In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
%matplotlib inline
```

In [2]:
```
df=pd.read_csv("Unemployment_in_India.csv")
df.head()
```

Out[2]:

	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 [3]:
```
df[' Date'] = pd.to_datetime(df[' Date'])
```

In [4]:
```
df.shape
```

Out[4]:
```
(768, 7)
```

In [5]:
```
df.describe()
```

Out[5]:

	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

In [6]:
```
df=df.dropna()
```

In [7]:
```
df.shape
```

Out[7]:
```
(740, 7)
```

In [8]:
```
Regions=df['Region'].unique()
Regions
```

Out[8]:
```
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', 'Chandigarh'],
      dtype=object)
```

In [9]:
```
Area=df['Area'].unique()
Area
```

Out[9]:
```
array(['Rural', 'Urban'], dtype=object)
```

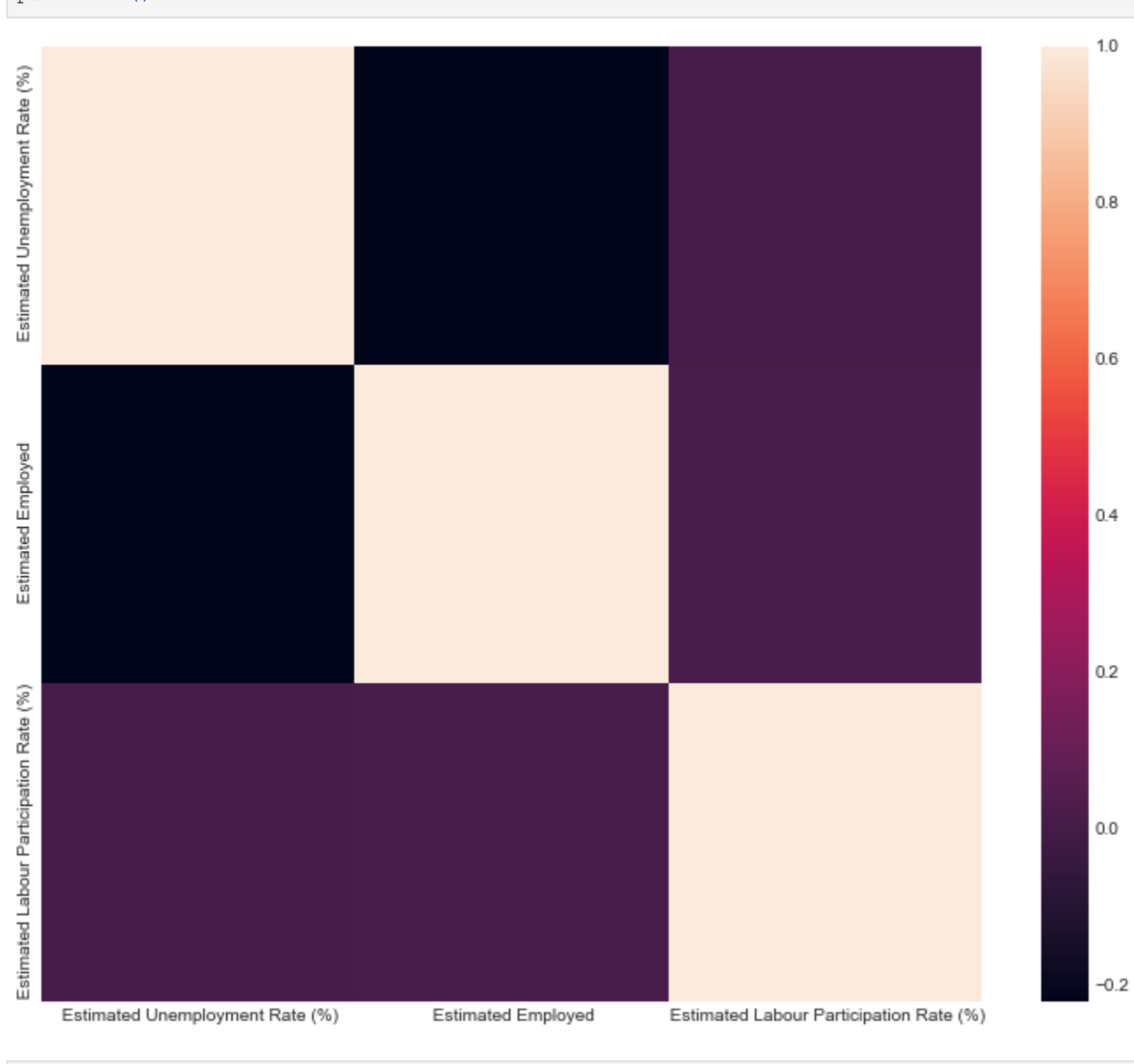
In [10]:
```
# convert column 'Date' to datetime format
df[' Date'] = pd.to_datetime(df[' Date'])

# extract range of dates
date_range = df[' Date'].max() - df[' Date'].min()

# display range
print(date_range)
```

```
396 days 00:00:00
```

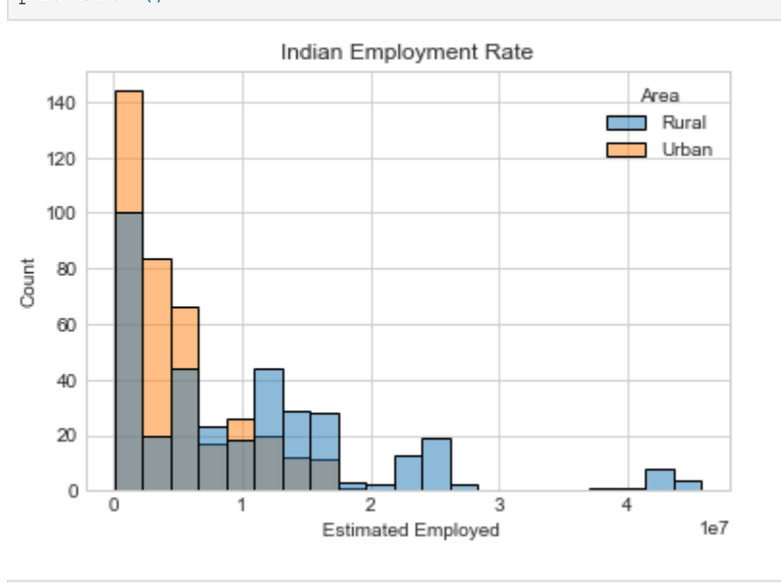
In [11]:
```
print(df.isnull().sum())
```

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

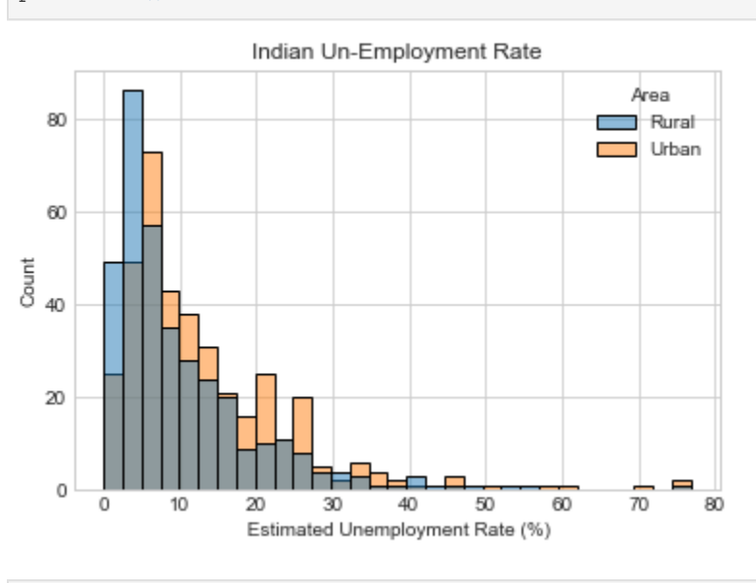
In [12]:
```
plt.style.use("seaborn-whitegrid")
plt.figure(figsize=(12,10))
sns.heatmap(df.corr())
plt.show()
```



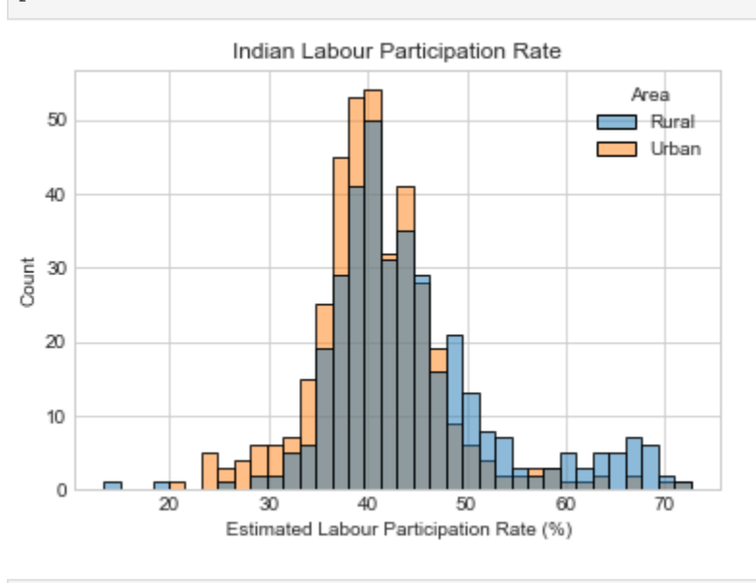
In [13]:
```
plt.title("Indian Employment Rate")
sns.histplot(x=" Estimated Employed",hue="Area",data=df)
plt.show()
```



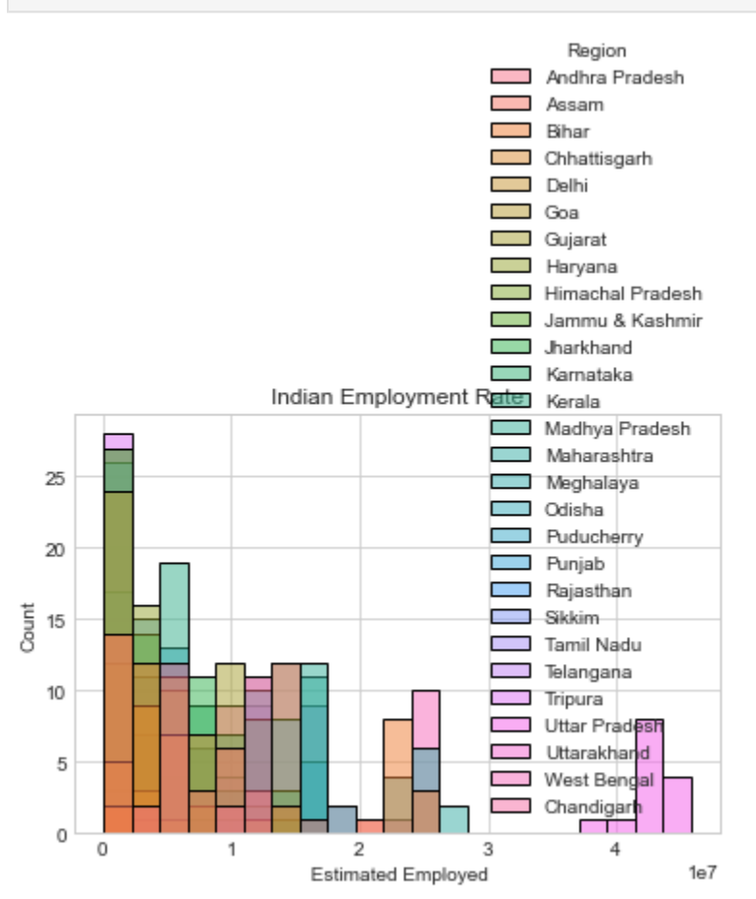
In [14]:
```
plt.title("Indian Un-Employment Rate")
sns.histplot(x=" Estimated Unemployment Rate (%)",hue="Area",data=df)
plt.show()
```



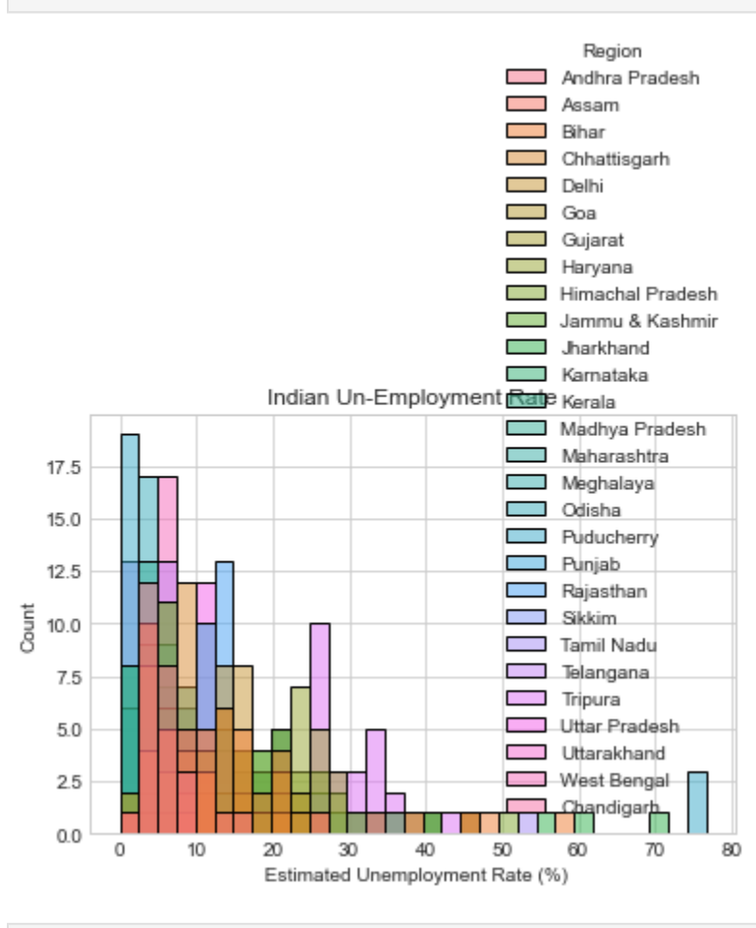
In [15]:
```
plt.title("Indian Labour Participation Rate")
sns.histplot(x=" Estimated Labour Participation Rate (%)",hue="Area",data=df)
plt.show()
```



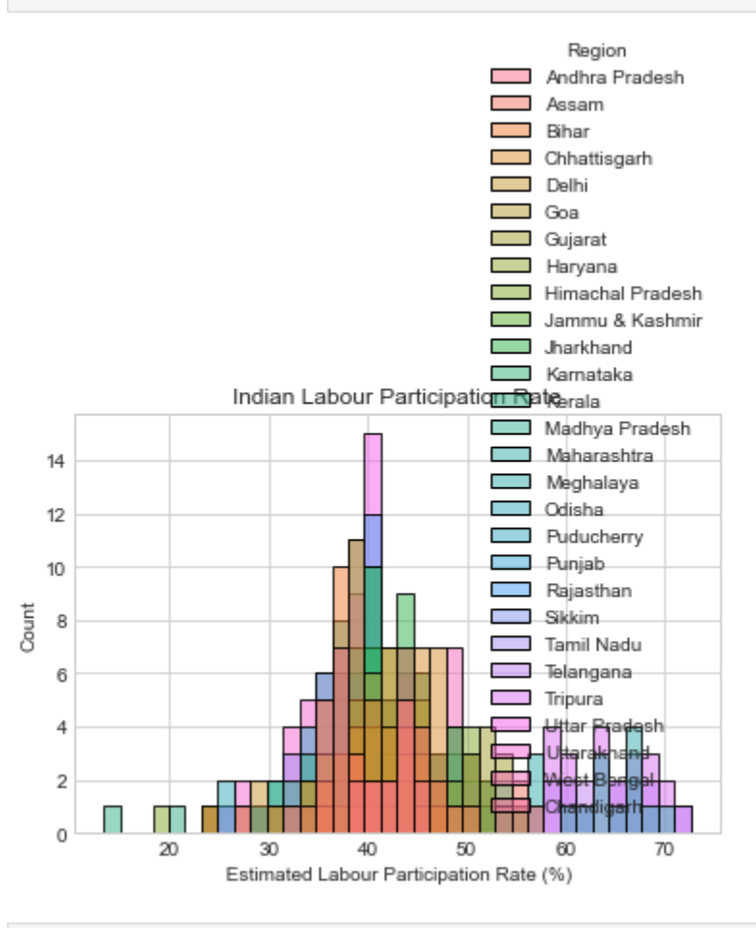
In [16]:
```
plt.title("Indian Employment Rate")
sns.histplot(x=" Estimated Employed",hue="Region",data=df)
plt.show()
```



In [17]:
```
plt.title("Indian Un-Employment Rate")
sns.histplot(x=" Estimated Unemployment Rate (%)",hue="Region",data=df)
plt.show()
```



In [18]:
```
plt.title("Indian Labour Participation Rate")
sns.histplot(x=" Estimated Labour Participation Rate (%)",hue="Region",data=df)
plt.show()
```

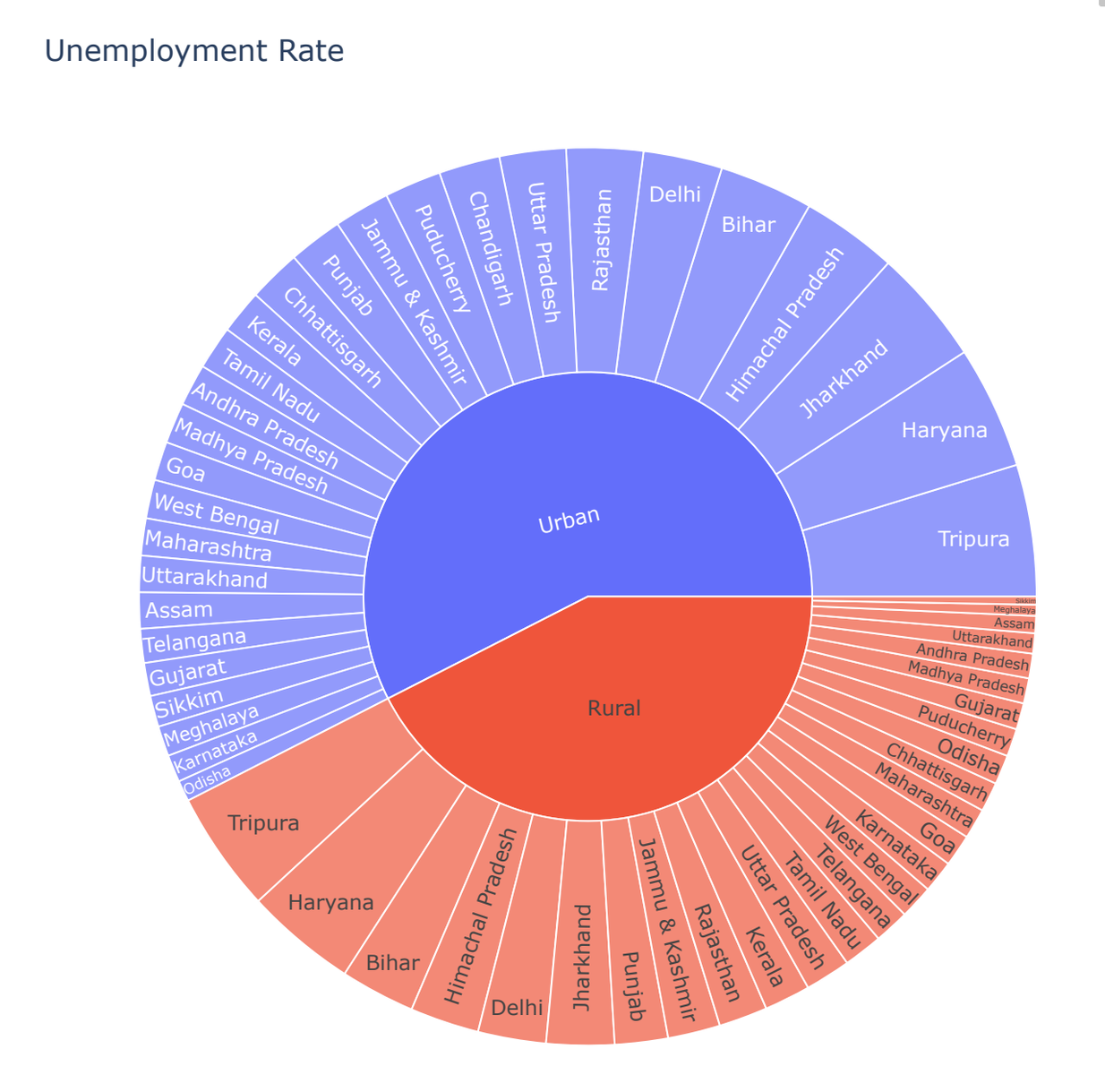


In [19]:
```
df.columns
```

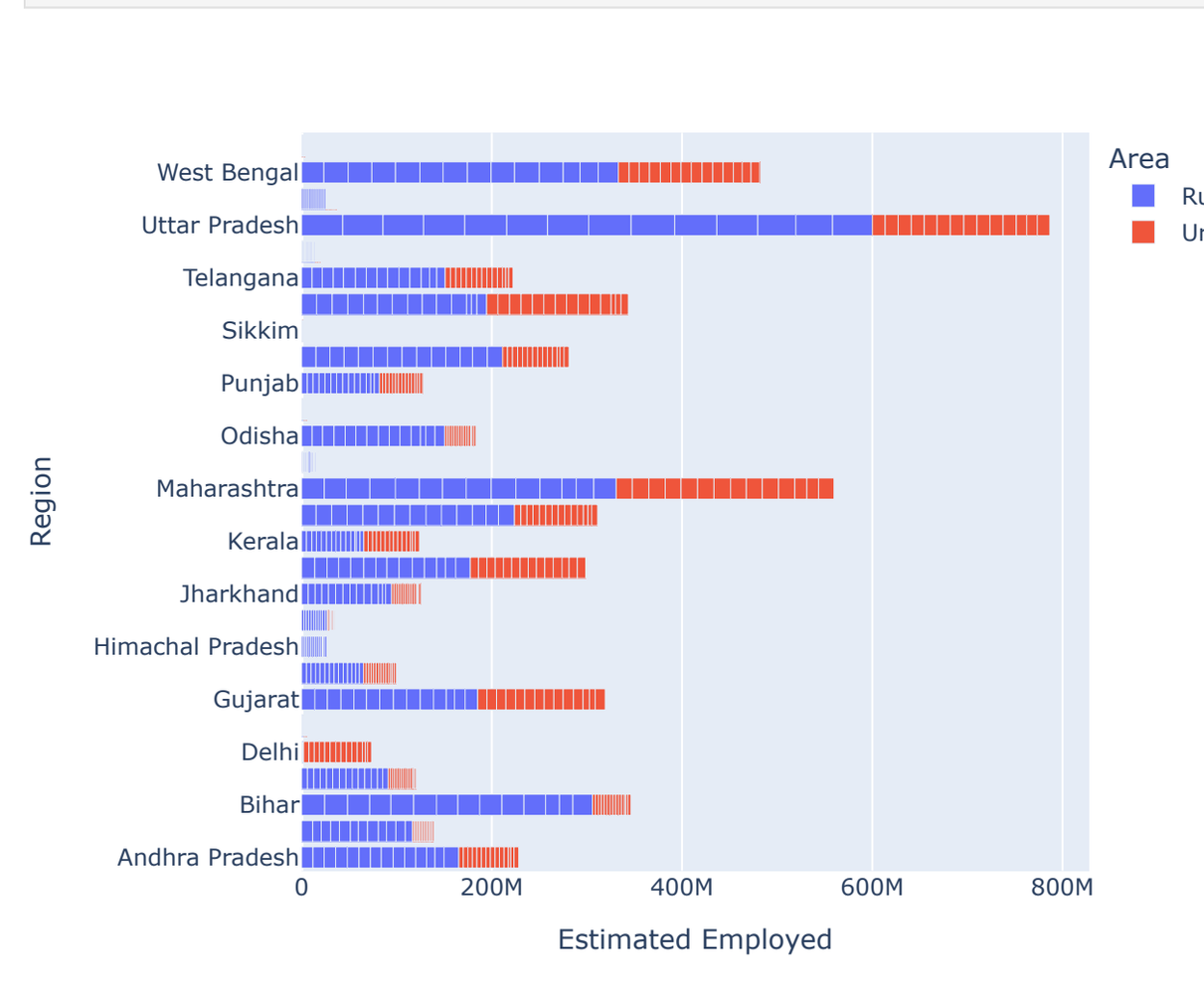
Out[19]:
```
Index(['Region', ' Date', ' Frequency', ' Estimated Unemployment Rate (%)',
       ' Estimated Employed', ' Estimated Labour Participation Rate (%)',
       'Area'],
      dtype='object')
```

In [20]:
```
unemployment=df[["Region","Area"," Estimated Unemployment Rate (%)"]]
figure=px.sunburst(unemployment,path=["Area","Region"],values=" Estimated Unemployment Rate (%)",width=700,height=700,color_continuous_scale="RdYlGn",title="Unemployment Rate")
figure.show()
```

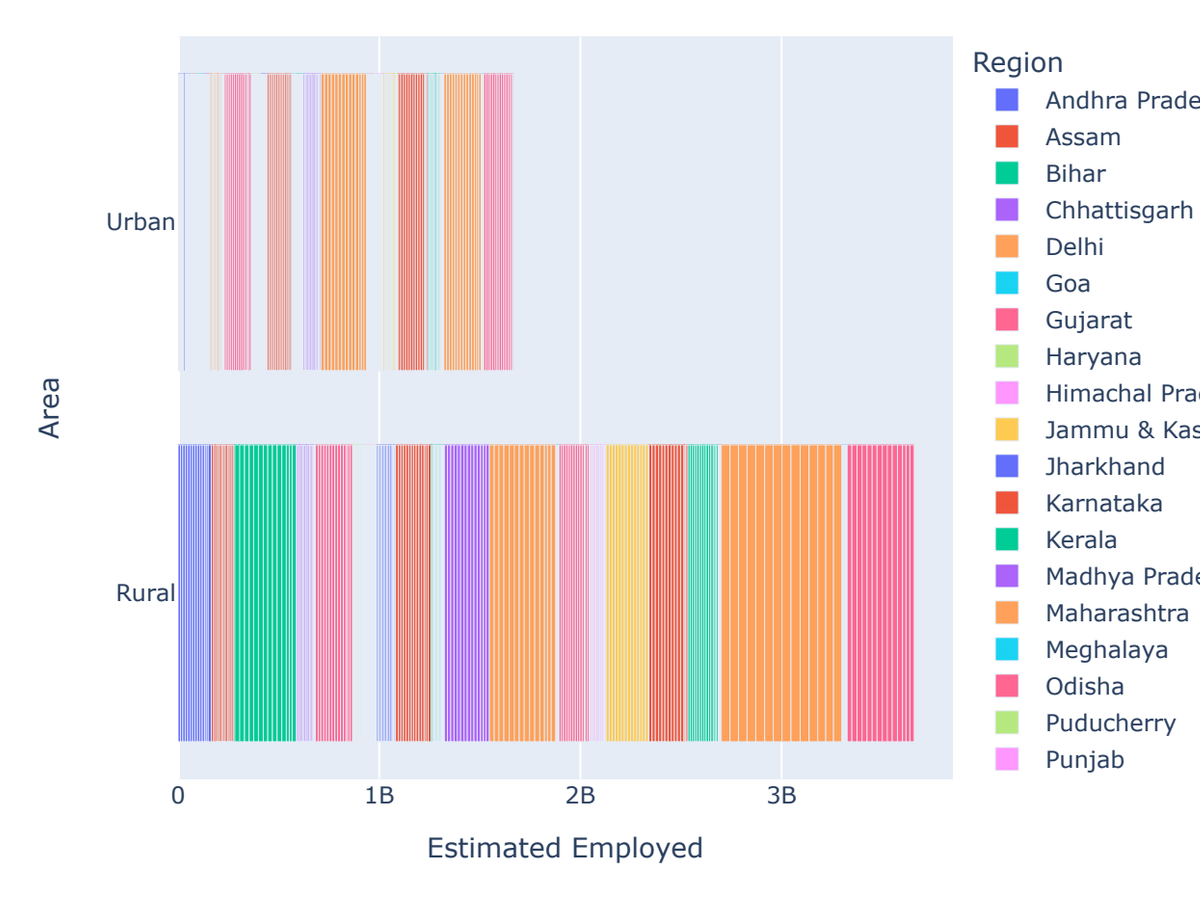
```
C:\Users\vinay\anaconda3\lib\site-packages\plotly\express\_core.py:1637: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df_all_trees = df_all_trees.append(df_tree, ignore_index=True)
C:\Users\vinay\anaconda3\lib\site-packages\plotly\express\_core.py:1637: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df_all_trees = df_all_trees.append(df_tree, ignore_index=True)
```



In [21]:
```
# Creating the bar chart
fig = px.bar(df, x=' Estimated Employed', y="Region", color='Area')

fig.show()
```

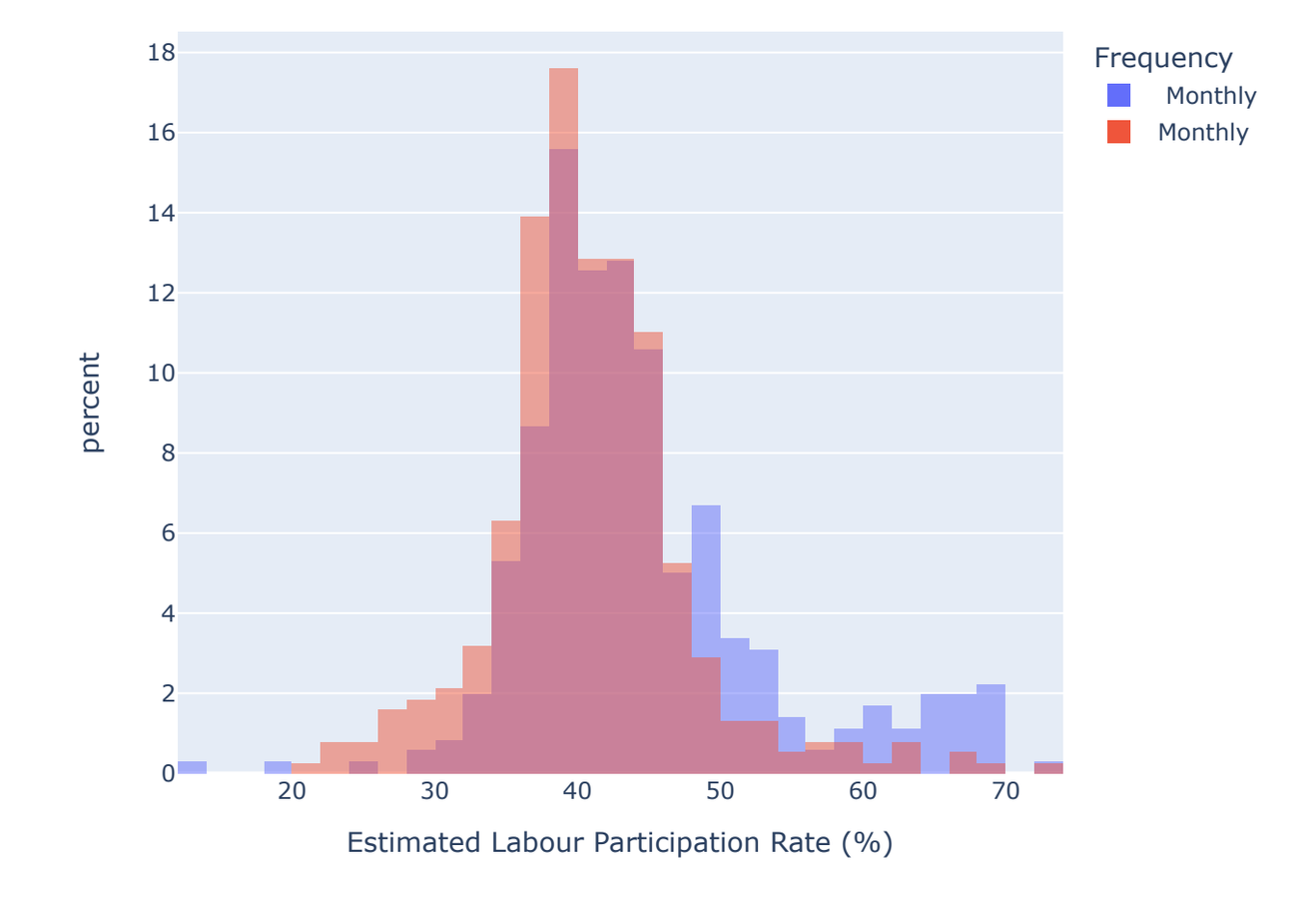


In [22]:
```
# Creating the bar chart
fig = px.bar(df, x=' Estimated Employed', y="Area", color='Region')

fig.show()
```

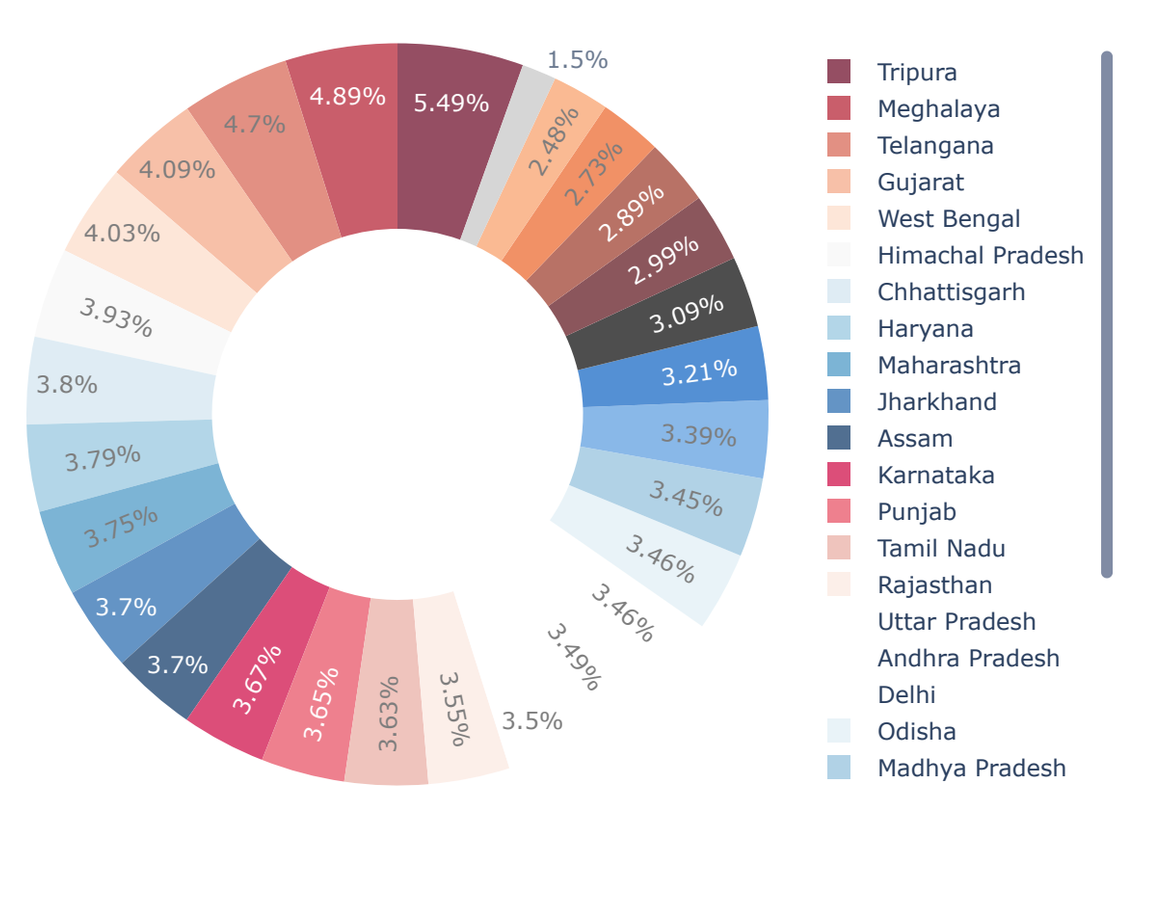


In [23]:
```
# plotting the scatter chart
fig = px.scatter(df, x=' Estimated Unemployment Rate (%)', y="Region", color='Area',symbol=' Frequency')

# showing the plot
fig.show()
```

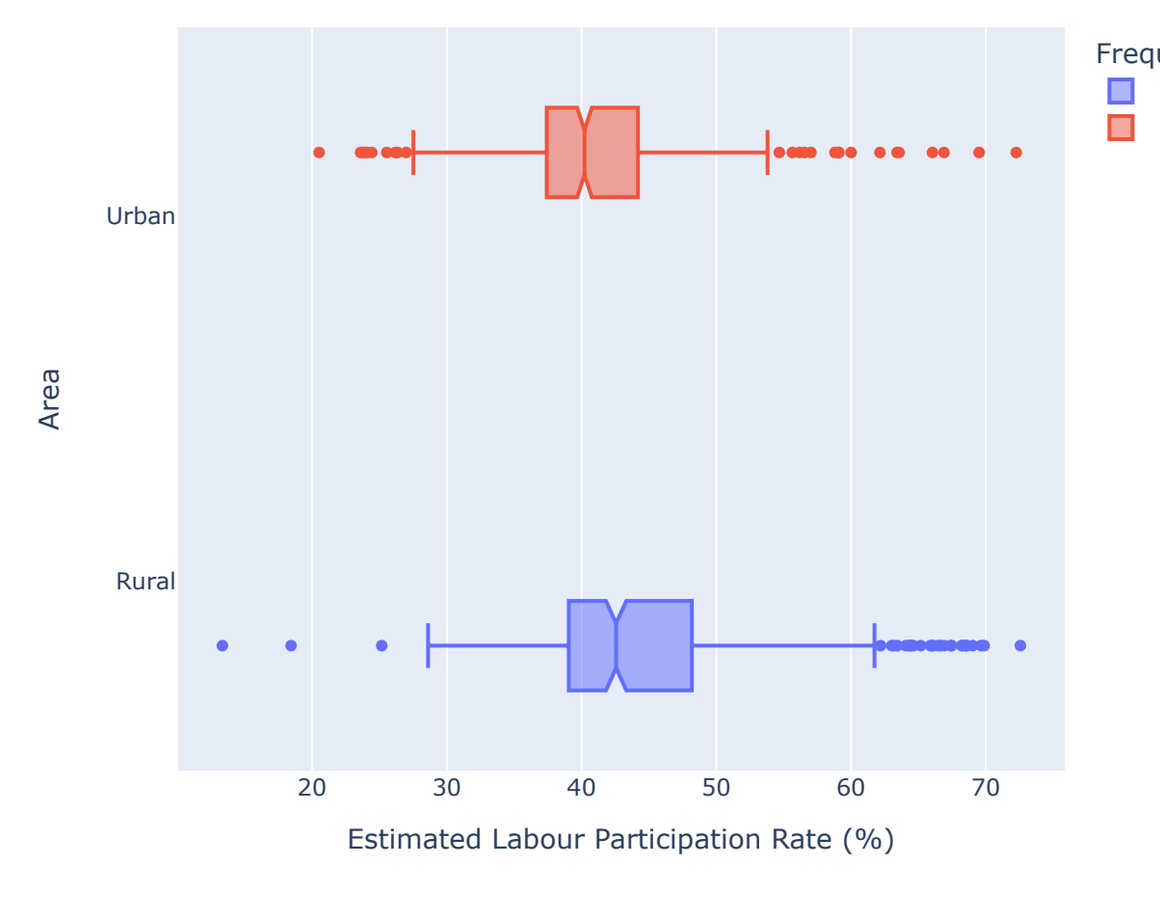
In [24]:
```
# plotting the histogram
fig = px.histogram(df, x=" Estimated Labour Participation Rate (%)", color=' Frequency',nbins=50, histnorm='percent',barmode='overlay')

# showing the plot
fig.show()
```



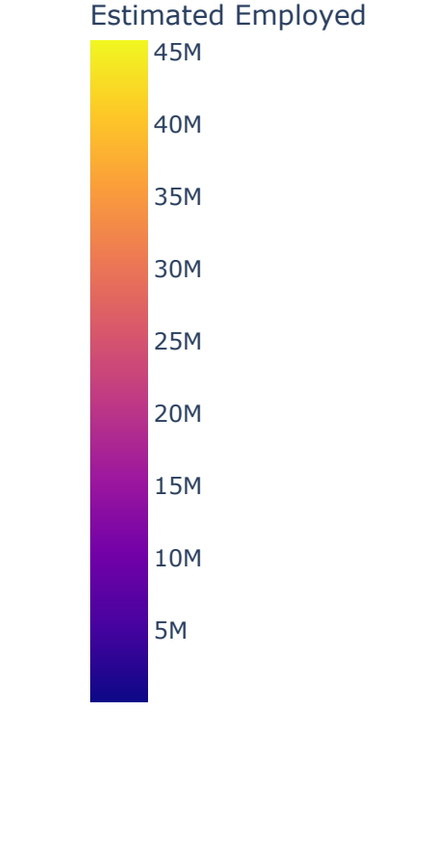
In [25]:
```
fig = px.pie(df, values=" Estimated Labour Participation Rate (%)", names="Region",color_discrete_sequence=px.colors.sequential.RdBu,opacity=0.7, hole=0.5)
fig.show()
```



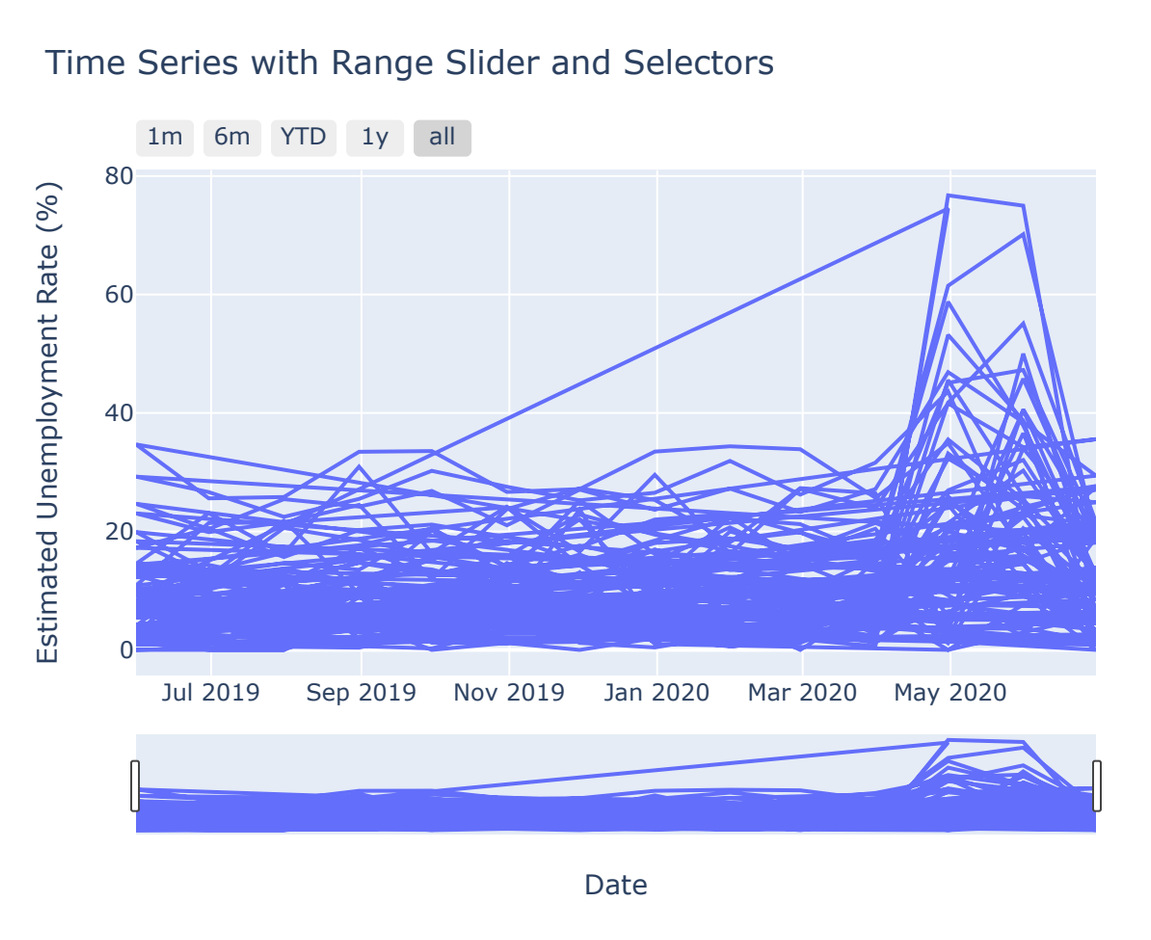
In [26]:
```
# plotting the boxplot
fig = px.box(df, x=" Estimated Labour Participation Rate (%)", y="Area", color=' Frequency',boxmode='group',notched=True)

# showing the plot
fig.show()
```



In [27]:
```
# plotting the figure
fig = px.scatter_3d(df, x=" Estimated Labour Participation Rate (%)", y="Region", z="Area", color=' Estimated Employed')
fig.show()
```



In [28]:
```
fig = px.line(df, x=' Date', y=' Estimated Unemployment Rate (%)', title='Time Series with Range Slider and Selectors')

fig.update_xaxes(
    rangeslider_visible=True,
    rangeselector=dict(
        buttons=list([
            dict(count=1, label="1m", step="month", stepmode="backward"),
            dict(count=6, label="6m", step="month", stepmode="backward"),
            dict(count=1, label="YTD", step="year", stepmode="todate"),
            dict(count=1, label="1y", step="year", stepmode="backward"),
            dict(step="all")
        ])
    )
)
fig.show()
```



In [ ]: