unemployment-data-analysis

September 23, 2023

1 Task 2 - Unemployment Data Analysis

[25]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import calendar
%matplotlib inline
```

[2]:
```
df=pd.read_csv("Unemployment_Rate_upto_11_2020.csv")
```

[4]:
```
df
```

[4]:
```
             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   
..              …           …        …                                …   
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]
```

[5]:
```
df.columns = ['States', 'Date', 'Frequency', 'Estimated_Unemployment_Rate',␣
 ↪'Estimated_Employed',
             'Estimated_Labour_Participation_Rate', 'Region', 'Longitude',␣
 ↪'Latitude']
df.head()     # Checking first five rows of the dataset
```

[5]:
```
           States        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

   Estimated_Employed  Estimated_Labour_Participation_Rate Region  Longitude  \
0            16635535                                41.02  South    15.9129
1            16545652                                40.90  South    15.9129
2            15881197                                39.18  South    15.9129
3            11336911                                33.10  South    15.9129
4            12988845                                36.46  South    15.9129

   Latitude
0     79.74
1     79.74
2     79.74
3     79.74
4     79.74
```

2 Understanding data and its analysis

[6]:
```
print(df.shape)
```

```
(267, 9)
```

[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   States                              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
```

[8]:
```
round(df.describe(),2)
```

[8]:
```
       Estimated_Unemployment_Rate  Estimated_Employed  \
count                       267.00              267.00
mean                         12.24         13962105.72
std                          10.80         13366318.36
min                           0.50           117542.00
25%                           4.84          2838930.50
50%                           9.65          9732417.00
75%                          16.76         21878686.00
max                          75.85         59433759.00

       Estimated_Labour_Participation_Rate  Longitude  Latitude
count                               267.00     267.00    267.00
mean                                 41.68      22.83     80.53
std                                   7.85       6.27      5.83
min                                  16.77      10.85     71.19
25%                                  37.26      18.11     76.09
50%                                  40.39      23.61     79.02
75%                                  44.06      27.28     85.28
max                                  69.69      33.78     92.94
```

```
[9]: df["Region"].value_counts()
```

```
[9]: North        79
     South        60
     West         50
     East         40
     Northeast    38
     Name: Region, dtype: int64
```

```
[11]: df['States'].unique()
```

```
[11]: 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)
```

```
[12]: df['States'].value_counts()
```

```
[12]: 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: States, dtype: int64
```

[13]:
```
# Checking the Frequency columns
df['Frequency'].value_counts()
```

[13]:
```
 M    267
Name: Frequency, dtype: int64
```

[14]:
```
df['Date'].value_counts()
```

[14]:
```
 31-03-2020    27
 31-05-2020    27
 30-06-2020    27
 31-07-2020    27
 31-08-2020    27
 30-09-2020    27
 31-10-2020    27
 31-01-2020    26
 29-02-2020    26
 30-04-2020    26
Name: Date, dtype: int64
```

3 Convert the ‘Date’ column to datetime format

[15]:
```
# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Create a new 'Month' column by extracting the month from the 'Date' column
df['Month'] = df['Date'].dt.month

# Display the updated DataFrame
print(df)
```

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

     Estimated_Employed  Estimated_Labour_Participation_Rate Region  \
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  Month
0      15.9129    79.740      1
1      15.9129    79.740      2
2      15.9129    79.740      3
3      15.9129    79.740      4
4      15.9129    79.740      5
..         …         …    …
262    22.9868    87.855      6
263    22.9868    87.855      7
264    22.9868    87.855      8
265    22.9868    87.855      9
266    22.9868    87.855     10

[267 rows x 10 columns]
```

[16]:

```python
# Extract the month and create a new 'Month' column
df['Month'] = df['Date'].dt.month.apply(lambda x: calendar.month_name[x])

# Print the updated Dataframe
df.head()
```

[16]:

```
           States       Date Frequency  Estimated_Unemployment_Rate  \
0  Andhra Pradesh 2020-01-31         M                         5.48
1  Andhra Pradesh 2020-02-29         M                         5.83
2  Andhra Pradesh 2020-03-31         M                         5.79
3  Andhra Pradesh 2020-04-30         M                        20.51
4  Andhra Pradesh 2020-05-31         M                        17.43

   Estimated_Employed  Estimated_Labour_Participation_Rate Region  Longitude  \
0            16635535                                41.02  South    15.9129
1            16545652                                40.90  South    15.9129
2            15881197                                39.18  South    15.9129
3            11336911                                33.10  South    15.9129
4            12988845                                36.46  South    15.9129
```

```
   Latitude     Month
0     79.74   January
1     79.74  February
2     79.74     March
3     79.74     April
4     79.74       May
```

[17]:

```
region_analysis = df.
 ↪groupby(['Region'])[['Estimated_Unemployment_Rate','Estimated_Employed',␣
 ↪'Estimated_Labour_Participation_Rate']].mean().reset_index()
region_analysis = round(region_analysis, 2)
region_analysis
```

[17]:

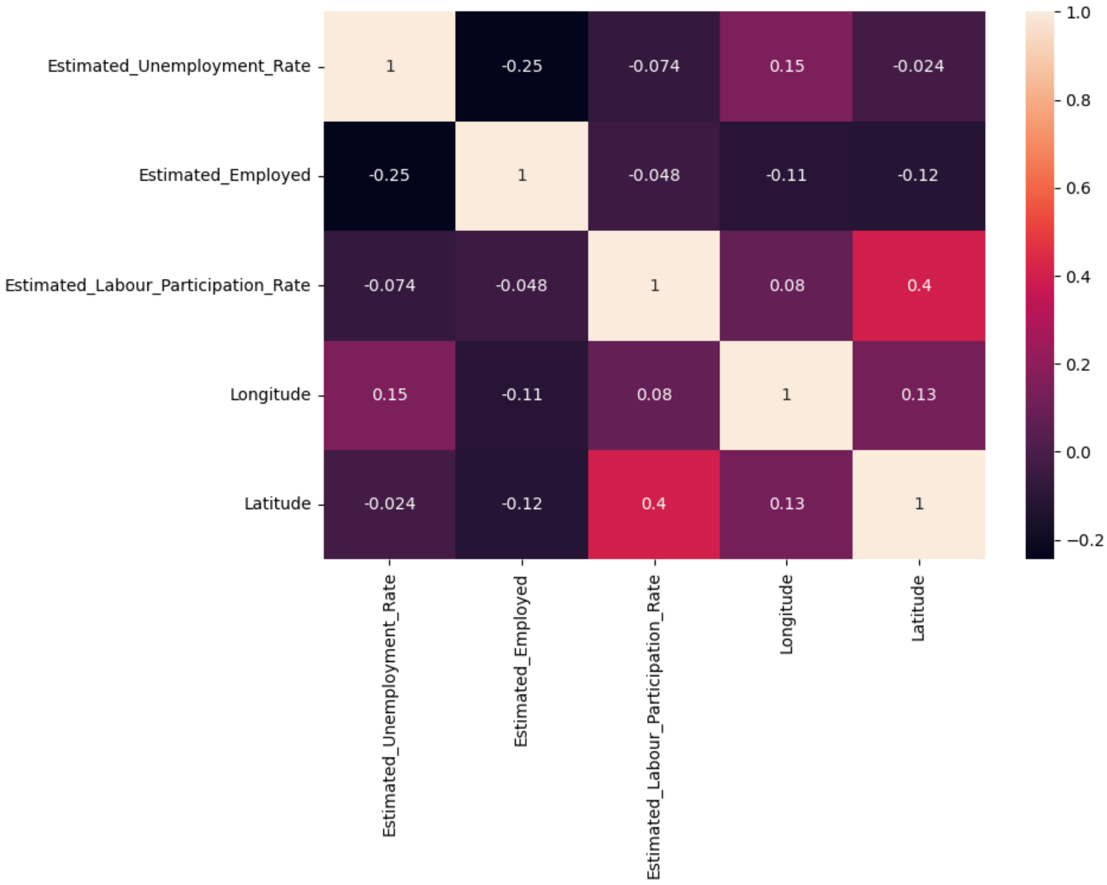
```
      Region  Estimated_Unemployment_Rate  Estimated_Employed  \
0       East                        13.92         19602366.90
1      North                        15.89         13072487.92
2  Northeast                        10.95          3617105.53
3      South                        10.45         14040589.33
4       West                         8.24         18623512.72

   Estimated_Labour_Participation_Rate
0                                40.11
1                                38.70
2                                52.06
3                                40.44
4                                41.26
```

[18]:

```
#Correlational Analysis
hm = df[['Estimated_Unemployment_Rate', 'Estimated_Employed',␣
 ↪'Estimated_Labour_Participation_Rate', 'Longitude','Latitude']].corr()
plt.figure(figsize= (9,6))

sns.heatmap(hm, annot =True)
```

[18]:

```
<Axes: >
```



4 Analysing Unemployment Rate by State And Region

[19]:

```
df.columns
```

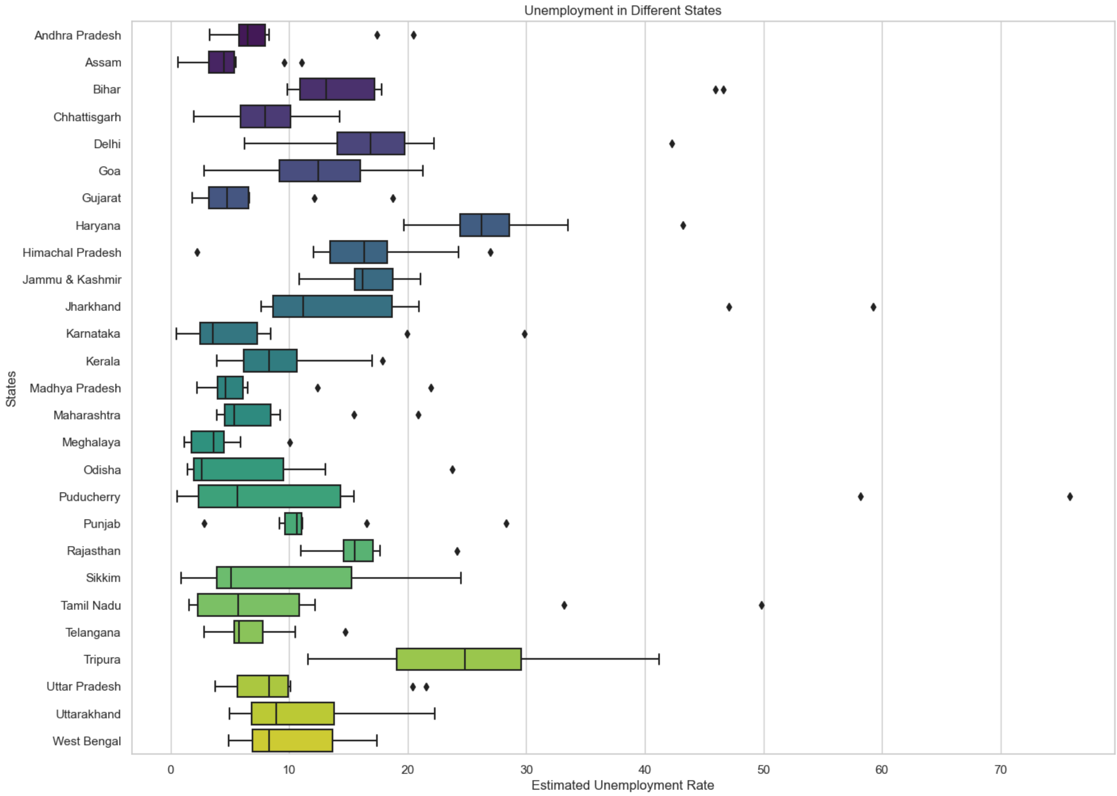
[19]:

```
Index(['States', 'Date', 'Frequency', 'Estimated_Unemployment_Rate',
       'Estimated_Employed', 'Estimated_Labour_Participation_Rate', 'Region',
       'Longitude', 'Latitude', 'Month'],
      dtype='object')
```

[62]:

```
#Create boxplot
sns.set(style="whitegrid")

# Transpose the DataFrame to have states on the horizontal axis
df_transposed = df.pivot(columns='States', values='Estimated_Unemployment_Rate')

# Create a box plot using Seaborn
plt.figure(figsize=(16, 12))  # Set the figure size
sns.boxplot(data=df_transposed, orient='h', palette='viridis')
```

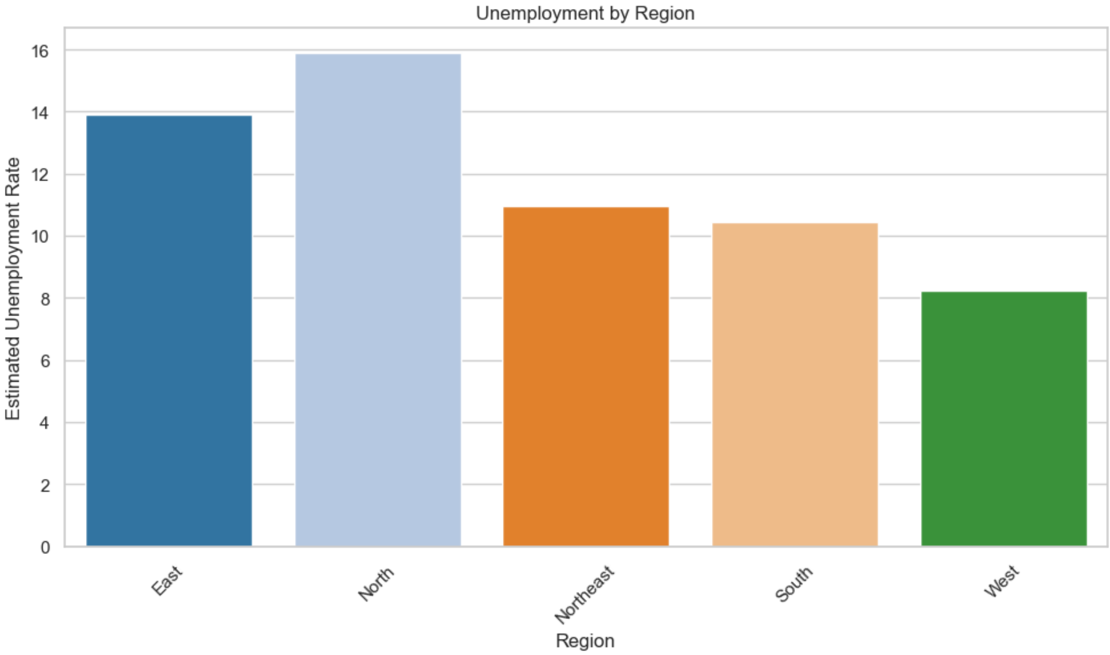
```
# Set the title and labels
plt.title('Unemployment in Different States')
plt.xlabel('Estimated Unemployment Rate')
plt.ylabel('States')

# Show the plot
plt.show()
```



[64]:

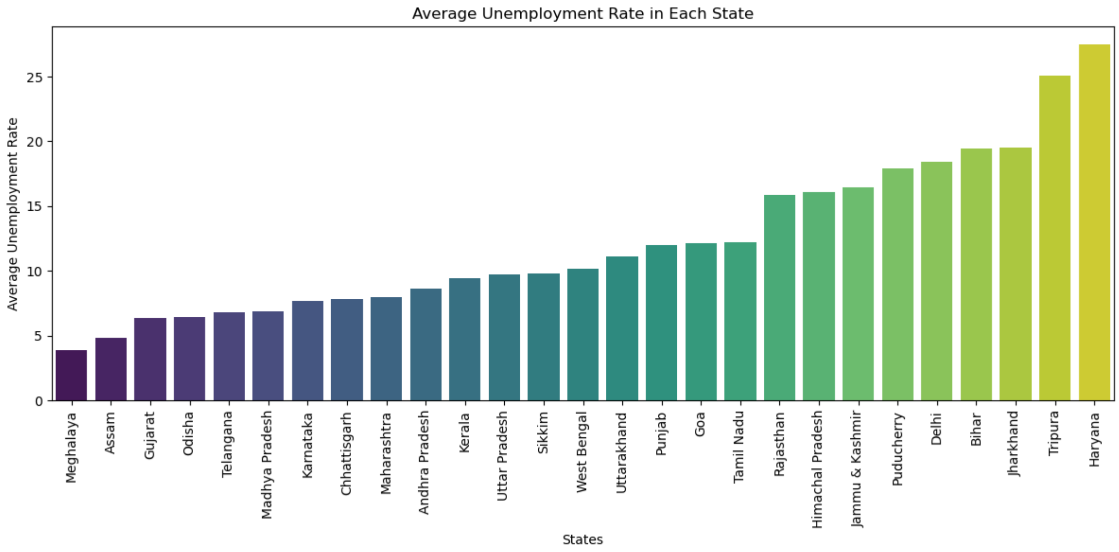
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'region_analysis' contains your dataset

# Create a colormap with a unique color for each region
colors = sns.color_palette('tab20', n_colors=len(region_analysis['Region']))

# Create a figure and axis using Seaborn
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Region', y='Estimated_Unemployment_Rate',␣
 ↪data=region_analysis, palette=colors)
```

```python
# Set the title and labels
plt.title('Unemployment by Region')
plt.xlabel('Region')
plt.ylabel('Estimated Unemployment Rate')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



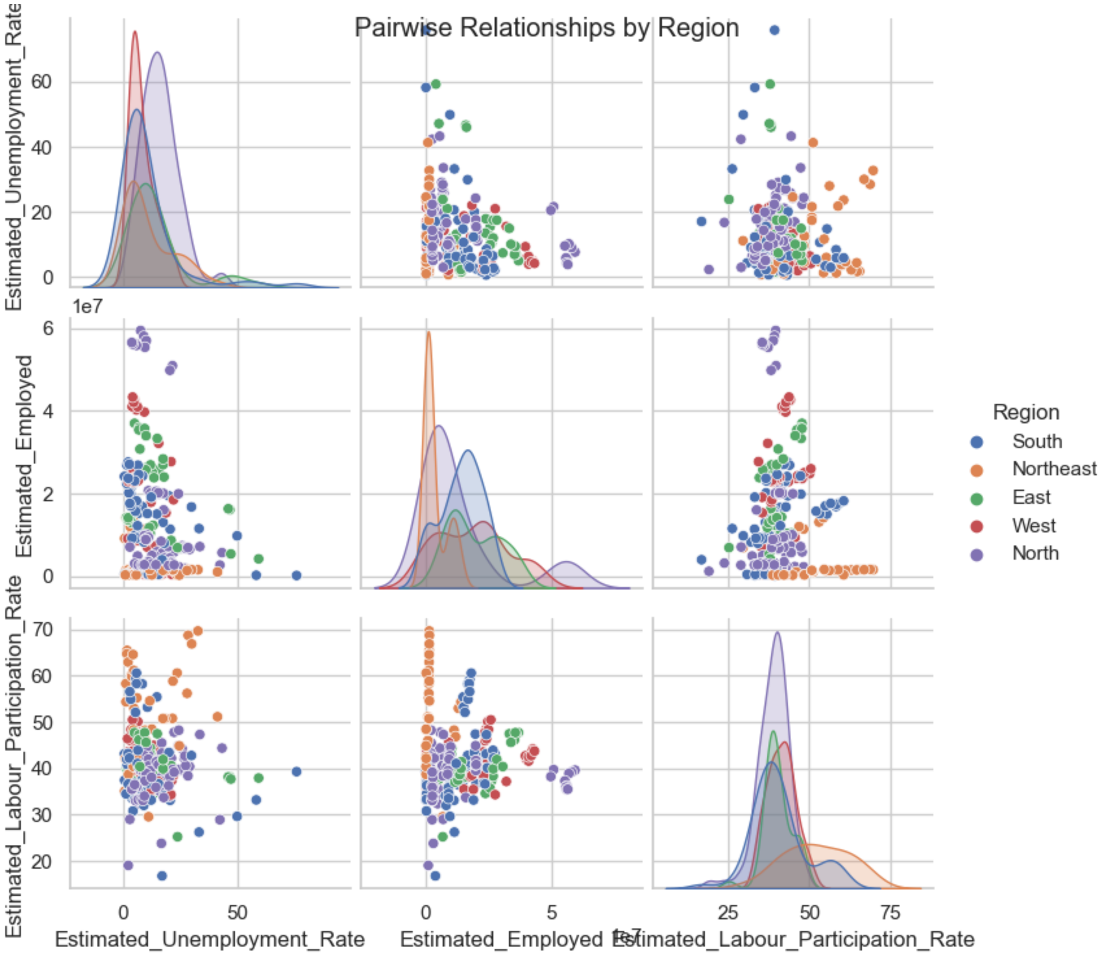
[54]:

```python
# Group by 'States' and calculate the mean of 'Estimated_Unemployment_Rate'
avg_unemp = df.groupby('States')['Estimated_Unemployment_Rate'].mean().
 ↪reset_index()

# Sort the DataFrame by 'Estimated_Unemployment_Rate'
avg_unemp = avg_unemp.sort_values('Estimated_Unemployment_Rate')

# Create a bar plot using Seaborn
plt.figure(figsize=(12, 6))
sns.barplot(data=avg_unemp, x='States', y='Estimated_Unemployment_Rate',␣
 ↪palette='viridis')
plt.title('Average Unemployment Rate in Each State')
plt.xlabel('States')
plt.ylabel('Average Unemployment Rate')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.tight_layout()

# Show the plot
plt.show()
```



[65]:
```
# Set the figure size
plt.figure(figsize=(12, 8))

# Select the columns of interest
pp = df[['Estimated_Unemployment_Rate', 'Estimated_Employed',␣
 ↪'Estimated_Labour_Participation_Rate', 'Region']]

# Create the pairplot with different colors for each region
sns.pairplot(pp, hue='Region')

# Set a title (optional)
plt.suptitle('Pairwise Relationships by Region')

# Show the plot
plt.show()
```

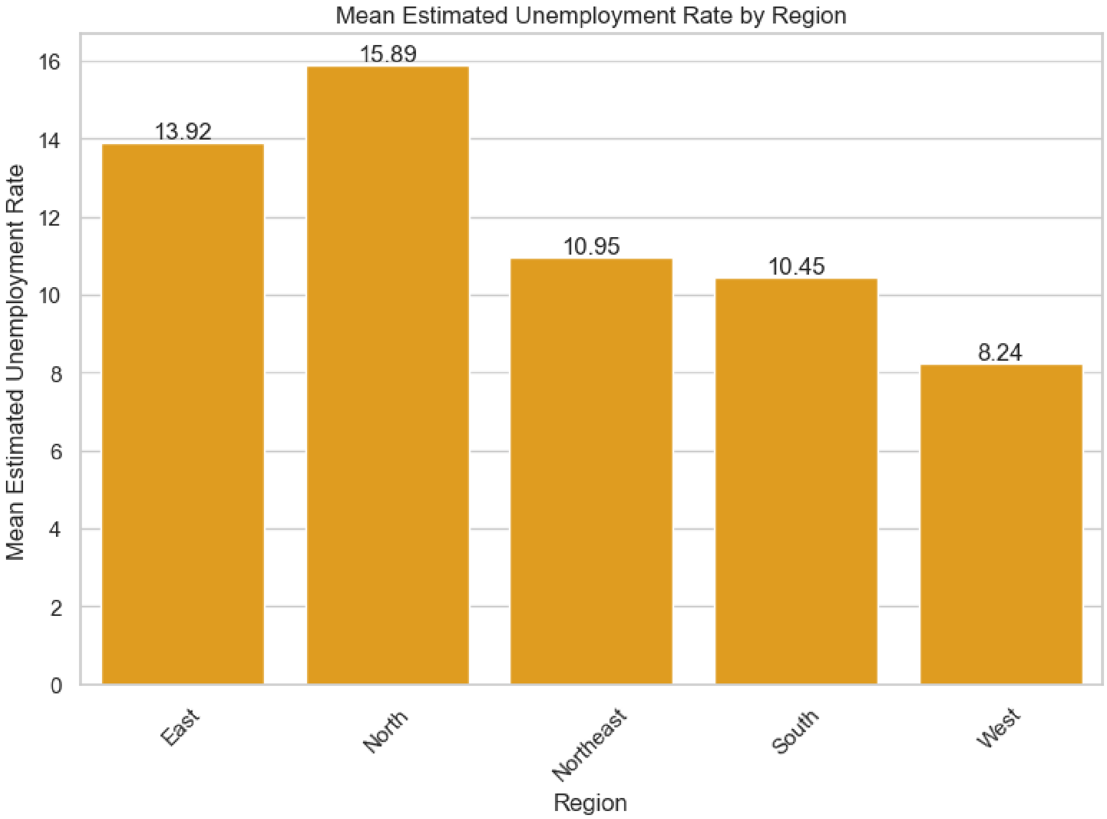
```
<Figure size 1200x800 with 0 Axes>
```



[83]:
```
#sunburst chart

unempl = df[['States','Region','Estimated_Unemployment_Rate',␣
 ↪'Estimated_Employed', 'Estimated_Labour_Participation_Rate']]
unempo = unempl.groupby(['Region','States'])['Estimated_Unemployment_Rate'].
 ↪mean().reset_index()
fig = px.sunburst(unempo, path = ['Region','States'], values =␣
 ↪'Estimated_Unemployment_Rate',title = 'Unemployment Rate to State and␣
 ↪Region', height = 600)
fig.show()
```

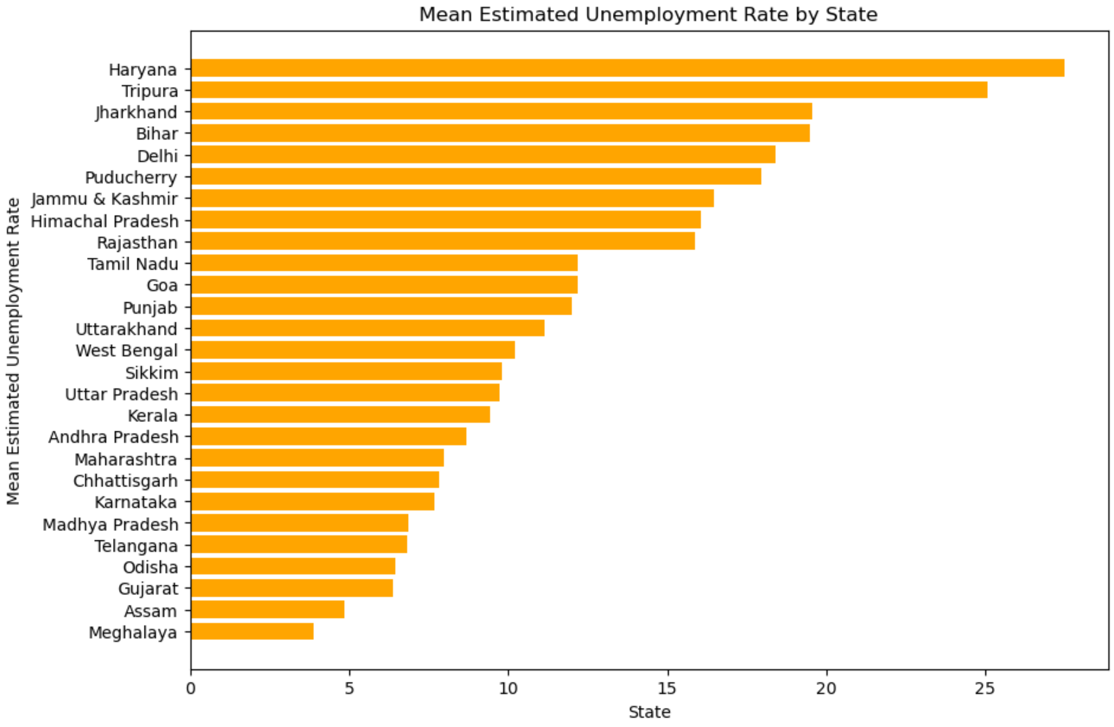
[66]:
```
# Group data by 'Region' and calculate the mean of 'Estimated_Unemployment_Rate'
region_means = df.groupby('Region')['Estimated_Unemployment_Rate'].mean().
 ↪reset_index()

# Set the style of Seaborn plots (optional)
sns.set(style="whitegrid")
```

```
# Create a bar chart with Seaborn
plt.figure(figsize=(8, 6))
sns.barplot(x='Region', y='Estimated_Unemployment_Rate', data=region_means,␣
 ↪color='orange')

# Set the title and labels
plt.title('Mean Estimated Unemployment Rate by Region')
plt.xlabel('Region')
plt.ylabel('Mean Estimated Unemployment Rate')
plt.xticks(rotation=45)  # Rotate x-axis labels for better visibility

# Annotate the values on top of each bar
for index, row in region_means.iterrows():
    plt.text(index, row['Estimated_Unemployment_Rate'],␣
 ↪f'{row["Estimated_Unemployment_Rate"]:.2f}', ha='center', va='bottom')

plt.tight_layout()  # Adjust spacing
plt.show()
```



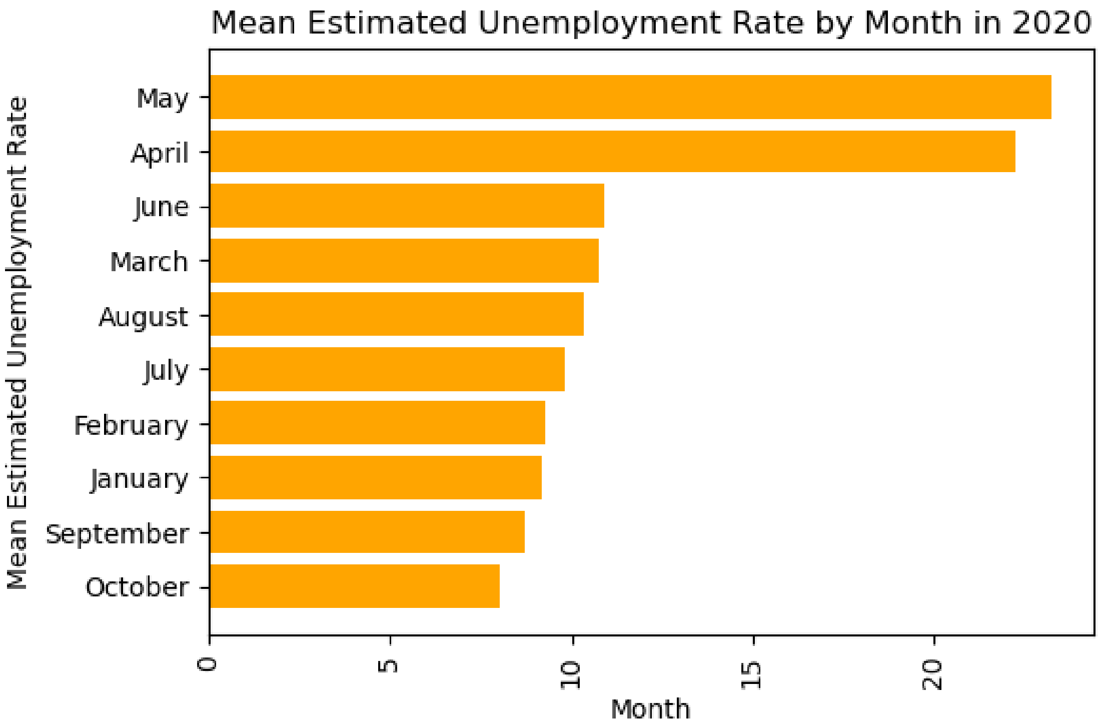
[43]:

```
# Group data by 'State' and calculate the mean of 'Estimated_Unemployment_Rate'
state_means = df.groupby('State')['EUR'].mean().sort_values(ascending=True)

# Create a bar chart with State names on the x-axis and mean Estimated␣
 ↪Unemployment Rate on the y-axis
plt.figure(figsize=(10, 7))
plt.barh(state_means.index, state_means.values, color='orange')
plt.title('Mean Estimated Unemployment Rate by State')
plt.xlabel('State')
plt.ylabel('Mean Estimated Unemployment Rate')

plt.show()
```



[80]:

```
# Group data by Month and calculate the mean of 'Estimated_Unemployment_Rate'
Date_means = df.groupby("Month")['Estimated_Unemployment_Rate'].mean().
 ↪sort_values(ascending=True)

plt.figure(figsize=(6,4))
plt.barh(Date_means.index, Date_means.values, color='orange')
plt.title("Mean Estimated Unemployment Rate by Month in 2020")
plt.xlabel("Month")
plt.ylabel("Mean Estimated Unemployment Rate")
plt.xticks(rotation=90)
plt.show()
```



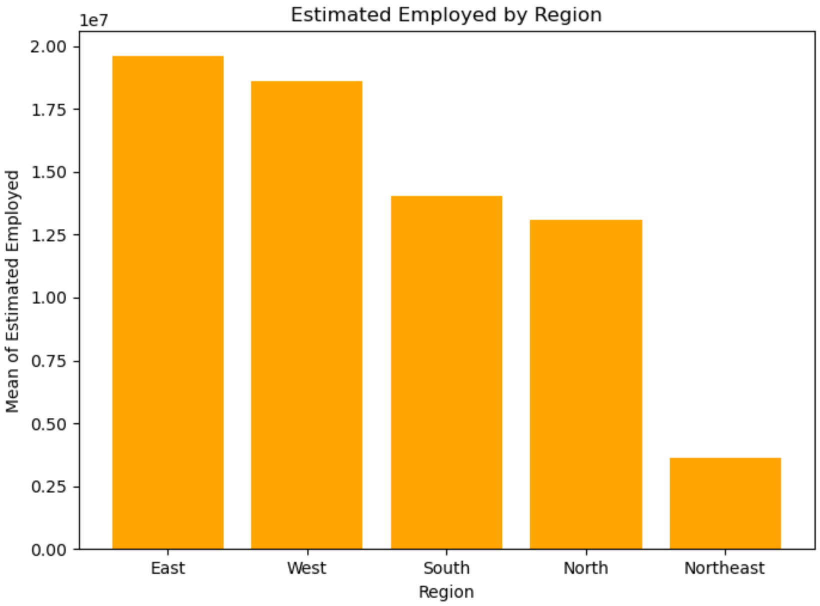
[81]:

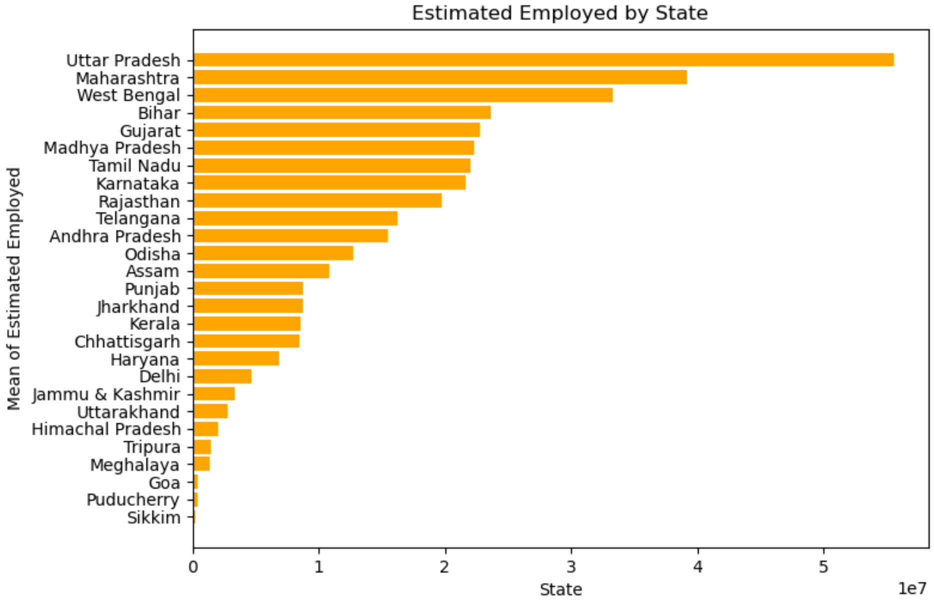
```
# Create a hierarchical DataFrame for the treemap
hierarchical_df = df.groupby(['Region',␣
 ↪'States'])['Estimated_Unemployment_Rate'].mean().reset_index()

# Create an interactive treemap using Plotly Express
fig = px.treemap(hierarchical_df, path=['Region', 'States'],␣
 ↪values='Estimated_Unemployment_Rate',
                 color='Estimated_Unemployment_Rate',␣
 ↪hover_data=['Estimated_Unemployment_Rate'],
                 color_continuous_scale='Viridis')

# Add labels for Estimated_Unemployment_Rate values
fig.update_traces(textinfo="label+value")

# Customize the treemap layout
fig.update_layout(title='Interactive Treemap of Estimated Unemployment Rate by␣
 ↪Region and State',
                  coloraxis_showscale=True)

# Show the interactive treemap
fig.show()
```

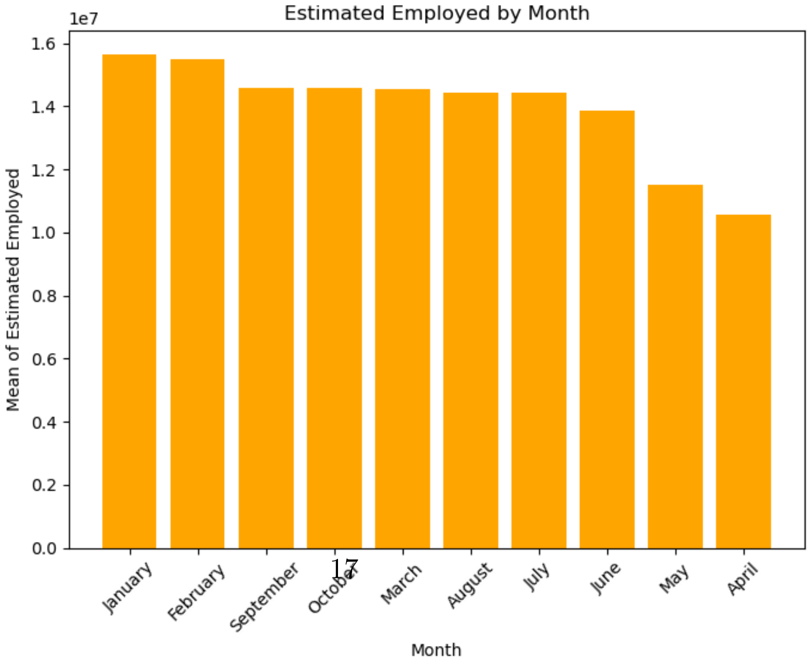
5 Analysing Estimated Employed¶

[82]:

```
# Group data by Region and calculate the mean of 'Estimated_Employed'
EE_region = df.groupby('Region')['Estimated_Employed'].mean().
  ↪sort_values(ascending=False)

# Group data by State and calculate the mean of 'Estimated_Employed'
EE_state = df.groupby('States')['Estimated_Employed'].mean().
  ↪sort_values(ascending=True)

# Group data by Month and calculate the mean of 'Estimated_Employed'
EE_month = df.groupby('Month')['Estimated_Employed'].mean().
  ↪sort_values(ascending=False)

# Create separate subplots for 'Region', 'States', and 'Month'
fig, axes = plt.subplots(3, 1, figsize=(8, 16))

# Creating bar chart for Region
axes[0].bar(EE_region.index, EE_region.values, color='orange')
axes[0].set_title("Estimated Employed by Region")
axes[0].set_xlabel("Region")
axes[0].set_ylabel("Mean of Estimated Employed")

# Creating bar chart for State
axes[1].barh(EE_state.index, EE_state.values, color='orange')
axes[1].set_title("Estimated Employed by State")
axes[1].set_xlabel("State")
axes[1].set_ylabel("Mean of Estimated Employed")


# Creating bar chart for Month
axes[2].bar(EE_month.index, EE_month.values, color='orange')
axes[2].set_title("Estimated Employed by Month")
axes[2].set_xlabel("Month")
axes[2].set_ylabel("Mean of Estimated Employed")
axes[2].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better␣
  ↪readability

# Adjust layout and display the subplots
plt.tight_layout()
plt.show()
```


Estimated Employed by Region
1e7
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Mean of Estimated Employed
East
West
South
North
Northeast
Region


Estimated Employed by State
Uttar Pradesh
Maharashtra
West Bengal
Bihar
Gujarat
Madhya Pradesh
Tamil Nadu
Karnataka
Rajasthan
Telangana
Andhra Pradesh
Odisha
Assam
Punjab
Jharkhand
Kerala
Chhattisgarh
Haryana
Delhi
Jammu & Kashmir
Uttarakhand
Himachal Pradesh
Tripura
Meghalaya
Goa
Puducherry
Sikkim
Mean of Estimated Employed
0
1
2
3
4
5
1e7
State


Estimated Employed by Month
1e7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Mean of Estimated Employed
January
February
September
October
March
August
July
June
May
April
Month

6 Insights for Estimated Unemployment Rate

1. May and April Months has the highest estimated unemployment rate.
2. North Region has the highest estimted unemployment rate. East Region is at the second place.¶
3. In North Region Haryana has the highest estimated unemployment rate of 27.447.¶
4. In Northeast Region Tripura has the highest estimated unemployment rate of 25.055.¶
5. In South Region Puducherry has the highest estimated unemployment rate of 19.942.¶
6. In East Region Jharkhand has the highest estimated unemployment rate of 19,539.¶
7. In West Region Goa has the highest estimated unemployment rate of 12.167.¶

[ ]: