29
14                         12.474000                               11.47
3                          12.656667                               11.66
4                          12.720000                               11.72
12                         13.793000                               12.79
19                         14.454000                               13.45
10                         14.562857                               13.56
9                          14.982000                               13.98
13                         15.404000                               14.40
20                         15.619000                               14.62
25                         16.478000                               15.48
5                          19.195000                               18.20
22                         19.475000                               18.48
24                         27.249000                               26.25
2                          27.459000                               26.46
18                         29.006250                               28.01
8                          30.887000                               29.89
11                         32.269000                               31.27
```

```
#percentage change in unemployment after lockdown

fig = px.bar(plot, x='Region',y='percentage change in unemployment',color='percentage change in unemployment',
            title='percentage change in Unemployment in each state after lockdown',template='ggplot2')
fig.show()
```


percentage change in Unemployment in each state after lockdown
percentage change in unemployment
0
5
10
15
20
25
30
Assam
Sikkim
Uttarakhand
Meghalaya
Odisha
Gujarat
Goa
West Bengal
Andhra Pradesh
Maharashtra
Telangana
Madhya Pradesh
Chandigarh
Chhattisgarh
Karnataka
Punjab
Jammu & Kashmir
Himachal Pradesh
Kerala
Rajasthan
Uttar Pradesh
Delhi
Tamil Nadu
Tripura
Bihar
Puducherry
Haryana
Jharkhand
Region
percentage change in unemployment
30
25
20
15
10