UNEMPLOYMENT IN INDIA

Introduction

Unemployment is a critical economic indicator that reflects the health of a country's labor market. In this analysis, we examine unemployment trends across various regions in India, exploring the distribution, variability, and trends in unemployment rates. Additionally, we investigate labor participation rates and employment figures to provide a comprehensive overview of the labor market.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('Unemployment in India.csv')
df1=pd.read_csv('Unemployment_Rate_upto_11_2020.csv')
```

Data Overview:

We utilized two datasets focusing on unemployment in India, covering multiple regions and spanning various time periods. The data was pre-processed to handle missing values, remove inconsistencies, and standardize column names. This preparation ensured the accuracy and reliability of our analysis.

In [3]:
```
df.head()
```

Out[3]:

	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
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707.0	43.50	Rural
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693.0	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762.0	44.68	Rural

In [4]:
```
df.tail()
```

Out[4]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
763	NaN	NaN	NaN	NaN	NaN	NaN	NaN
764	NaN	NaN	NaN	NaN	NaN	NaN	NaN
765	NaN	NaN	NaN	NaN	NaN	NaN	NaN
766	NaN	NaN	NaN	NaN	NaN	NaN	NaN
767	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Area'],
      dtype='object')
```

In [6]:
```
df.shape
```

Out[6]:
```
(768, 7)
```

In [7]:
```
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
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Region                                      28
 Date                                       28
 Frequency                                  28
 Estimated Unemployment Rate (%)            28
 Estimated Employed                         28
 Estimated Labour Participation Rate (%)    28
Area                                        28
dtype: int64
```

In [9]:
```
df.dropna(inplace=True)
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Region                                      0
 Date                                       0
 Frequency                                  0
 Estimated Unemployment Rate (%)            0
 Estimated Employed                         0
 Estimated Labour Participation Rate (%)    0
Area                                        0
dtype: int64
```

In [11]:
```
df.duplicated().any()
```

Out[11]:
```
False
```

In [65]:
```
df.to_csv(r'C:\Users\Sanjay\OneDrive\Desktop\Amith\Unemployee_data.csv', index
=True) #Export data to powerbi
```

In [13]:
```
df.Region.value_counts()
```

Out[13]:
```
Delhi               28
Chhattisgarh        28
West Bengal         28
Himachal Pradesh    28
Rajasthan           28
Gujarat             28
Punjab              28
Uttar Pradesh       28
Haryana             28
Jharkhand           28
Tamil Nadu          28
Andhra Pradesh      28
Karnataka           28
Maharashtra         28
Bihar               28
Odisha              28
Kerala              28
Madhya Pradesh      28
Tripura             28
Telangana           28
Uttarakhand         27
Meghalaya           27
Puducherry          26
Assam               26
Goa                 24
Jammu & Kashmir     21
Sikkim              17
Chandigarh          12
Name: Region, dtype: int64
```

In [14]:
```
#remove leading and trailing spaces from the column names
df.columns=df.columns.str.strip()
```

In [15]:
```
df.head()
```

Out[15]:

	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
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707.0	43.50	Rural
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693.0	43.97	Rural
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762.0	44.68	Rural

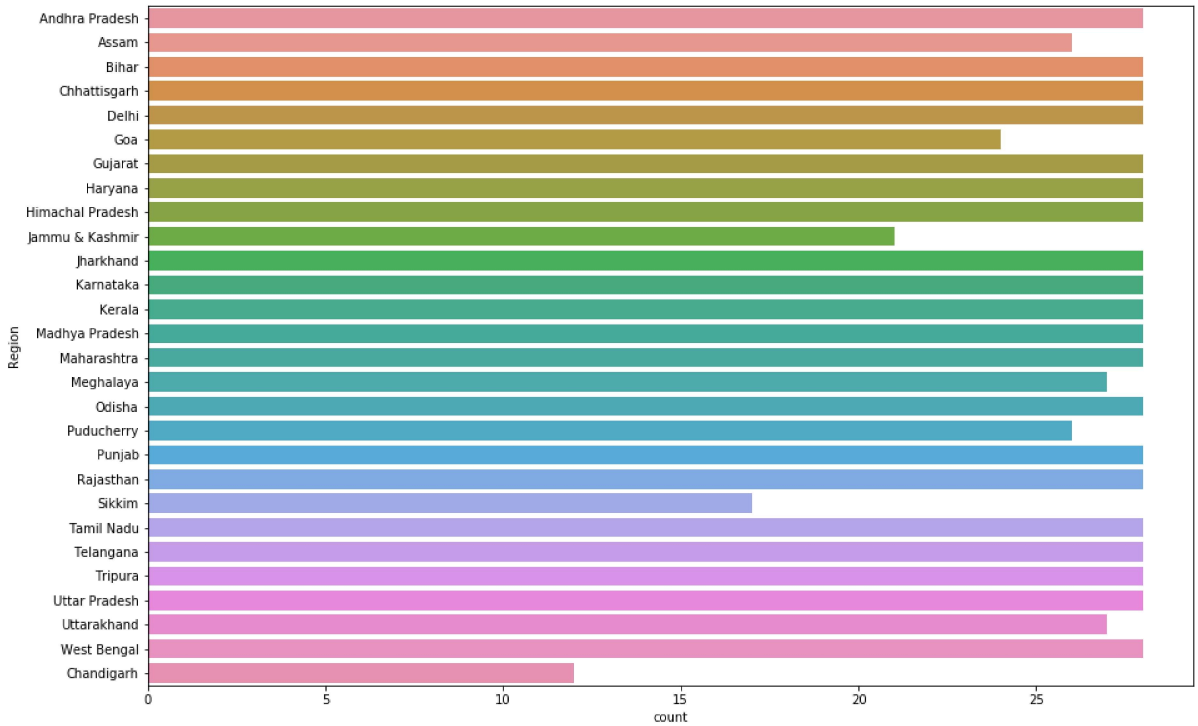
In [16]:
```
df['Estimated Unemployment Rate (%)']
```

Out[16]:
```
0       3.65
1       3.05
2       3.75
3       3.32
4       5.17
        ...
749     7.55
750     6.67
751    15.63
752    15.22
753     9.86
Name: Estimated Unemployment Rate (%), Length: 740, dtype: float64
```

Data Visualization

In [17]:
```
plt.figure(figsize=(15,10))
sns.countplot(y="Region",data=df)
plt.show
```

Out[17]: <function matplotlib.pyplot.show(*args, **kw)>



Analysis and Conclusion:

The count plot shows that some regions have more data than others. This means we might not have a clear picture of unemployment in regions with less data.

In [18]:
```
df.describe()
```

Out[18]:

	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

Highest and Lowest Average unemployment rates

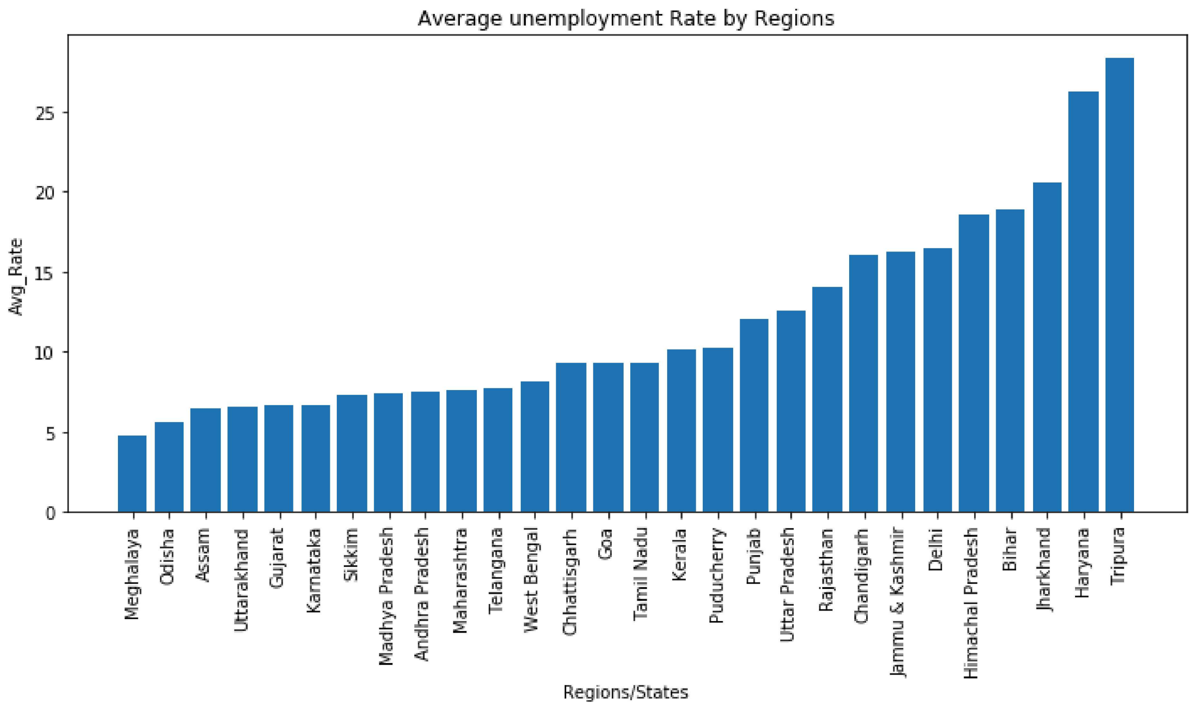
In [19]:
```
avg_unemployment_rate=df.groupby('Region')['Estimated Unemployment Rate (%)'].
mean()
state_with_highest_unemployment=avg_unemployment_rate.idxmax()
high_unemployment_rate=avg_unemployment_rate.max()
state_with_lowest_unemployment=avg_unemployment_rate.idxmin()
low_unemploy_rate=avg_unemployment_rate.min()

print(f"state  with Avg high unemployment : {state_with_highest_unemploymen
t}")
print(f"Avg high unemployment rate : {high_unemployment_rate}")

print(f"state  with Avg low unemployment : {state_with_lowest_unemployment}")
print(f"Avg low unemployment rate : {low_unemploy_rate}")
```

```
state  with Avg high unemployment : Tripura
Avg high unemployment rate : 28.350357142857142
state  with Avg low unemployment : Meghalaya
Avg low unemployment rate : 4.7988888888888885
```

In [20]:
```
avg_unemployment_rate.sort_values(ascending=True,inplace=True)
avg_unemployment_rate
```

Out[20]:
```
Region
Meghalaya            4.798889
Odisha               5.657857
Assam                6.428077
Uttarakhand          6.582963
Gujarat              6.663929
Karnataka            6.676071
Sikkim               7.249412
Madhya Pradesh       7.406429
Andhra Pradesh       7.477143
Maharashtra          7.557500
Telangana            7.737857
West Bengal          8.124643
Chhattisgarh         9.240357
Goa                  9.274167
Tamil Nadu           9.284286
Kerala              10.123929
Puducherry          10.215000
Punjab              12.031071
Uttar Pradesh       12.551429
Rajasthan           14.058214
Chandigarh          15.991667
Jammu & Kashmir     16.188571
Delhi               16.495357
Himachal Pradesh    18.540357
Bihar               18.918214
Jharkhand           20.585000
Haryana             26.283214
Tripura             28.350357
Name: Estimated Unemployment Rate (%), dtype: float64
```

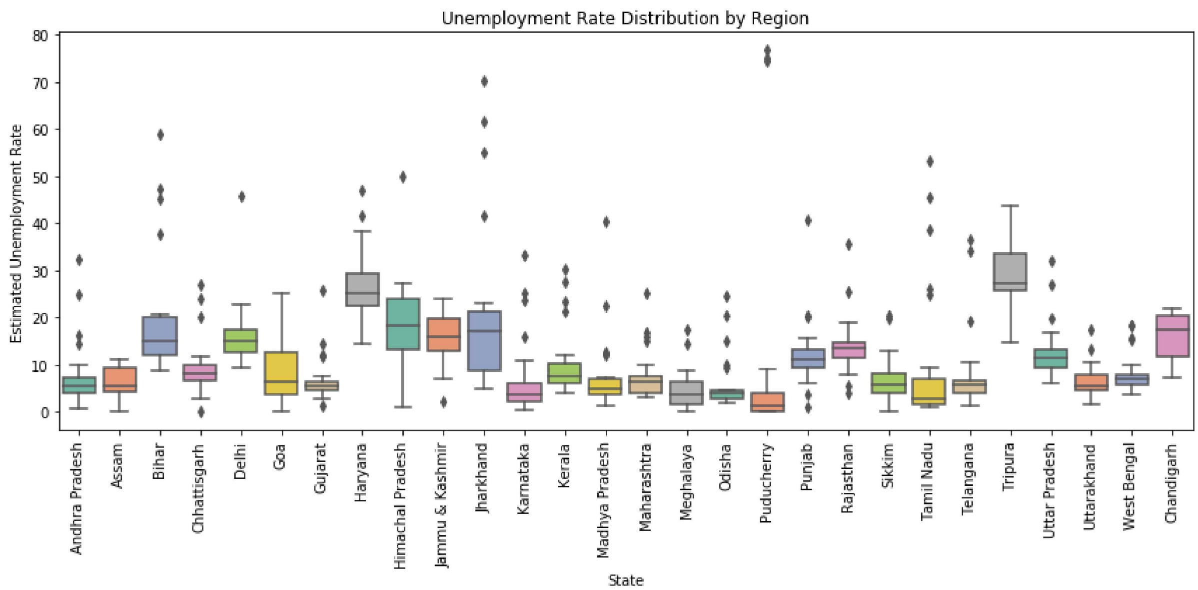
In [21]:
```
Regions=avg_unemployment_rate.index
Avg_Rate=avg_unemployment_rate.values

plt.figure(figsize=(10,6))
plt.bar(Regions,Avg_Rate)
plt.title('Average unemployment Rate by Regions')
plt.xlabel('Regions/States')
plt.ylabel('Avg_Rate')
plt.xticks(rotation=90)
plt.tight_layout()
```



Analysis and Conclusion:

The bar chart shows that some areas have higher unemployment than others. Tripura and Haryana have the highest unemployment, while Meghalaya and Odisha have the lowest. This means some areas have better job opportunities than others.

In [22]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Region', y='Estimated Unemployment Rate (%)', data=df,palette
="Set2")
plt.title("Unemployment Rate Distribution by Region")
plt.xlabel("State")
plt.ylabel("Estimated Unemployment Rate")
plt.xticks(rotation=90)
plt.tight_layout()
```

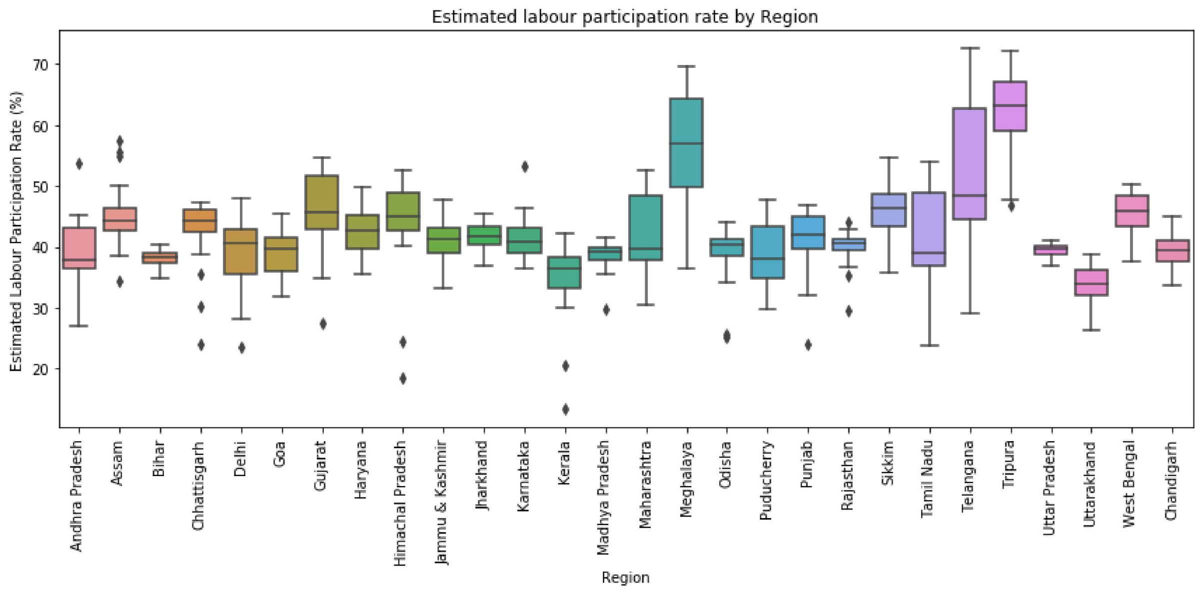


Analysis and Conclusion:
The box plot shows how unemployment rates vary between regions:

=> Regions with Wide Ranges: Areas like "Haryana," "Himachal Pradesh," and "Tripura" have large variations in unemployment rates and many outliers. This means these regions experience bigger swings in unemployment, which might indicate economic instability.

=> Regions with Narrow Ranges: Places like "Tamil Nadu" and "Uttarakhand" have more consistent unemployment rates and fewer outliers. This suggests these regions have more stable and predictable economic conditions.

In [23]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x='Region',data=df,y='Estimated Labour Participation Rate (%)')
plt.title('Estimated labour participation rate by Region')
plt.xticks(rotation=90)
plt.tight_layout()
```

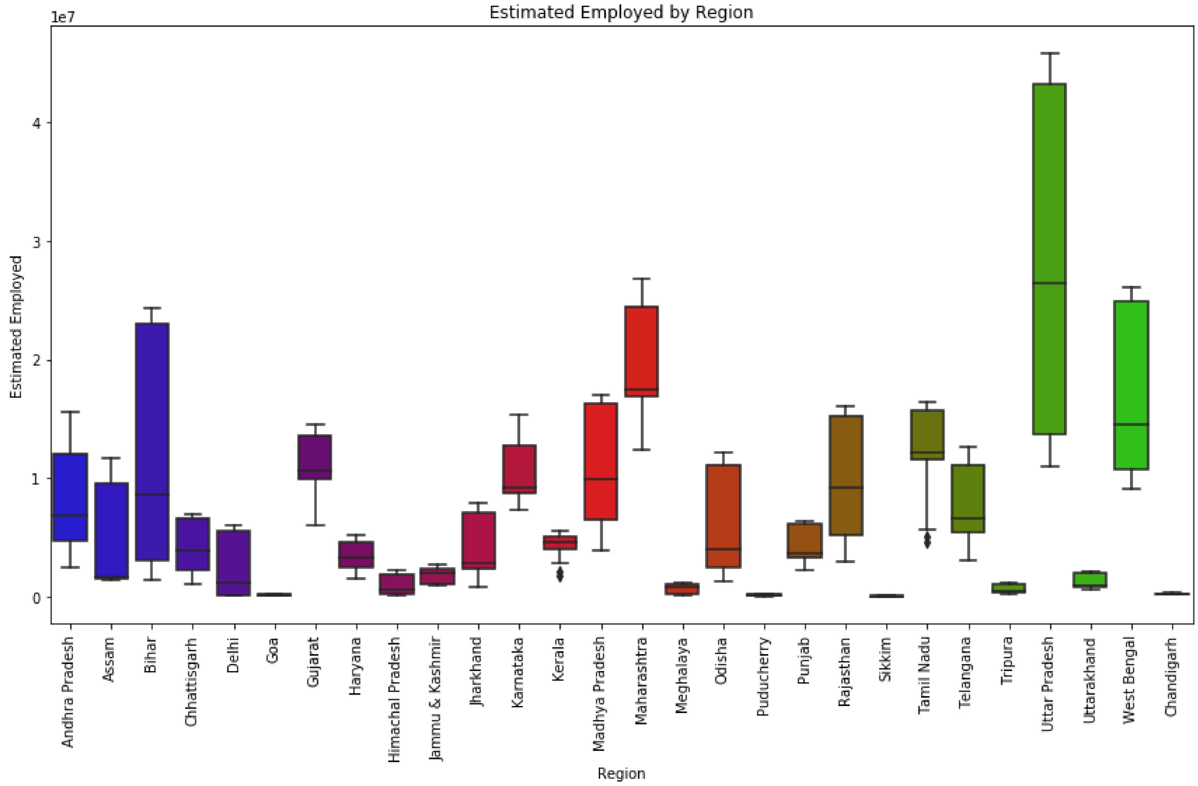


Analysis and Conclusion: The box plot shows how labor participation rates differ across regions in India:

Regions with Big Fluctuations: Areas like "Meghalaya," "Tamil Nadu," and "Telangana" have large variations in labor participation rates. This could mean these regions are facing economic challenges or changes in policies that affect employment.

Regions with Stable Rates: Places like "Punjab" and "Haryana" have more consistent labor participation rates. This suggests their labor markets are more stable and the economic environment is likely more resilient.

In [24]:
```
plt.figure(figsize=(12,8))
sns.boxplot(x='Region',y='Estimated Employed',data=df,palette='brg')
plt.title('Estimated Employed by Region')
plt.xticks(rotation=90)
plt.tight_layout()
```



Analysis and Conclusion:
The box plot shows how estimated employment varies across different regions in India:

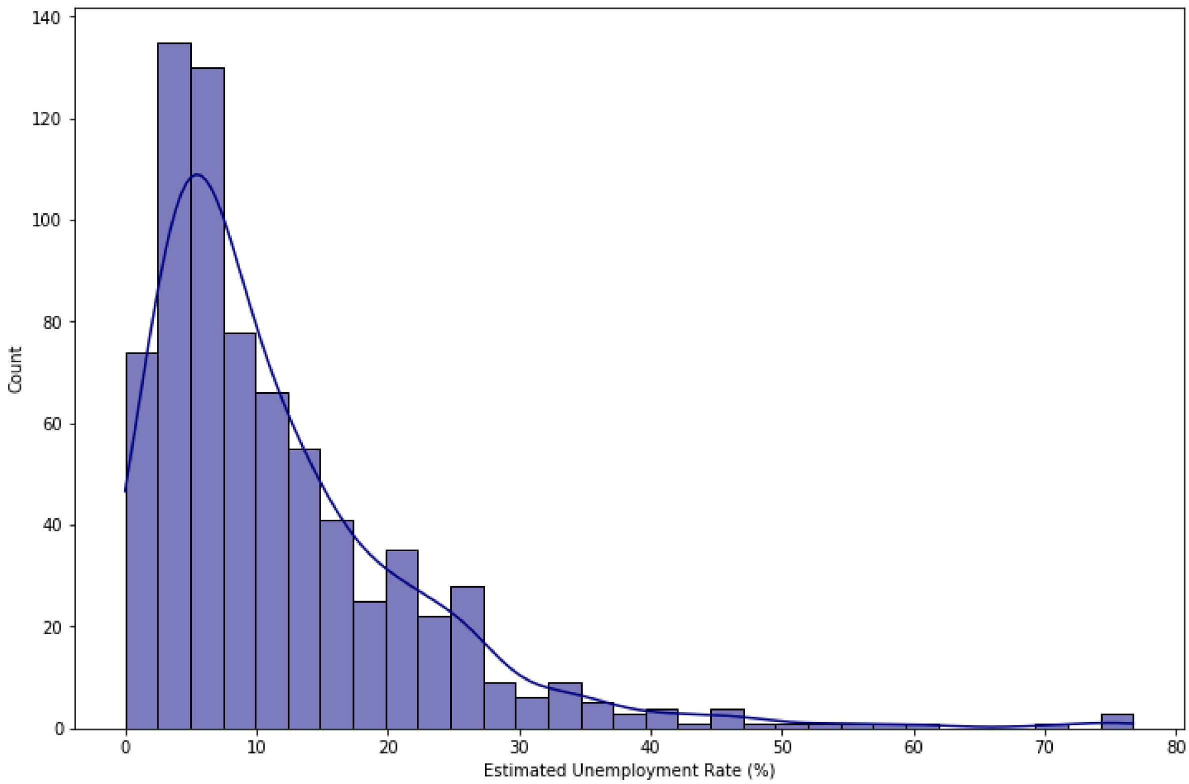
Regions with High Variability: Areas like "Uttar Pradesh," "Bihar," and "West Bengal" have a lot of variation in employment, suggesting they have large and diverse labor markets.

Regions with Low Variability: Places like "Goa" and "Puducherry" have less variation, indicating their labor markets are smaller and more stable.

This difference in variability likely comes from differences in population size, economic activities, and industries in each region.

In [25]:
```
plt.figure(figsize=(12,8))
sns.histplot(data=df, x="Estimated Unemployment Rate (%)", kde=True,color="navy")
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x241805cab88>

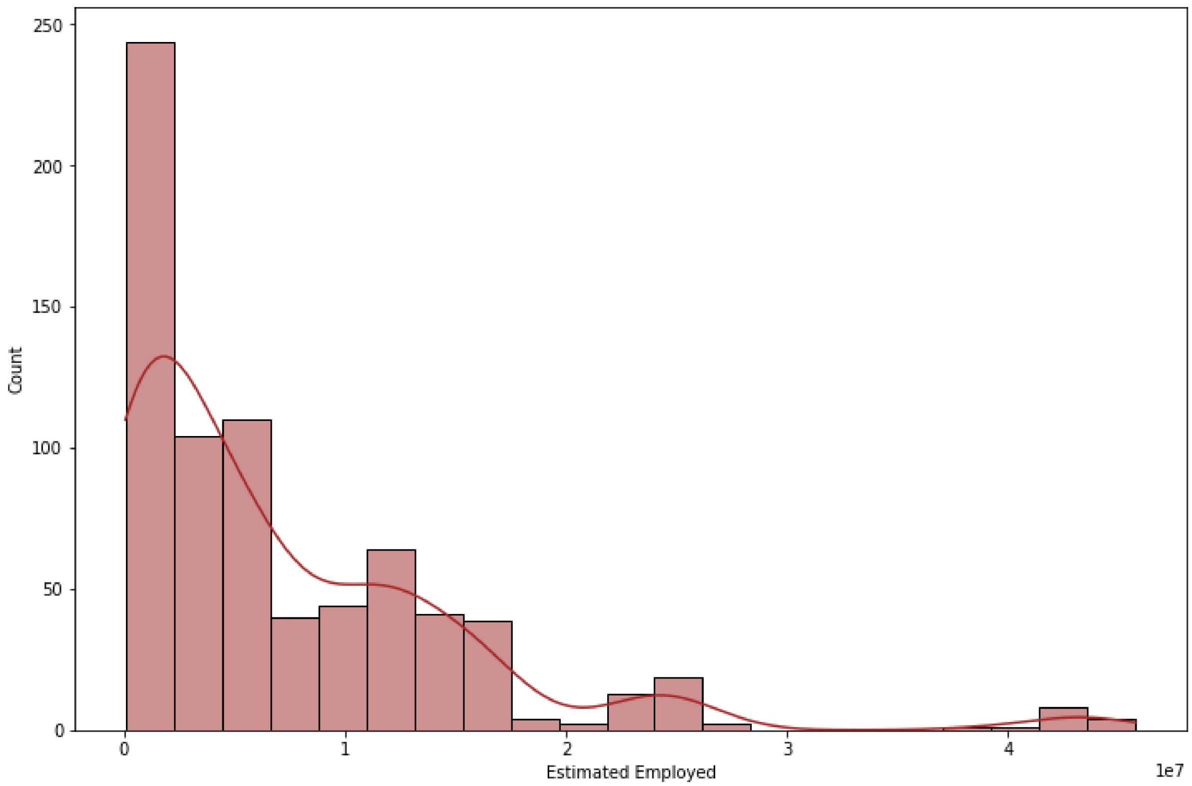


Analysis and Conclusion:

The histogram shows that most regions in India have low unemployment rates, with many areas grouped at the lower end. The KDE curve confirms this, showing a left-skewed distribution, meaning low unemployment rates are common. However, a few regions have much higher unemployment rates, which could indicate economic problems or challenges in those areas. Overall, unemployment is low in most places, but some regions may need extra help to improve their situation.

In [26]:
```
plt.figure(figsize=(12,8))
sns.histplot(data=df, x="Estimated Employed", kde=True,color="brown")
```

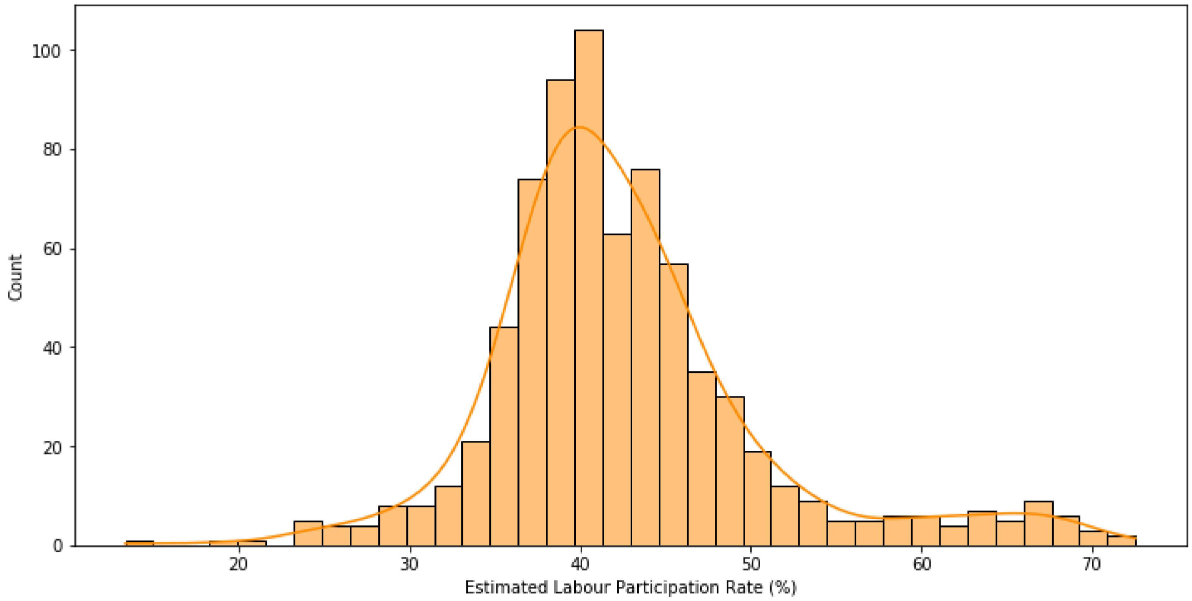
Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x241802eec48>
```



Analysis and Conclusion:
The histogram shows that most regions have lower employment numbers, but a few regions stand out with much higher employment. The right-skewed KDE curve confirms this, indicating that while low employment is typical, some regions have significantly higher figures, likely due to differences in economic conditions or population.

In [27]:
```
plt.figure(figsize=(12,6))
sns.histplot(data=df, x="Estimated Labour Participation Rate (%)", kde=True,co
lor="darkorange")
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x2418005f648>

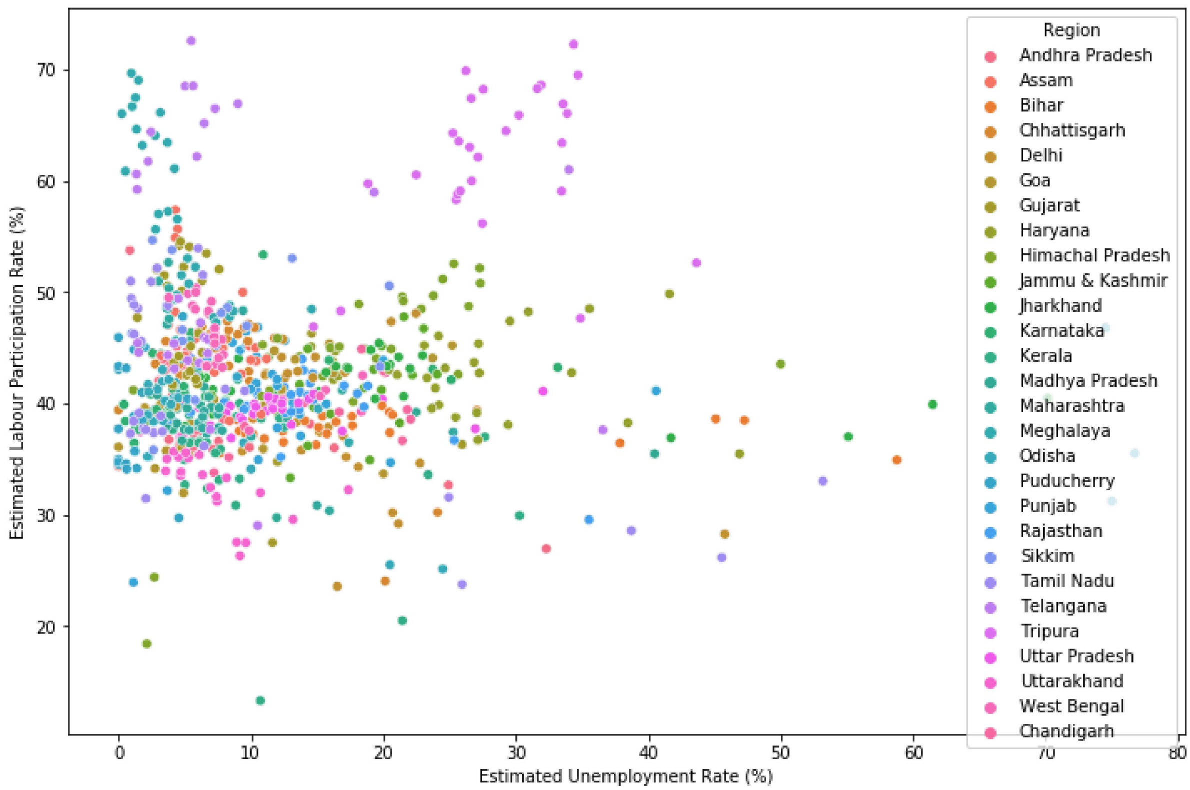


Analysis and Conclusion:
The histogram shows that most regions have labor participation rates around 40%, with the KDE curve confirming this near-normal distribution. This suggests that labor market engagement is fairly consistent across regions, with few extreme variations.

Visualisation of Unemployment Rate and Labor Participation Rate

In [28]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x='Estimated Unemployment Rate (%)',y='Estimated Labour Participation Rate (%)',hue='Region')
```

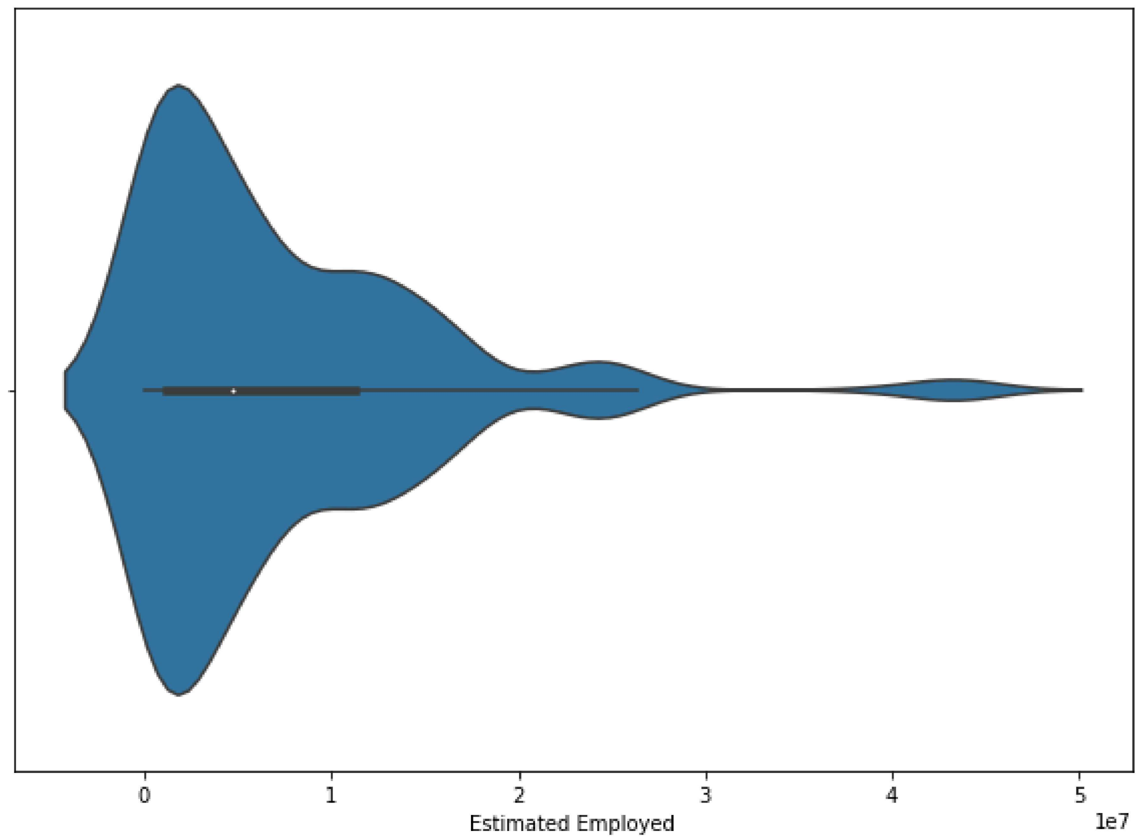
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x24180110808>



Analysis and Conclusion:
The scatter plot shows a negative relationship between unemployment and labor participation rates in Indian states, meaning higher unemployment is often associated with lower labor force participation. Addressing unemployment could help boost labor force engagement, and targeted policies might improve both participation and employment in different regions.

In [29]:
```
plt.figure(figsize=(10,7))
sns.violinplot(x=df["Estimated Employed"])
```

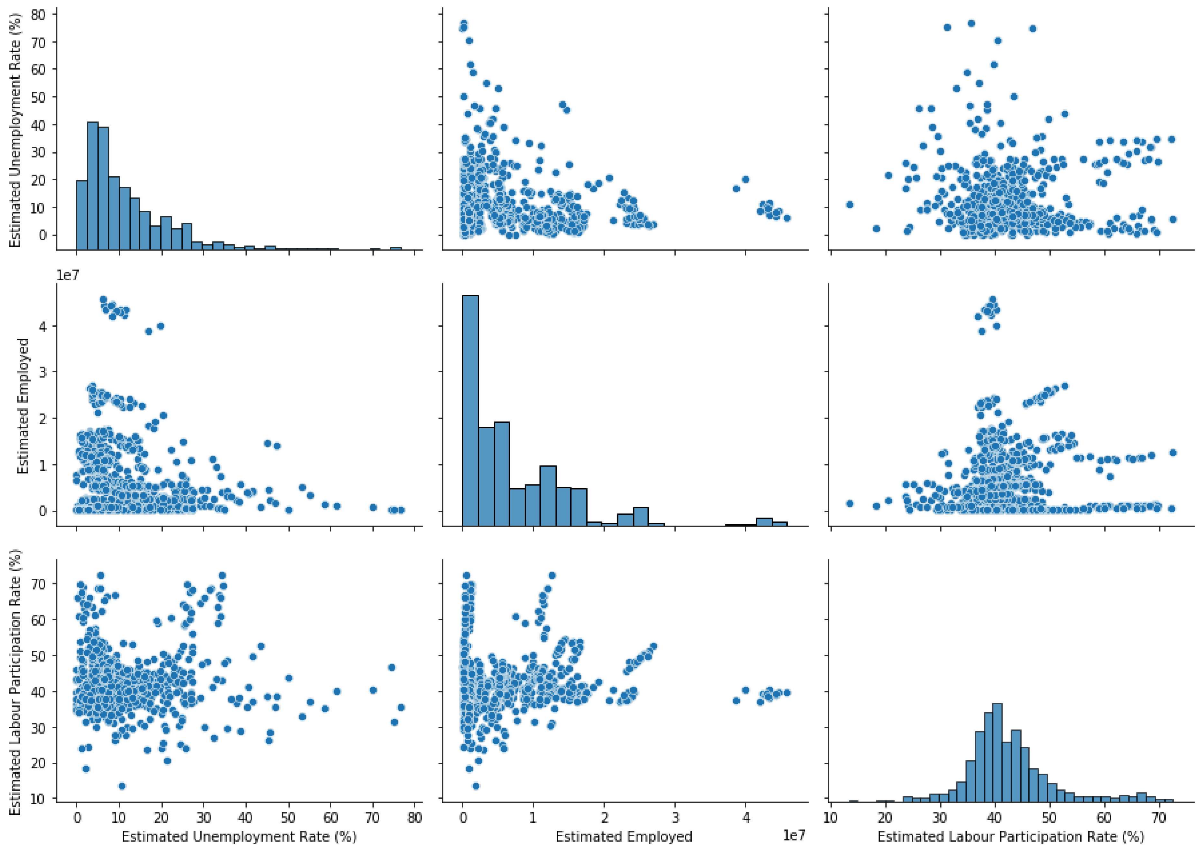
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x24180bbd888>



Analysis and Conclusion:

The violin plot shows that most regions have relatively low employment numbers, with employment mostly clustered in the lower range. Only a few regions have much higher employment figures.

In [30]:
```
# Plotting pairplot to study relationship between all the variables
Pair=df[['Estimated Unemployment Rate (%)','Estimated Employed','Estimated Lab
our Participation Rate (%)']]
sns.pairplot(Pair,height=3, aspect=1.4)
plt.tight_layout()
```



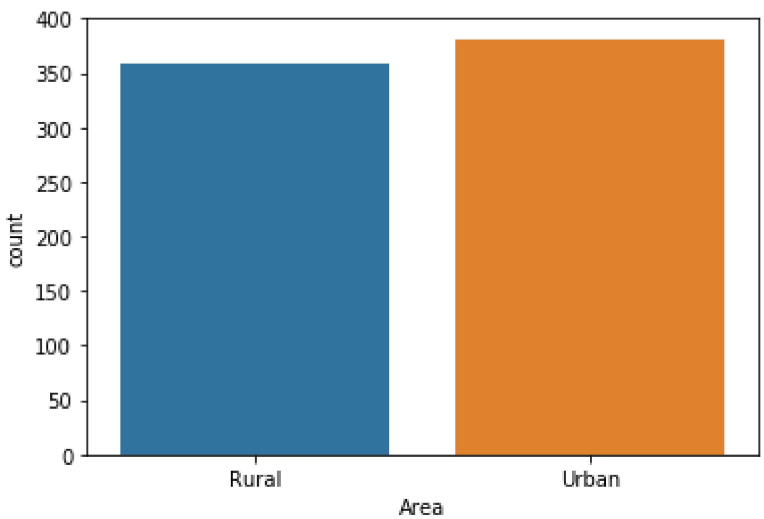
Analysis and Conclusion:

The scatter plots show that as the unemployment rate increases, both the number of employed individuals and the labor participation rate decrease. This means that higher unemployment is linked to fewer people working and fewer people participating in the labor force. To boost both employment and labor participation, it could be helpful to focus on reducing unemployment in India with targeted policies.

In [31]:
```
df["Area"].value_counts()
```

Out[31]:
```
Urban    381
Rural    359
Name: Area, dtype: int64
```

In [32]:
```
sns.countplot(x="Area",data=df)
plt.show()
```



Analysis and Conclusion:
The count plot shows more data from Urban areas than from Rural areas, suggesting a possible bias toward urban data. This might make it harder to fully understand the unemployment challenges in rural regions and could affect the overall findings.

In [33]:
```
df1.head()
```

Out[33]:

	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 [34]:
```
df1.tail()
```

Out[34]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latit
262	West Bengal	30-06-2020	M	7.29	30726310	40.39	East	22.9868	87.
263	West Bengal	31-07-2020	M	6.83	35372506	46.17	East	22.9868	87.
264	West Bengal	31-08-2020	M	14.87	33298644	47.48	East	22.9868	87.
265	West Bengal	30-09-2020	M	9.35	35707239	47.73	East	22.9868	87.
266	West Bengal	31-10-2020	M	9.98	33962549	45.63	East	22.9868	87.

In [35]:
```
df1.shape
```

Out[35]: (267, 9)

In [36]:
```
df1.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 [37]:
```
df1.Region.value_counts()
```

Out[37]:
```
Delhi               10
Jharkhand           10
Meghalaya           10
Tamil Nadu          10
Andhra Pradesh      10
Uttarakhand         10
Chhattisgarh        10
Karnataka           10
West Bengal         10
Himachal Pradesh    10
Maharashtra         10
Goa                 10
Puducherry          10
Bihar               10
Rajasthan           10
Odisha              10
Kerala              10
Gujarat             10
Madhya Pradesh      10
Tripura             10
Punjab              10
Uttar Pradesh       10
Assam               10
Telangana           10
Haryana             10
Jammu & Kashmir      9
Sikkim               8
Name: Region, dtype: int64
```

In [38]:
```
df1.isnull().sum()
```

Out[38]:
```
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 [39]:
```
df1.describe()
```

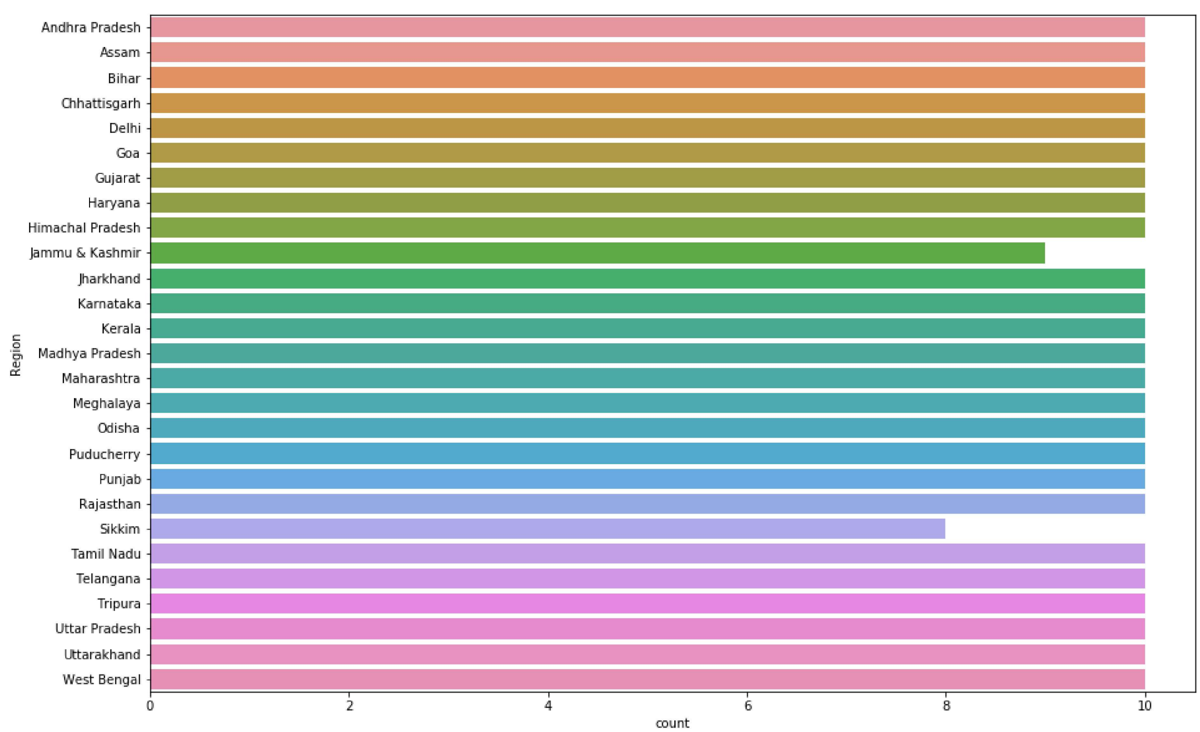
Out[39]:

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	longitude	latitude
count	267.000000	2.670000e+02	267.000000	267.000000	267.000000
mean	12.236929	1.396211e+07	41.681573	22.826048	80.532425
std	10.803283	1.336632e+07	7.845419	6.270731	5.831738
min	0.500000	1.175420e+05	16.770000	10.850500	71.192400
25%	4.845000	2.838930e+06	37.265000	18.112400	76.085600
50%	9.650000	9.732417e+06	40.390000	23.610200	79.019300
75%	16.755000	2.187869e+07	44.055000	27.278400	85.279900
max	75.850000	5.943376e+07	69.690000	33.778200	92.937600

In [40]:
```
#remove leading and trailing spaces from the column names
df1.columns = df1.columns.str.strip()
```

In [41]:
```
plt.figure(figsize=(15,10))
sns.countplot(y="Region",data=df1)
plt.show
```

Out[41]: <function matplotlib.pyplot.show(*args, **kw)>

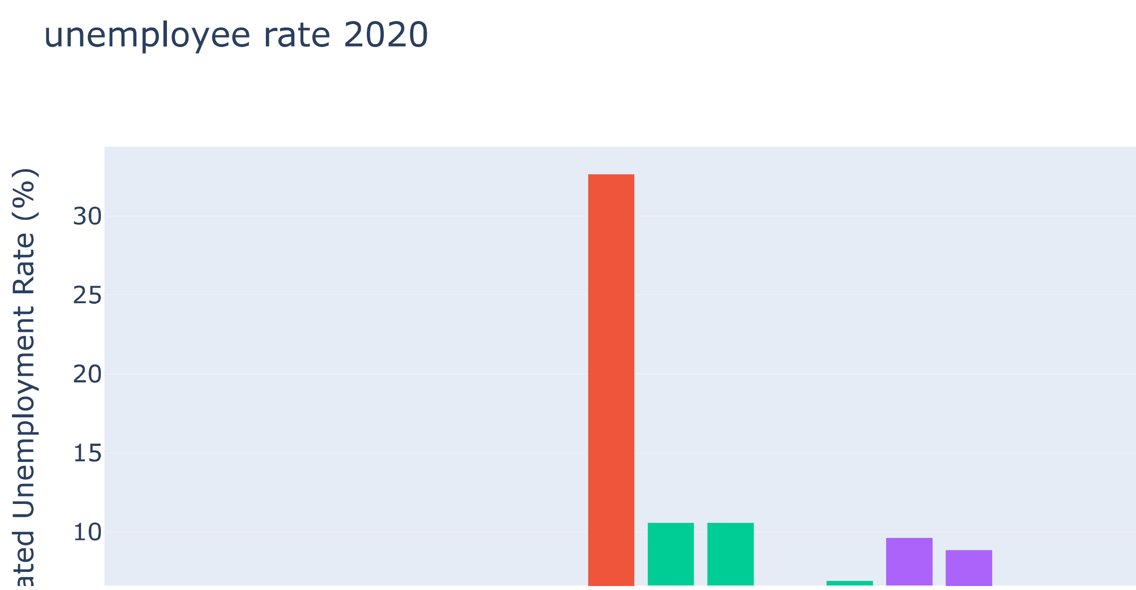


Analysis and Conclusion:
The count plot shows that some Indian regions have more data points than others. This uneven distribution could impact how well the analysis represents all regions. To get accurate and fair conclusions about regional unemployment patterns, it's important to consider this imbalance in the data.

In [42]:
```
import plotly.express as go
fig = go.bar(df1, x="Region", y="Estimated Employed", title="Estimated Employe
d",
             animation_frame='Date',template='plotly',color="Region")
fig.show()
```

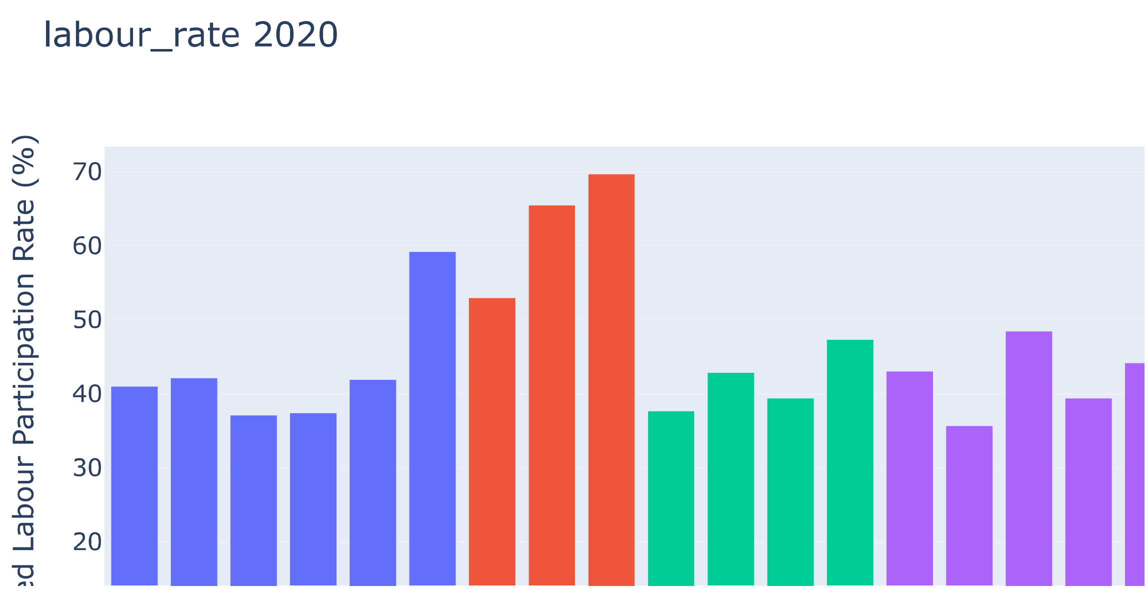
Analysis and Conclusion:
The bar chart shows that from January to October 2020, the "North" region had much higher employment levels compared to other states. This suggests we need to investigate why the "North" is performing better and create policies to boost employment in other regions while maintaining the gains in the "North."

In [43]:
```
fig = go.bar(df1, x="Region", y="Estimated Unemployment Rate (%)", title="unem
ployee rate 2020",
             animation_frame='Date',template='plotly',color="Region.1")
fig.show()
```



Analysis and Conclusion:
The bar chart reveals that unemployment rates varied widely across Indian states from January to October 2020. While many states saw changes, some consistently had higher unemployment. This suggests we need to find out which regions are struggling the most and understand why, so we can create targeted policies to address these issues.

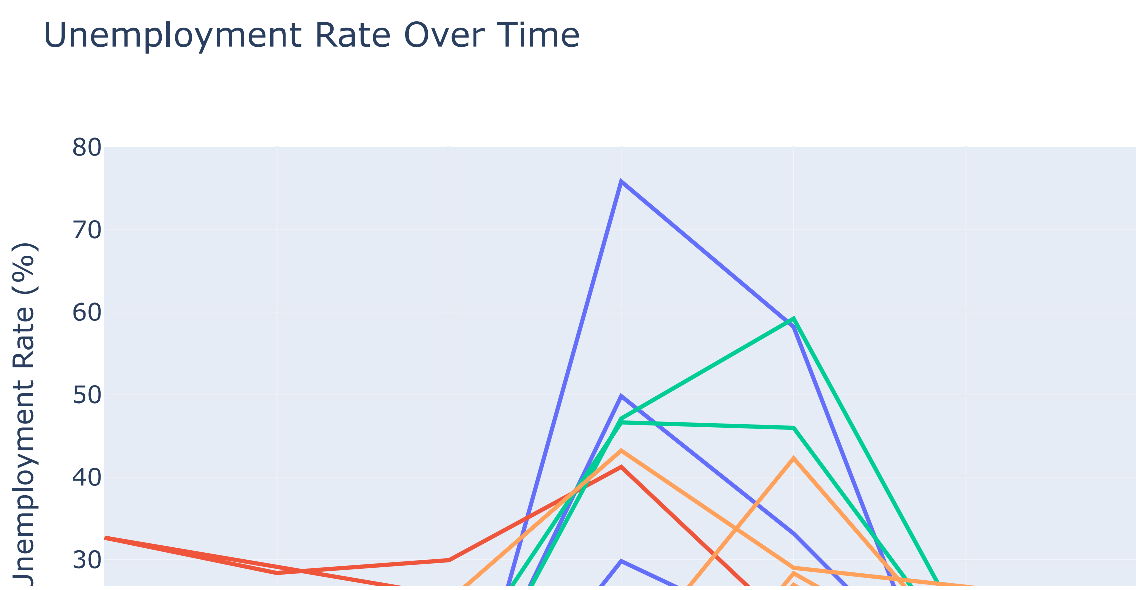
In [44]:
```
fig = go.bar(df1, x="Region", y="Estimated Labour Participation Rate (%)", tit
le="labour_rate 2020",
             animation_frame='Date',template='plotly',color="Region.1")
fig.show()
```



Analysis and Conclusion:
The bar charts reveal that "Telangana," "Assam," "Meghalaya," and "Tripura" consistently have higher labor participation rates, often exceeding 50% in the second half of 2020. In contrast, "Kerala" and "Puducherry" show lower participation rates, generally below 30%. While most regions maintain participation rates between 40% and 60%, the significant disparities highlight stronger workforce engagement in certain regions and notable regional differences in labor participation.

Time Series Line Plot for Unemployment Rate by Region

In [45]:
```
fig = go.line(df1, x='Date', y="Estimated Unemployment Rate (%)", color='Region.1',
                    title='Unemployment Rate Over Time', template='plotly')
fig.show()
```



Analysis and Conclusion:
The line plot shows that unemployment rates varied over time, with sharp increases around mid-2020, especially in the Northeast and North, likely due to COVID-19 disruptions. Other regions had more stable unemployment levels, showing different impacts and resilience.

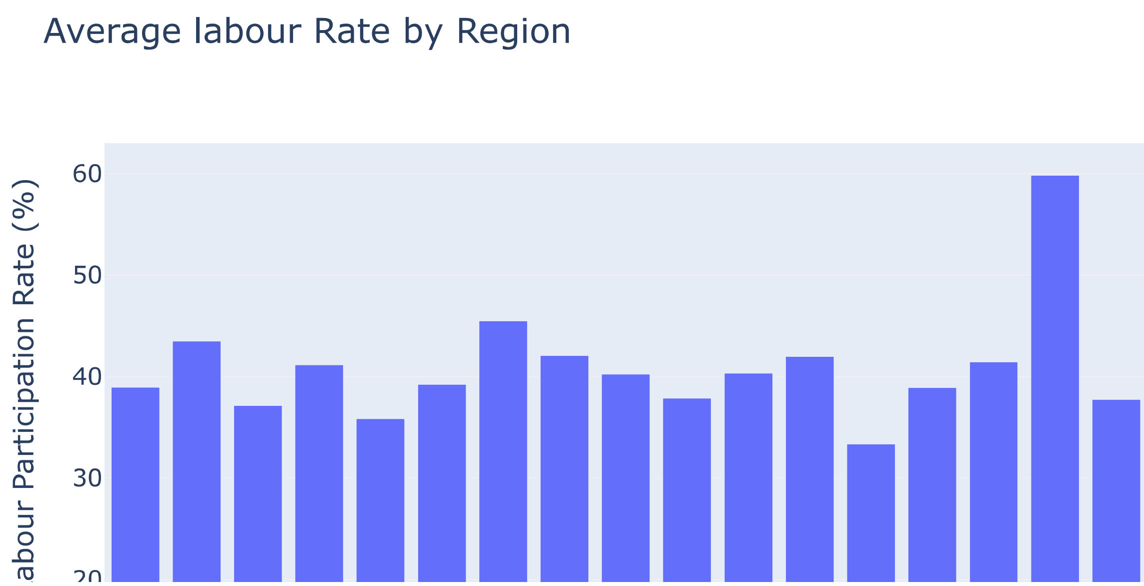
Bar Plot for Average Unemployment Rate by Region

In [46]:
```
avg_unemployment = df1.groupby('Region')['Estimated Unemployment Rate (%)'].me
an().reset_index()

bar_plot = go.bar(avg_unemployment, x='Region', y='Estimated Unemployment Rate
(%)',
                     title='Average Unemployment Rate by Region', template='plot
ly')
bar_plot.show()
```

Analysis and Conclusion:
The bar plot displays significant variability in average unemployment rates across regions. Regions such as "Tripura" and "Haryana" exhibit notably higher average rates, indicating they may have faced more severe economic challenges. Conversely, regions like "Assam" and "Meghalaya" show lower average rates, suggesting relatively better employment conditions. This information can assist policymakers in crafting tailored employment strategies for different regions.

Bar Plot for Average labour Rate by Region

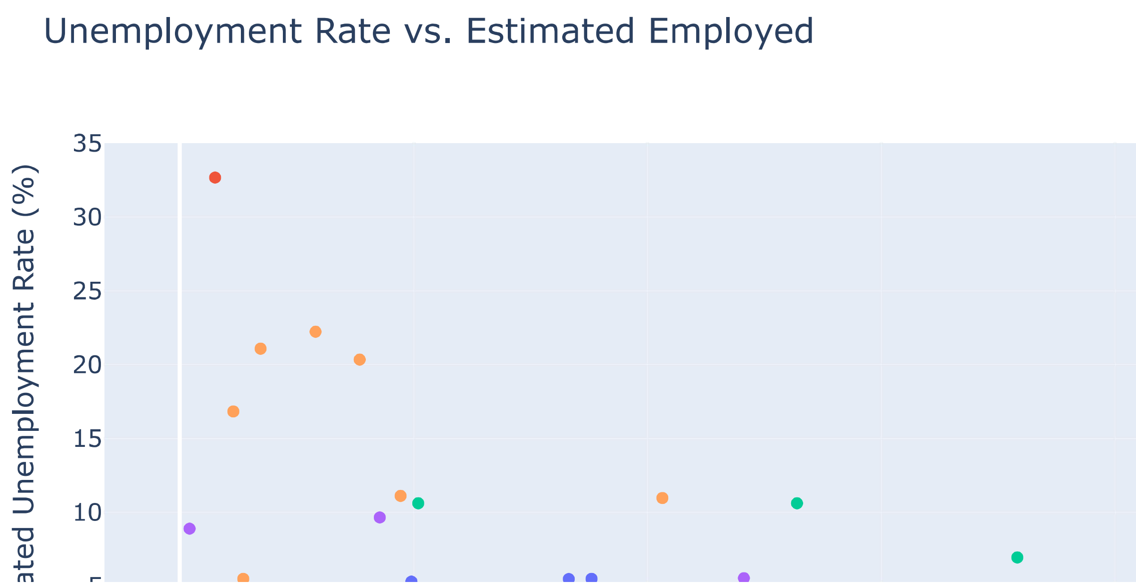
In [47]:
```
avg_unemployment = df1.groupby('Region')['Estimated Labour Participation Rate (%)'].mean().reset_index()

bar_plot = go.bar(avg_unemployment, x='Region', y='Estimated Labour Participation Rate (%)',
                   title='Average labour Rate by Region', template='plotly')
bar_plot.show()
```



alysis and Conclusion:
The bar plot reveals variability in average labor participation rates across regions, with some regions showing notably higher or lower rates. "Meghalaya" has the highest average labor participation rate, whereas "Kerala," "Delhi," and "Uttarakhand" have relatively lower rates. Higher participation rates may indicate more active labor markets, while lower rates could signal challenges in workforce engagement. Policymakers may need to address these challenges to enhance participation and overall economic activity in lower-performing regions.

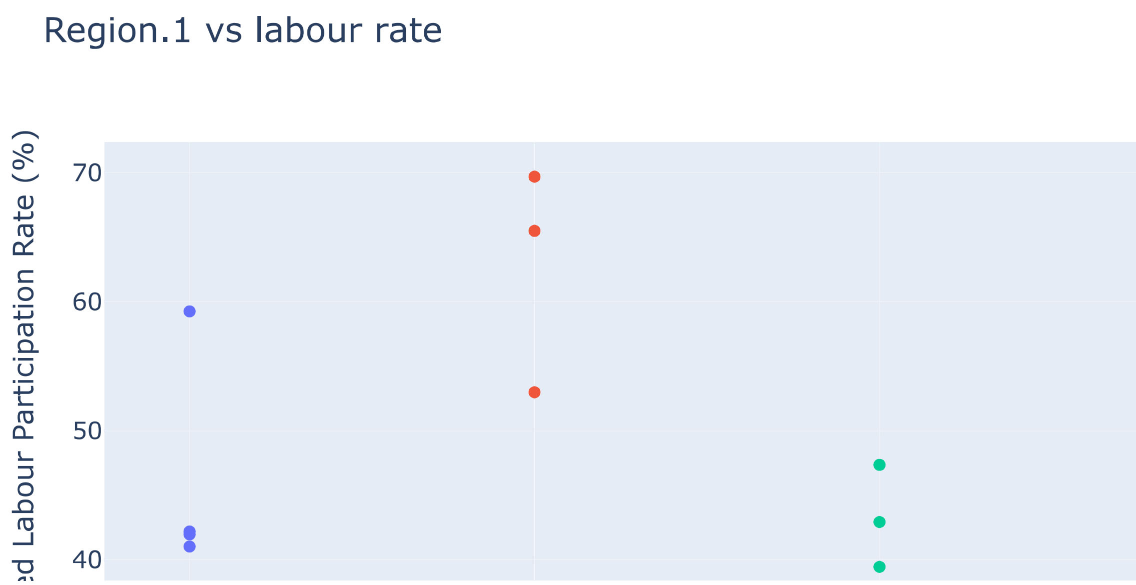
Visualisation of Scatter Plot for Unemployment Rateand Estimated Employed

In [48]:
```
fig = go.scatter(df1, x='Estimated Employed', y='Estimated Unemployment Rate (%)',
                        color='Region.1', title='Unemployment Rate vs. Estimated Employed',animation_frame="Date")
fig.show()
```



Analysis and Conclusion:
The data indicates a general trend where regions with a higher number of employed individuals tend to have lower unemployment rates. However, there is notable variation within regions, suggesting that other factors beyond the total number of employed individuals also influence unemployment rates. This illustrates how unemployment rates can differ across regions.

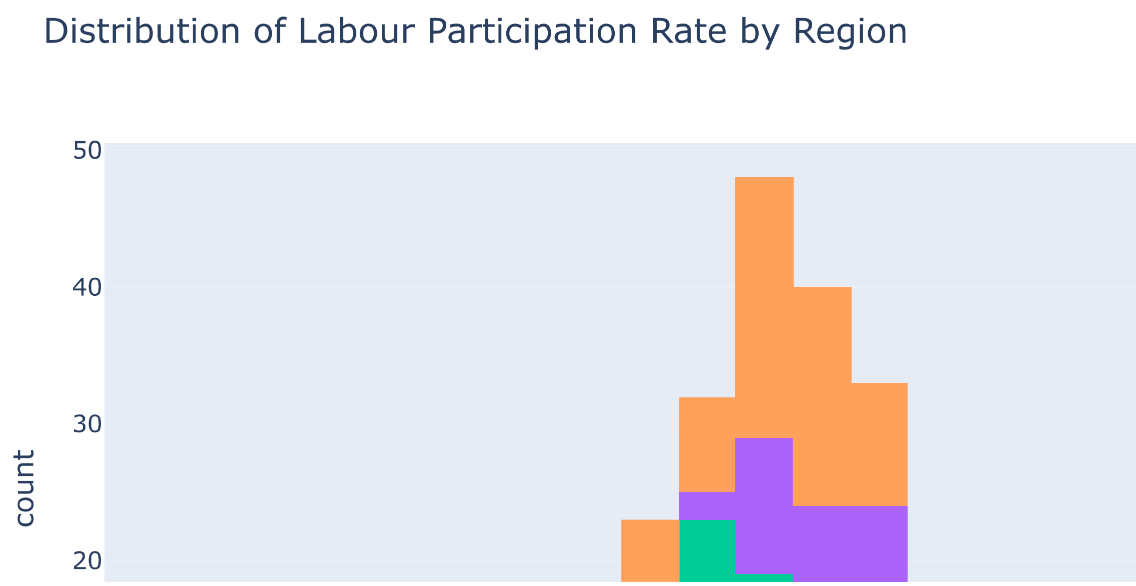
In [49]:
```
fig = go.scatter(df1, x='Region.1', y='Estimated Labour Participation Rate (%)',
                              color='Region.1', title='Region.1 vs labour rate',animation_frame="Date")
fig.show()
```



Analysis and Conclusion:
The Northeast region consistently shows high estimated labor participation rates throughout the period, indicating strong labor force engagement. This trend may be influenced by factors such as favorable economic conditions, cultural norms, and supportive government policies.

Visualisation of Histogram for Estimated Labour Participation Rate

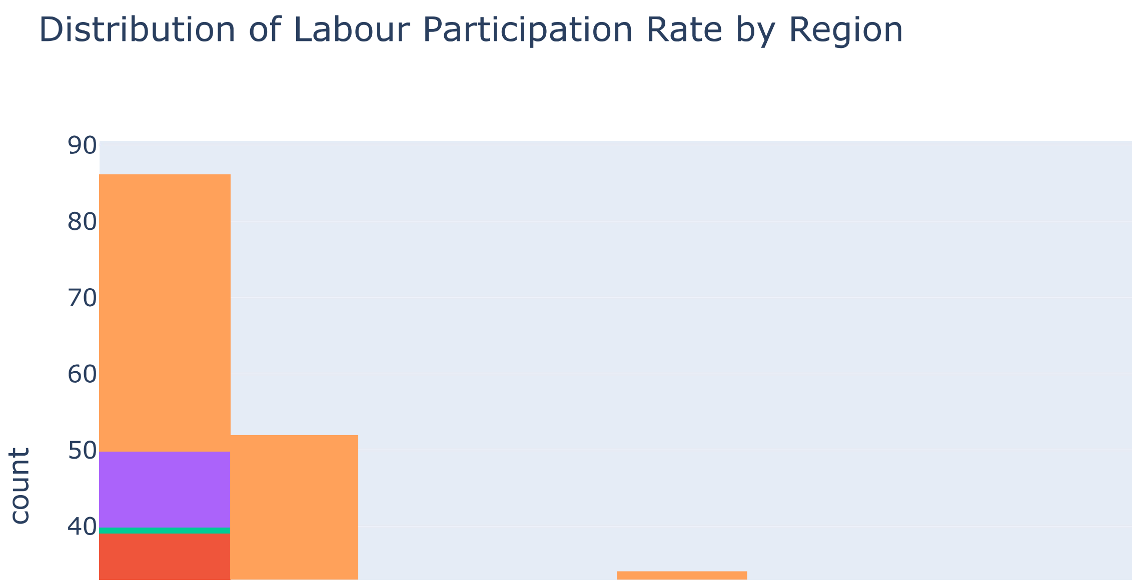
In [50]:
```
fig = go.histogram(df1, x='Estimated Labour Participation Rate (%)', color='Region.1',
                         title='Distribution of Labour Participation Rate by Region', template='plotly')
fig.show()
```



Analysis and Conclusion:
The histogram shows that the Northeast region has the highest concentration of observations in the upper ranges of labor participation rates, indicating a larger proportion of individuals actively engaged in the workforce compared to other regions. This suggests that factors such as a strong work ethic or economic necessity may drive higher labor participation in this region.

Visualisation of Histogram for Estimated Employed

In [51]:
```
fig = go.histogram(df1, x='Estimated Employed', color='Region.1',
                             title='Distribution of Labour Participation Rate by R
egion', template='plotly')
fig.show()
```

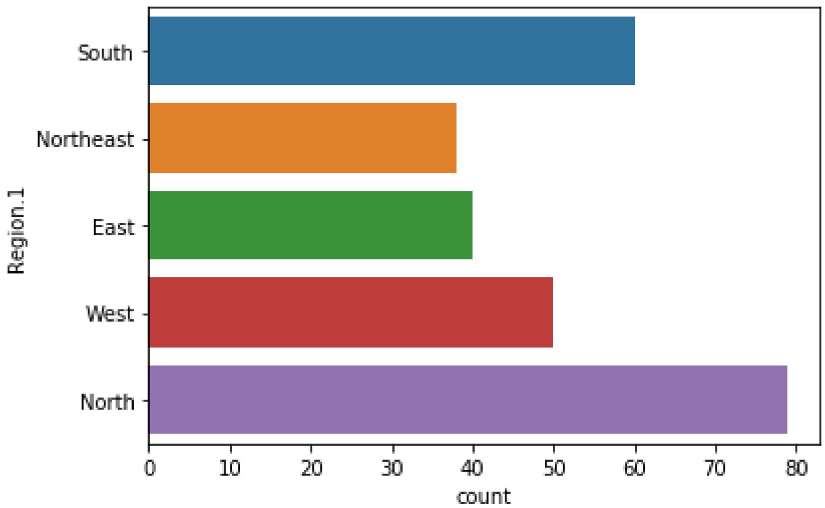


Analysis and Conclusion:
The grouped histogram shows that the Northeast region consistently has the highest proportion of individuals across various labor participation rate ranges, indicating widespread engagement in the labor force. In contrast, regions like the South and West have a higher concentration of individuals in lower participation rate ranges, suggesting a greater proportion of people are not actively engaged in the labor force.

In [52]:
```
df1["Region.1"].value_counts()
```

Out[52]:
```
North        79
South        60
West         50
East         40
Northeast    38
Name: Region.1, dtype: int64
```

In [53]:
```
sns.countplot(y="Region.1",data=df1)
plt.show()
```



Analysis and Conclusion:
The bar plot illustrates that the North region has the highest number of observations, followed by the South and West regions. The Northeast and East regions have relatively lower counts. This indicates that the North region is the most prominently represented in the dataset, with varying levels of representation in other regions.

The End