

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

```
In [2]: df=pd.read_csv("Unemployment in India1.csv")
```

```
In [3]: df
```

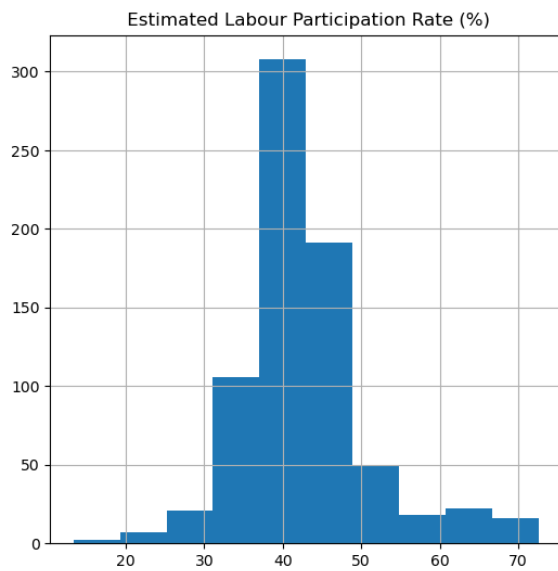
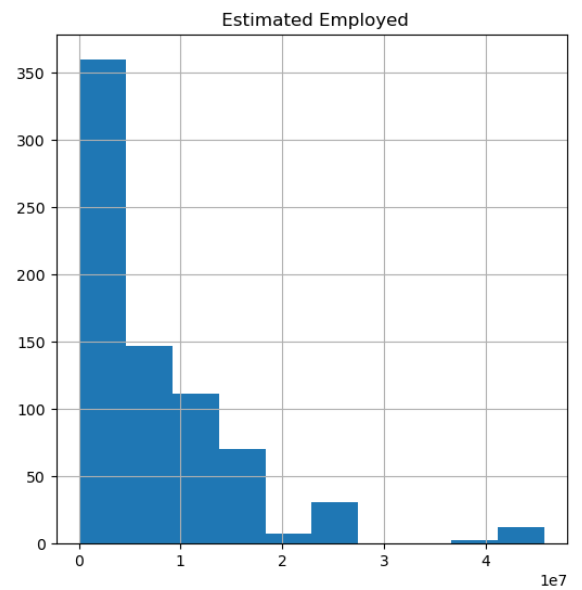
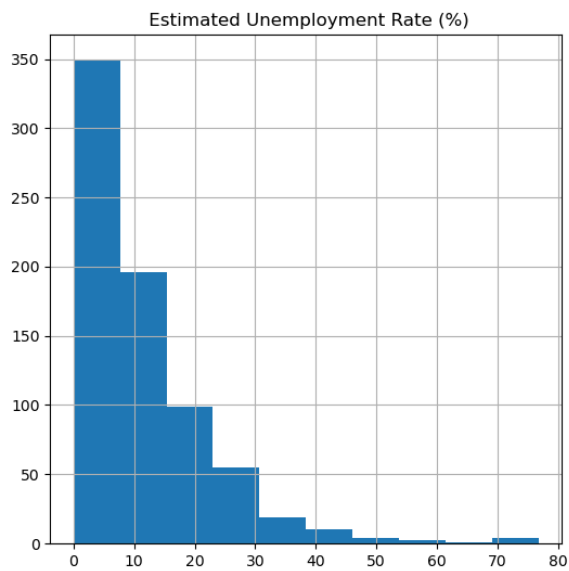
```
Out[3]:
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area
<b>0</b>	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139	43.24	Rural
<b>1</b>	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881	42.05	Rural
<b>2</b>	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707	43.50	Rural
<b>3</b>	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693	43.97	Rural
<b>4</b>	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762	44.68	Rural
...	...	...	...	...	...	...	...
<b>735</b>	West Bengal	29-02-2020	Monthly	7.55	10871168	44.09	Urban
<b>736</b>	West Bengal	31-03-2020	Monthly	6.67	10806105	43.34	Urban
<b>737</b>	West Bengal	30-04-2020	Monthly	15.63	9299466	41.20	Urban
<b>738</b>	West Bengal	31-05-2020	Monthly	15.22	9240903	40.67	Urban
<b>739</b>	West Bengal	30-06-2020	Monthly	9.86	9088931	37.57	Urban

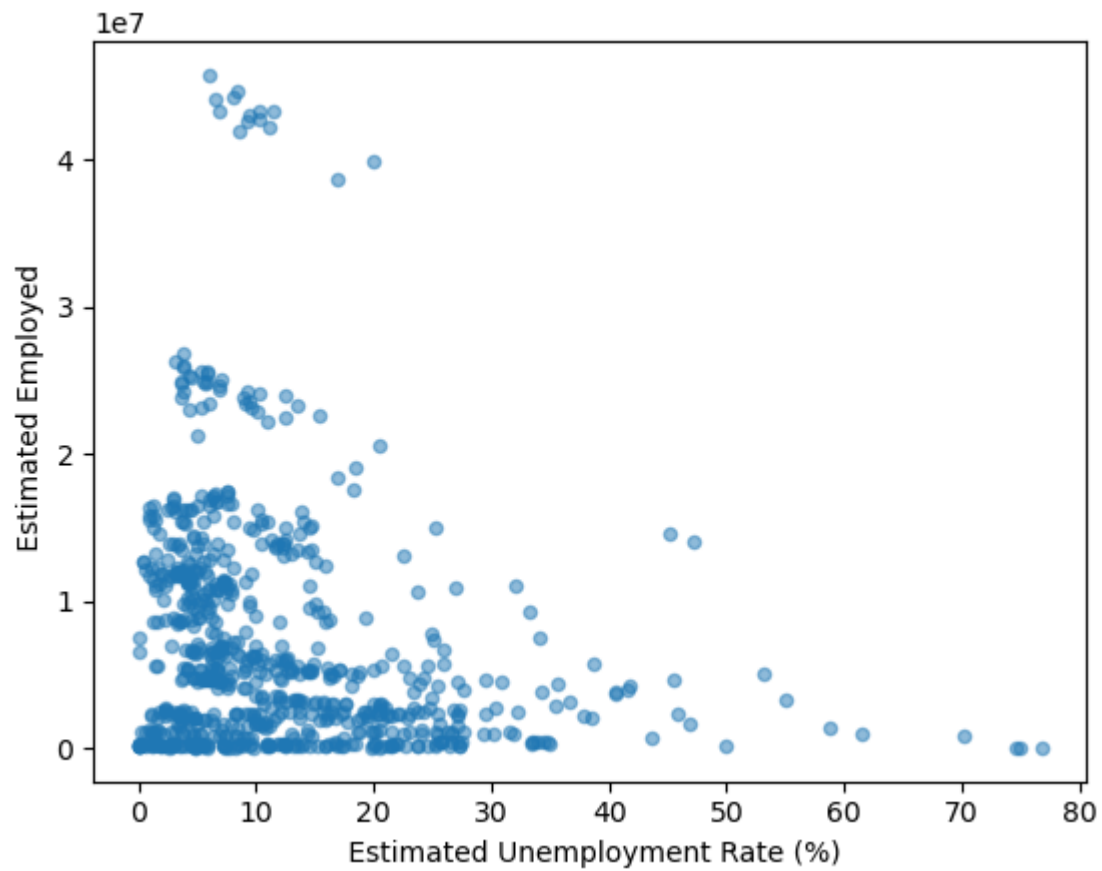
740 rows × 7 columns

```
In [4]: df.hist(figsize=(14,14))
plt.show
```

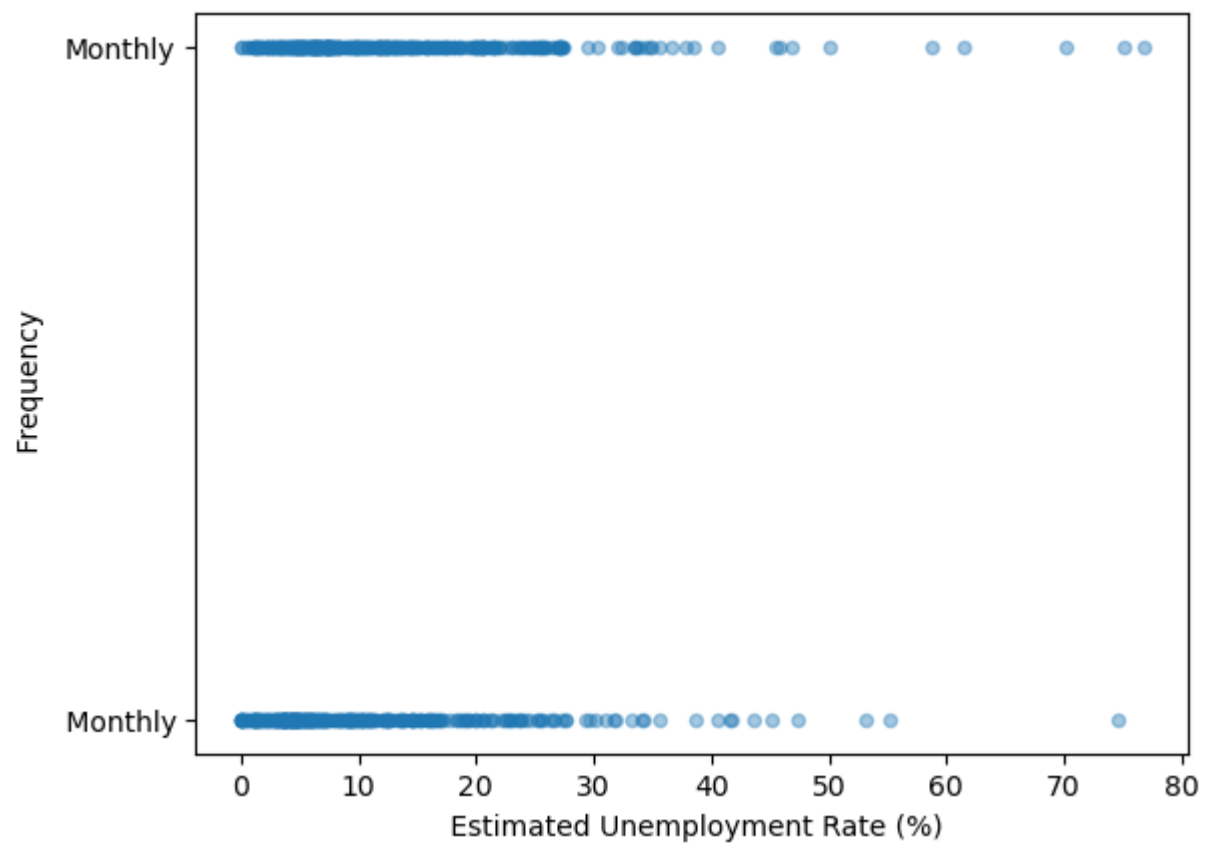
```
Out[4]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [5]: df.plot.scatter(x=" Estimated Unemployment Rate (%)", y=" Estimated Employed", alpha=0.5)
plt.show()
```



```
In [50]: df.plot.scatter(x=" Estimated Unemployment Rate (%)", y=" Frequency", alpha=0.4)
plt.show()
```



```
In [7]: df['total']=df.iloc[:,-34:].sum(axis=1)
df.head()
```

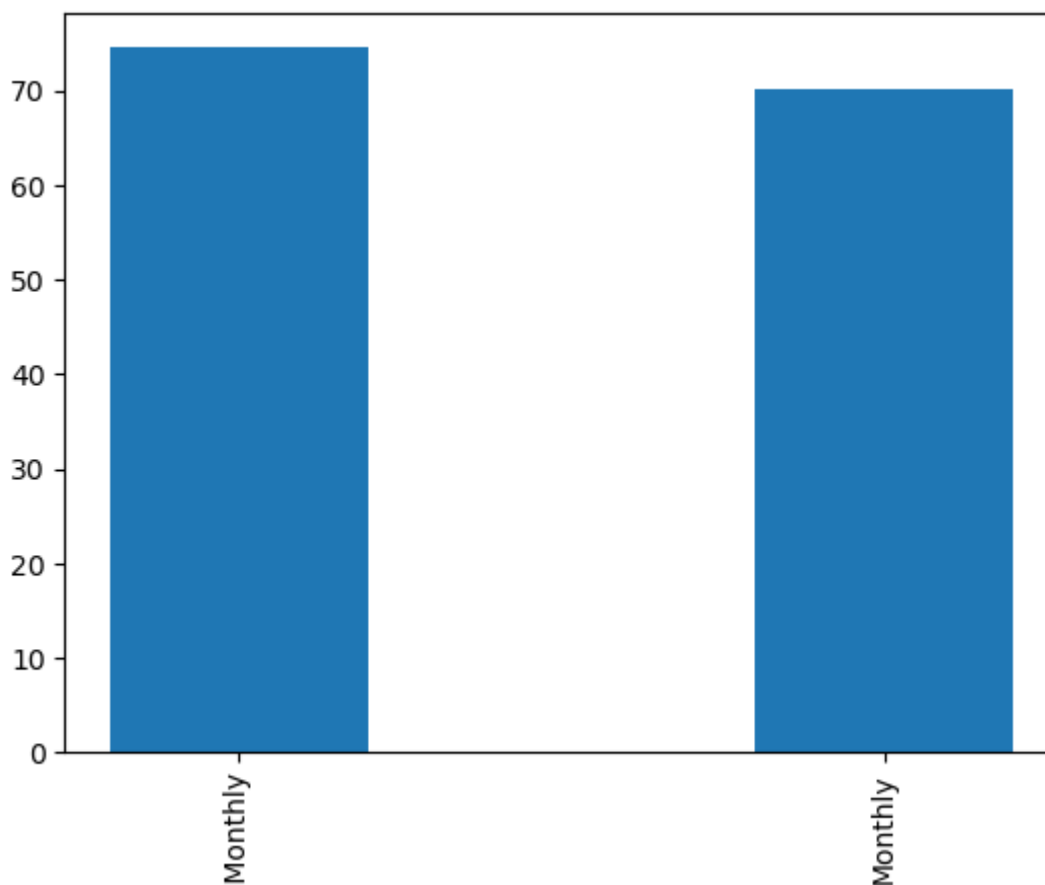
C:\Users\KIIT\AppData\Local\Temp\ipykernel\_12496\633210928.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

```
df['total']=df.iloc[:,-34:].sum(axis=1)
```

Out[7]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	total
0	Andhra Pradesh	31-05-2019	Monthly	3.65	11999139	43.24	Rural	11999185.89
1	Andhra Pradesh	30-06-2019	Monthly	3.05	11755881	42.05	Rural	11755926.10
2	Andhra Pradesh	31-07-2019	Monthly	3.75	12086707	43.50	Rural	12086754.25
3	Andhra Pradesh	31-08-2019	Monthly	3.32	12285693	43.97	Rural	12285740.29
4	Andhra Pradesh	30-09-2019	Monthly	5.17	12256762	44.68	Rural	12256811.85

```
In [8]: plt.xticks(rotation='vertical')
plt.bar(x=" Frequency", height=" Estimated Unemployment Rate (%)", visible=True, width=0.5)
plt.show()
```



```
In [9]: df.dtypes
```

```
Out[9]: Region          object
        Date           object
        Frequency      object
        Estimated Unemployment Rate (%)  float64
        Estimated Employed          int64
        Estimated Labour Participation Rate (%)  float64
        Area          object
        total         float64
        dtype: object
```

```
In [10]: data=df[:200]
```

```
In [11]: print(data)
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	\
0	Andhra Pradesh	31-05-2019	Monthly	3.65	
1	Andhra Pradesh	30-06-2019	Monthly	3.05	
2	Andhra Pradesh	31-07-2019	Monthly	3.75	
3	Andhra Pradesh	31-08-2019	Monthly	3.32	
4	Andhra Pradesh	30-09-2019	Monthly	5.17	
..	...	...	...	...	
535	Kerala	31-07-2019	Monthly	7.58	
536	Kerala	31-08-2019	Monthly	7.69	
537	Kerala	30-09-2019	Monthly	5.52	
538	Kerala	31-10-2019	Monthly	5.35	
539	Kerala	30-11-2019	Monthly	6.71	

	Estimated Employed	Estimated Labour Participation Rate (%)	Area	\
0	11999139	43.24	Rural	
1	11755881	42.05	Rural	
2	12086707	43.50	Rural	
3	12285693	43.97	Rural	
4	12256762	44.68	Rural	
..	...	...	...	
535	4105211	33.13	Urban	
536	4448650	35.91	Urban	
537	4640642	36.57	Urban	
538	4644510	36.49	Urban	
539	4062767	32.36	Urban	

	total
0	11999185.89
1	11755926.10
2	12086754.25
3	12285740.29
4	12256811.85
..	...
535	4105251.71
536	4448693.60
537	4640684.09
538	4644551.84
539	4062806.07

[540 rows x 8 columns]

In [12]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 540 entries, 0 to 539
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Region                                540 non-null    object
1   Date                                  540 non-null    object
2   Frequency                             540 non-null    object
3   Estimated Unemployment Rate (%)       540 non-null    float64
4   Estimated Employed                    540 non-null    int64
5   Estimated Labour Participation Rate (%) 540 non-null    float64
6   Area                                  540 non-null    object
7   total                                 540 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 33.9+ KB
```

```
In [13]: import matplotlib.pyplot as plt

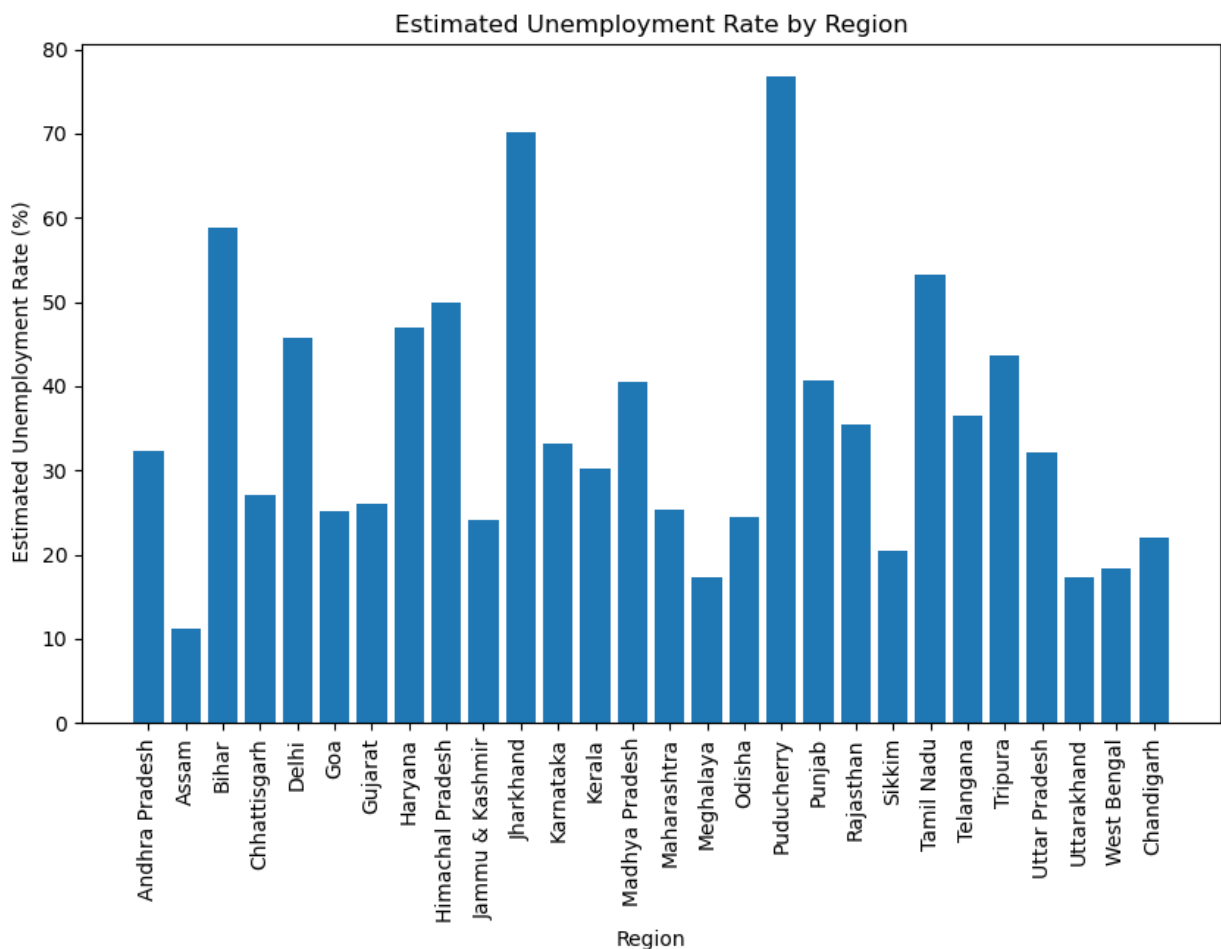
# Assuming you have your data stored in a DataFrame called 'df'

# Create a bar plot
plt.figure(figsize=(10, 6)) # Adjust the figure size if needed
plt.bar(x=df['Region'], height=df[' Estimated Unemployment Rate (%)'])

# Set the title and labels for the plot
plt.title('Estimated Unemployment Rate by Region')
plt.xlabel('Region')
plt.ylabel(' Estimated Unemployment Rate (%)')

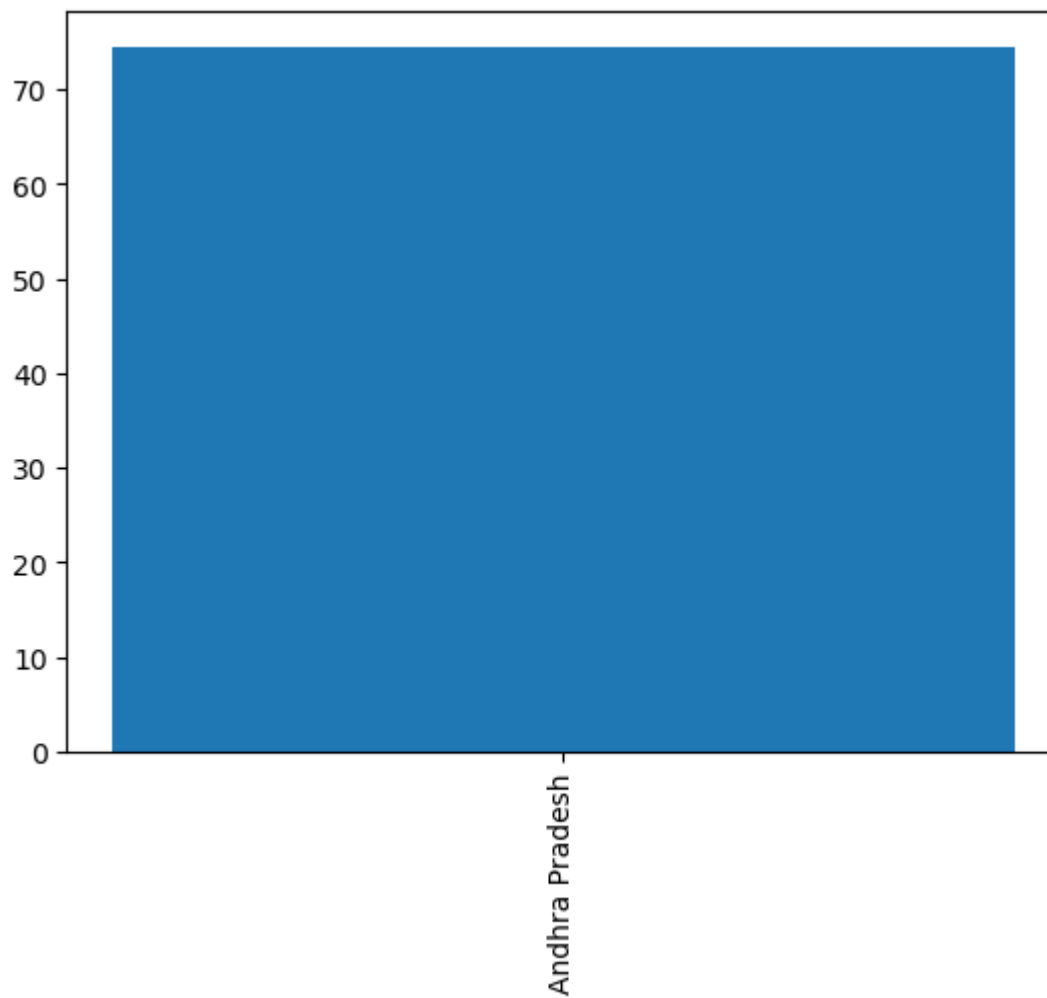
# Rotate x-axis labels vertically
plt.xticks(rotation='vertical')

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



```
In [14]: plt.xticks(rotation='vertical')
plt.bar(x="Andhra Pradesh",height=" Estimated Unemployment Rate (%)",width=0.1, bottom=0)
```

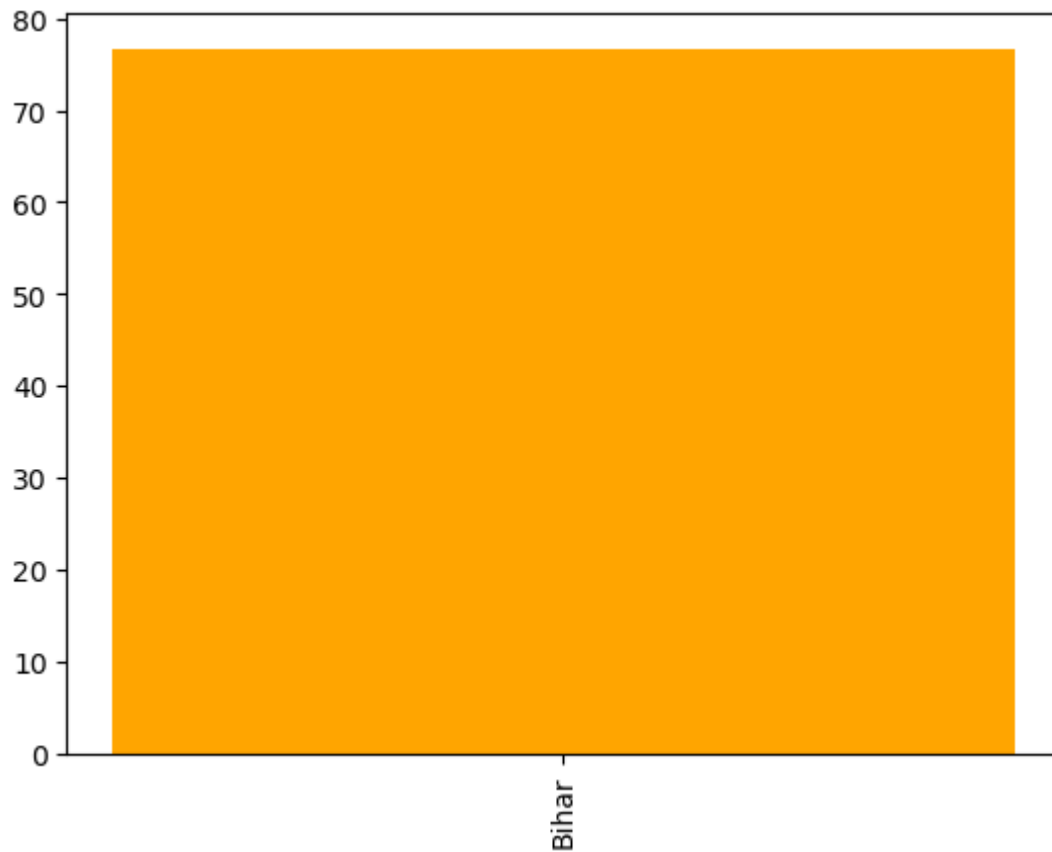
```
Out[14]: <BarContainer object of 540 artists>
```



```
In [59]: plt.xticks(rotation='vertical')
plt.bar(x="Bihar",height=" Estimated Unemployment Rate (%)",width=0.2, align='center')
```

```
Out[59]: <BarContainer object of 720 artists>
```





```
In [15]: import matplotlib.pyplot as plt

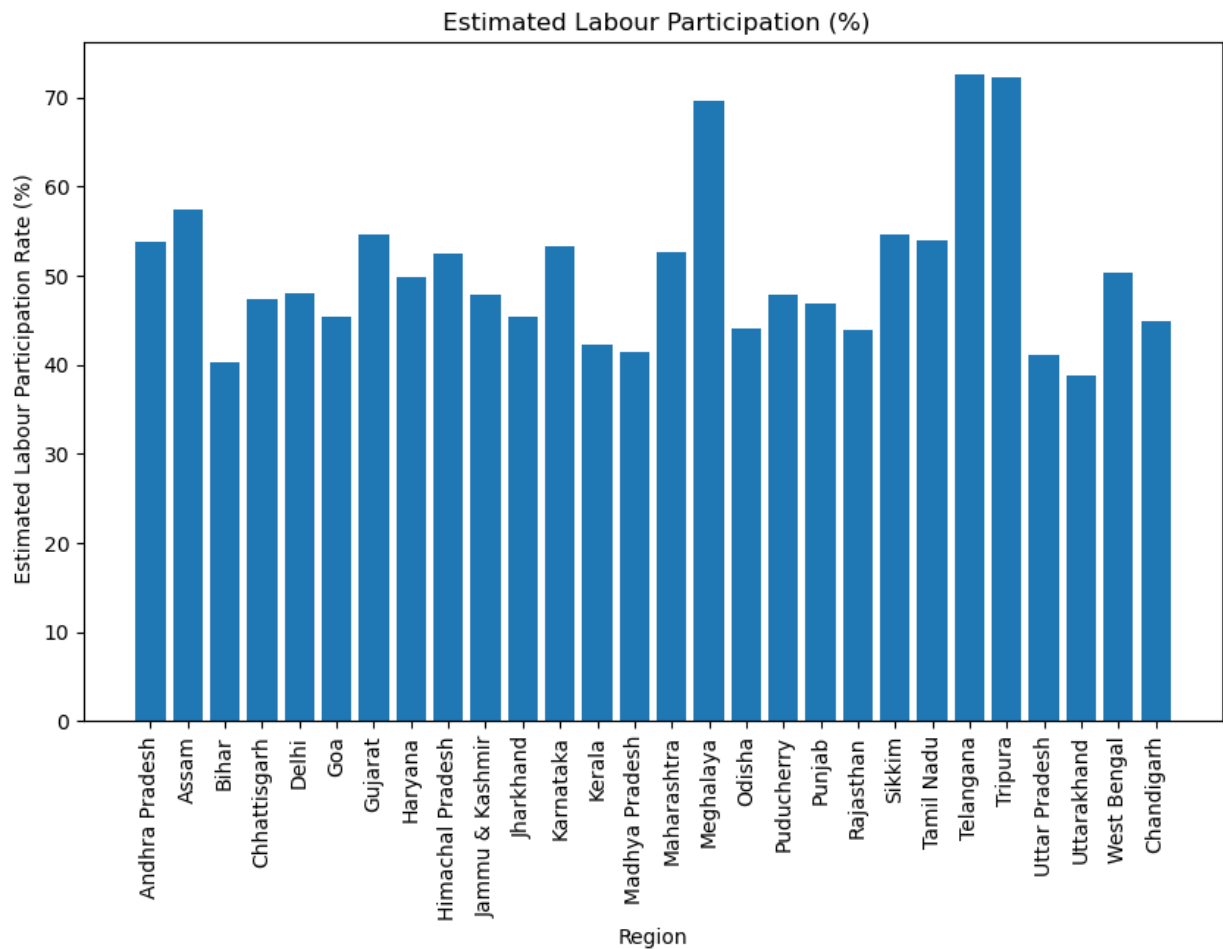
# Assuming you have your data stored in a DataFrame called 'df'

# Create a bar plot
plt.figure(figsize=(10, 6)) # Adjust the figure size if needed
plt.bar(x=df['Region'], height=df[' Estimated Labour Participation Rate (%)'])

# Set the title and labels for the plot
plt.title(' Estimated Labour Participation (%)')
plt.xlabel('Region')
plt.ylabel(' Estimated Labour Participation Rate (%)')

# Rotate x-axis labels vertically
plt.xticks(rotation='vertical')

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



```
In [16]: x=df.loc[df["Region"]=="West Bengal"]
print(x[" Estimated Unemployment Rate (%)"].sum())
```

227.49

```
In [17]: x=df.loc[df["Region"]=="Andhra Pradesh"]
print(x[" Estimated Unemployment Rate (%)"].sum())
```

209.35999999999999

```
In [18]: x=df.loc[df["Region"]=="Assam"]
print(x[" Estimated Unemployment Rate (%)"].sum())
```

167.13

```
In [19]: x=df.loc[df["Region"]=="Bihar"]
print(x[" Estimated Unemployment Rate (%)"].sum())
```

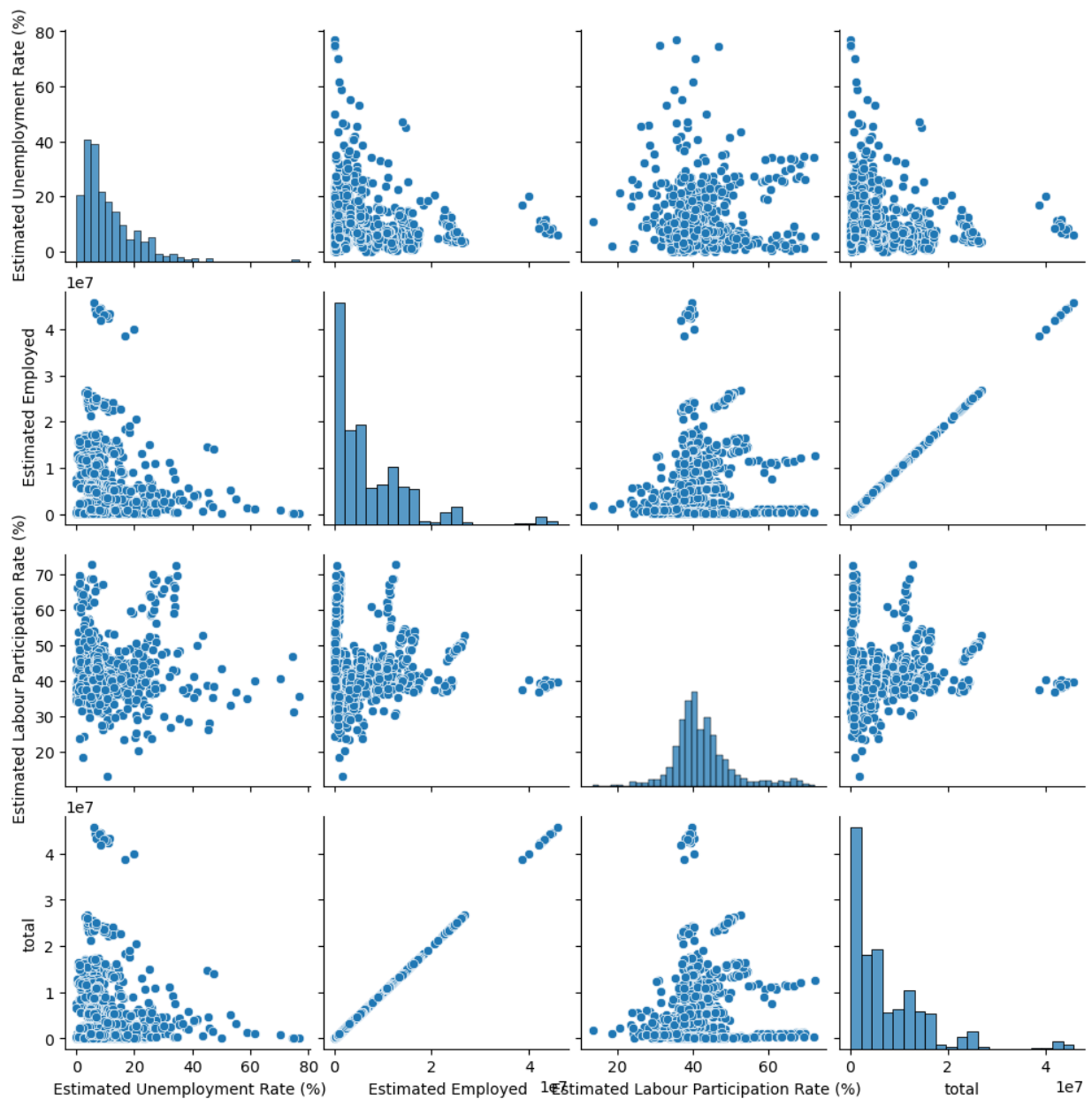
529.71

```
In [20]: x=df.loc[df["Region"]=="Goa"]
print(x[" Estimated Unemployment Rate (%)"].sum())
```

222.57999999999998

```
In [21]: sns.pairplot(df)
```

```
Out[21]: <seaborn.axisgrid.PairGrid at 0x2a977a7a130>
```



```
In [22]: x = input("Enter a region : ")
         rslt_df = df[df['Region'] == x]
         rslt_df
```

Enter a region : Bihar

Out[22]:

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	total
<b>26</b>	Bihar	31-05-2019	Monthly	9.27	24322330	39.75	Rural	24322379.02
<b>27</b>	Bihar	30-06-2019	Monthly	10.20	24097712	39.71	Rural	24097761.91
<b>28</b>	Bihar	31-07-2019	Monthly	13.44	23248875	39.66	Rural	23248928.10
<b>29</b>	Bihar	31-08-2019	Monthly	11.00	22260203	36.85	Rural	22260250.85
<b>30</b>	Bihar	30-09-2019	Monthly	8.87	23905700	38.57	Rural	23905747.44
<b>31</b>	Bihar	31-10-2019	Monthly	12.47	24053140	40.31	Rural	24053192.78
<b>32</b>	Bihar	30-11-2019	Monthly	12.40	22445989	37.51	Rural	22446038.91
<b>33</b>	Bihar	31-12-2019	Monthly	10.16	22914530	37.25	Rural	22914577.41
<b>34</b>	Bihar	31-01-2020	Monthly	9.13	23409006	37.54	Rural	23409052.67
<b>35</b>	Bihar	29-02-2020	Monthly	9.61	23168192	37.28	Rural	23168238.89
<b>36</b>	Bihar	31-03-2020	Monthly	15.39	22667882	38.88	Rural	22667936.27
<b>37</b>	Bihar	30-04-2020	Monthly	45.09	14645275	38.63	Rural	14645358.72
<b>38</b>	Bihar	31-05-2020	Monthly	47.26	14050319	38.50	Rural	14050404.76
<b>39</b>	Bihar	30-06-2020	Monthly	20.49	20622566	37.40	Rural	20622623.89
<b>387</b>	Bihar	31-05-2019	Monthly	19.90	3029344	39.80	Urban	3029403.70

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	total
<b>388</b>	Bihar	30-06-2019	Monthly	13.29	3248864	39.35	Urban	3248916.64
<b>389</b>	Bihar	31-07-2019	Monthly	16.41	3059744	38.36	Urban	3059798.77
<b>390</b>	Bihar	31-08-2019	Monthly	17.66	2994763	38.03	Urban	2994818.69
<b>391</b>	Bihar	30-09-2019	Monthly	20.46	2992082	39.25	Urban	2992141.71
<b>392</b>	Bihar	31-10-2019	Monthly	14.06	3173429	38.45	Urban	3173481.51
<b>393</b>	Bihar	30-11-2019	Monthly	17.62	3081077	38.86	Urban	3081133.48
<b>394</b>	Bihar	31-12-2019	Monthly	14.91	2977857	36.29	Urban	2977908.20
<b>395</b>	Bihar	31-01-2020	Monthly	20.69	2988665	38.99	Urban	2988724.68
<b>396</b>	Bihar	29-02-2020	Monthly	15.11	3113464	37.87	Urban	3113516.98
<b>397</b>	Bihar	31-03-2020	Monthly	15.73	3049637	37.29	Urban	3049690.02
<b>398</b>	Bihar	30-04-2020	Monthly	58.77	1400962	34.94	Urban	1401055.71
<b>399</b>	Bihar	31-05-2020	Monthly	37.87	2207026	36.45	Urban	2207100.32
<b>400</b>	Bihar	30-06-2020	Monthly	12.45	3124663	36.54	Urban	3124711.99

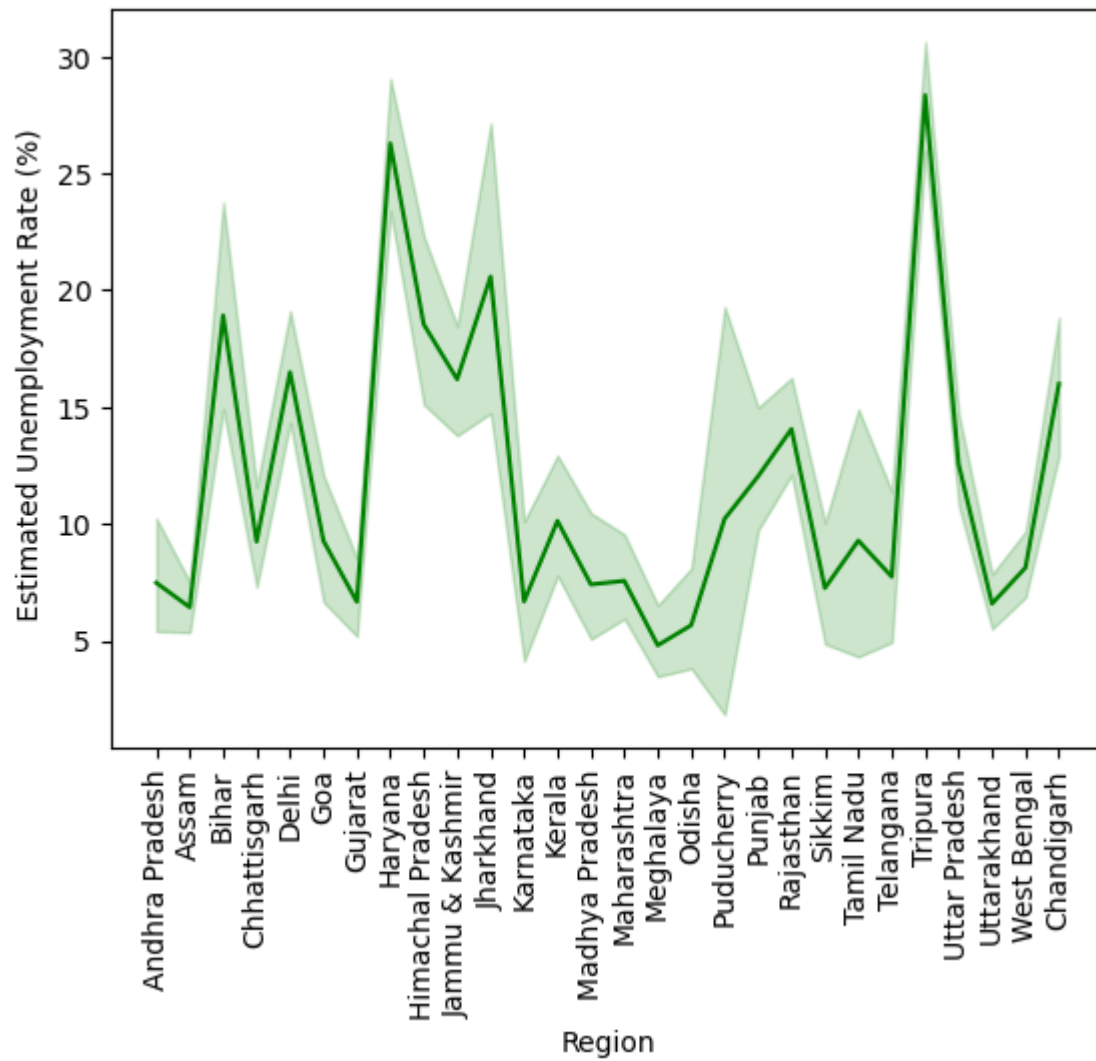
```
In [23]: sns.heatmap(df.corr(),annot=True)
```

```
Out[23]: <AxesSubplot:>
```



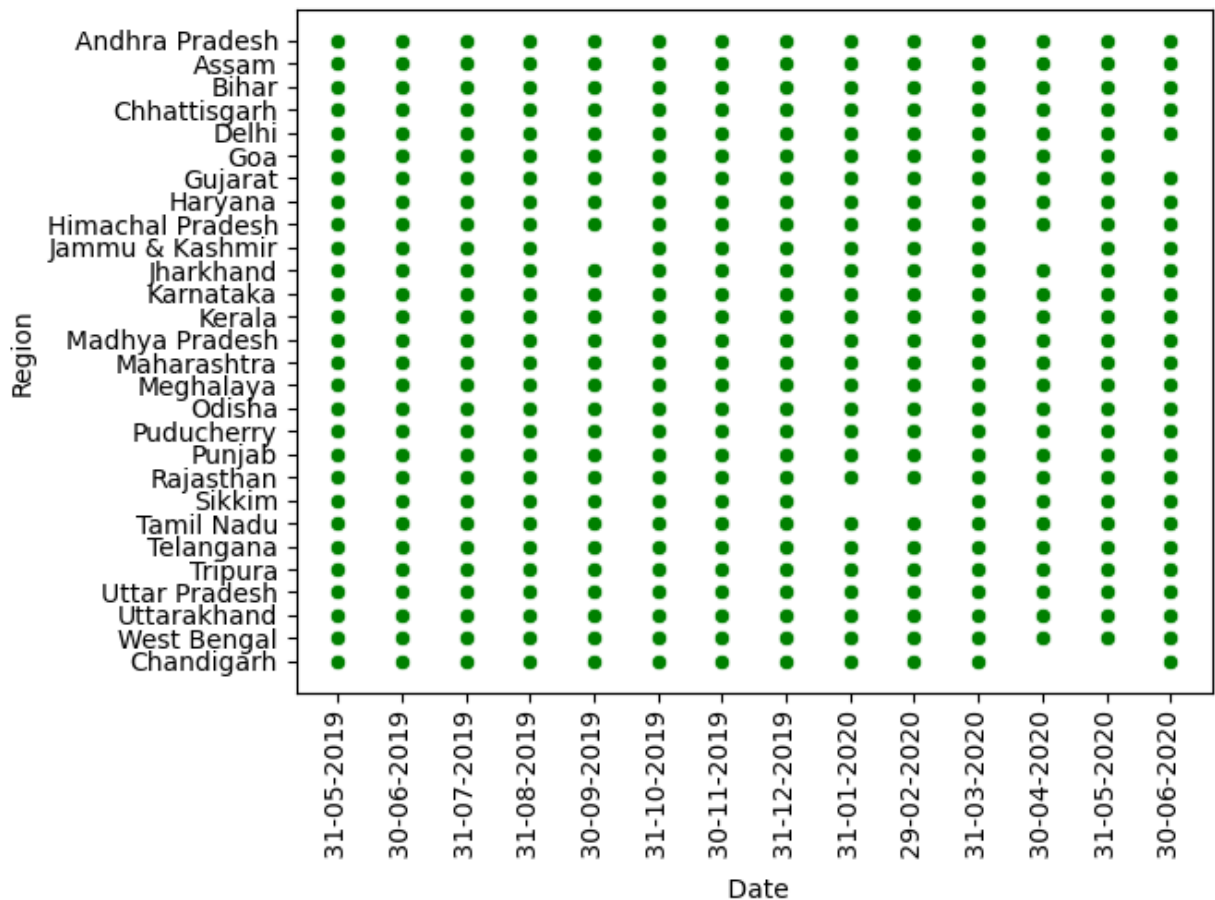
```
In [24]: plt.xticks(rotation='vertical')
sns.lineplot(x=df['Region'], y=df[' Estimated Unemployment Rate (%)'], color = 'green')

Out[24]: <AxesSubplot:xlabel='Region', ylabel=' Estimated Unemployment Rate (%)'>
```

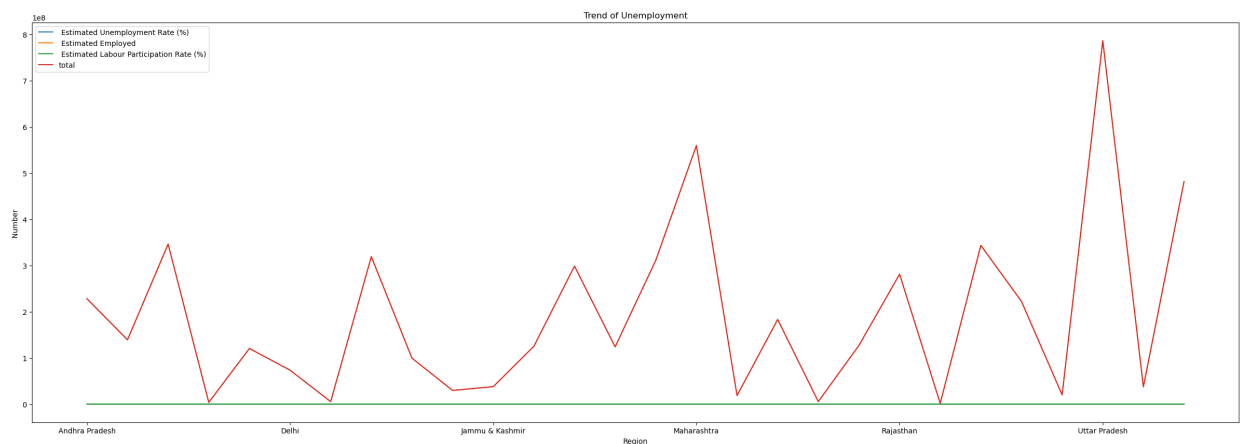


```
In [25]: plt.xticks(rotation='vertical')
sns.scatterplot(x=df['Date'], y=df['Region'], color = 'green', data=df)
```

```
Out[25]: <AxesSubplot:xlabel='Date', ylabel='Region'>
```



```
In [65]: grouped = df.groupby('Region').sum()
grouped.plot(kind='line', figsize=(30, 10))
plt.title("Trend of Unemployment")
plt.xlabel("Region")
plt.ylabel("Number ")
plt.show()
```



```
In [28]: pd.pivot_table(data = df, values = ' Estimated Unemployment Rate (%)', index =['Region'
```



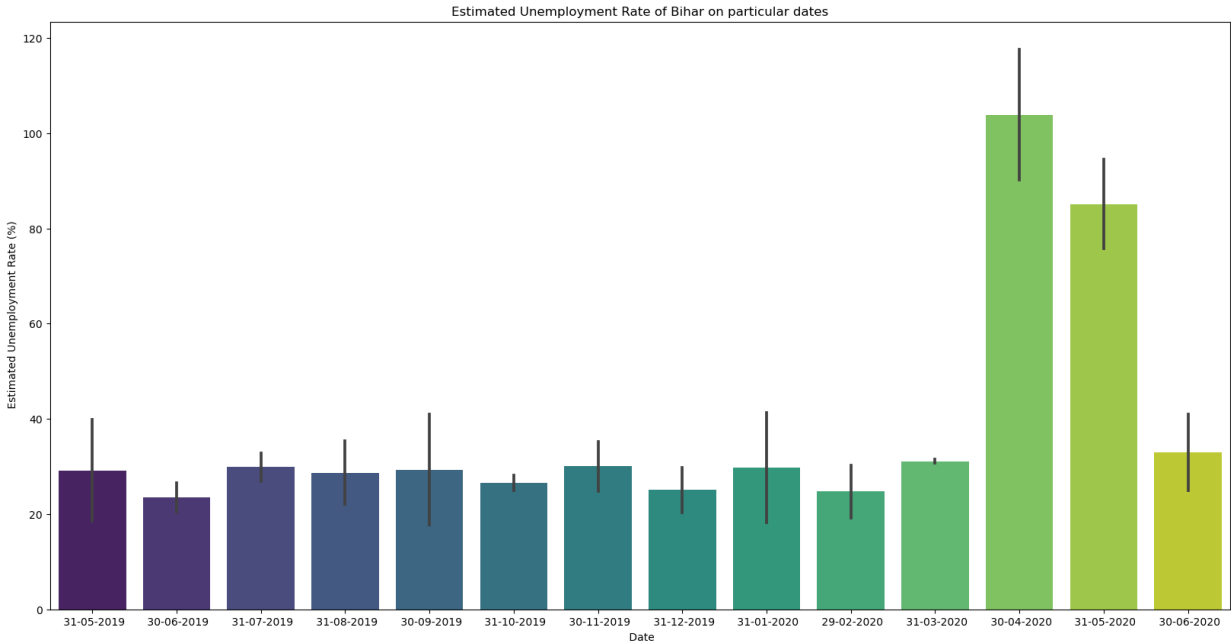
Out[28]:

Estimated Unemployment Rate (%)				
Region	Date	Area		
Andhra Pradesh	29-02-2020	Rural	5.91	
		Urban	5.66	
	30-04-2020	Rural	16.29	
		Urban	32.30	
	30-06-2019	Rural	3.05	
...	...	...	...	
West Bengal	31-08-2019	Urban	7.27	
	31-10-2019	Rural	6.78	
		Urban	7.83	
	31-12-2019	Rural	5.79	
		Urban	7.24	

740 rows × 1 columns

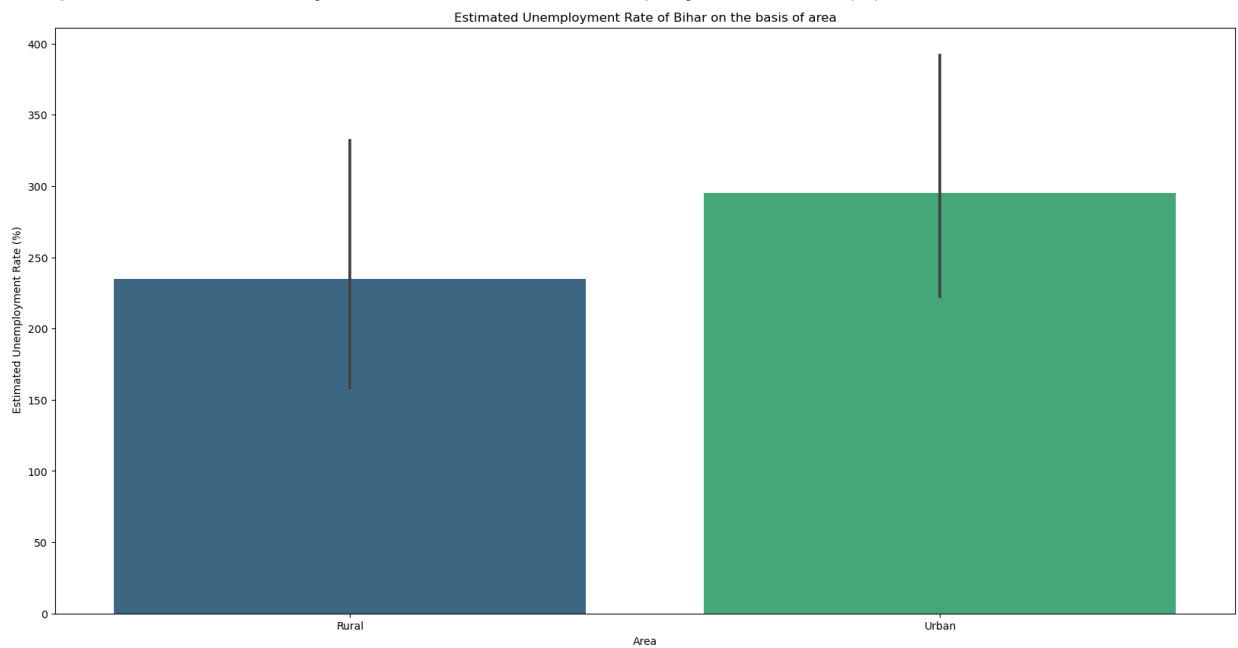
```
In [40]: pq2 = df.loc[df['Region']=='Bihar']
plt.figure(figsize=(20,10))
plt.title("Estimated Unemployment Rate of Bihar on particular dates")
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x=" Date",estimator=sum,pale
```

```
Out[40]: <AxesSubplot:title={'center': 'Estimated Unemployment Rate of Bihar on particular date
s'}, xlabel=' Date', ylabel=' Estimated Unemployment Rate (%)'>
```



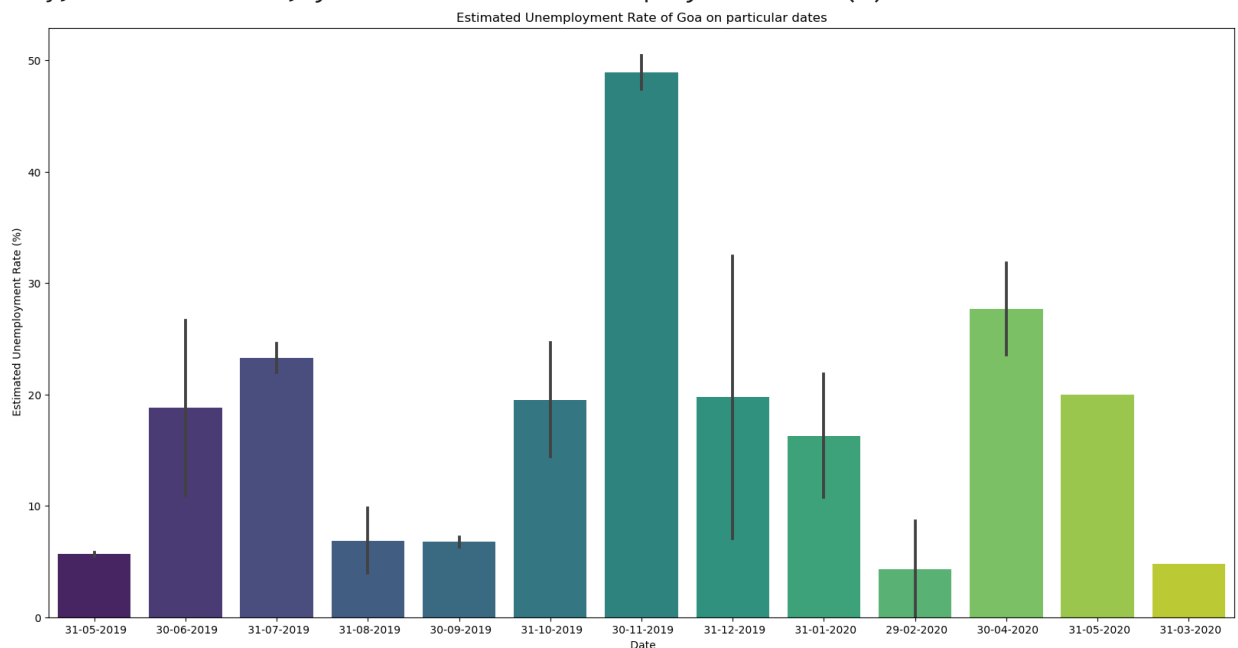
```
In [45]: pq2 = df.loc[df['Region']=='Bihar']
plt.figure(figsize=(20,10))
plt.title("Estimated Unemployment Rate of Bihar on the basis of area")
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x="Area",estimator=sum,pale
```

Out[45]: <AxesSubplot:title={'center': 'Estimated Unemployment Rate of Bihar on the basis of area'}, xlabel='Area', ylabel=' Estimated Unemployment Rate (%)'>



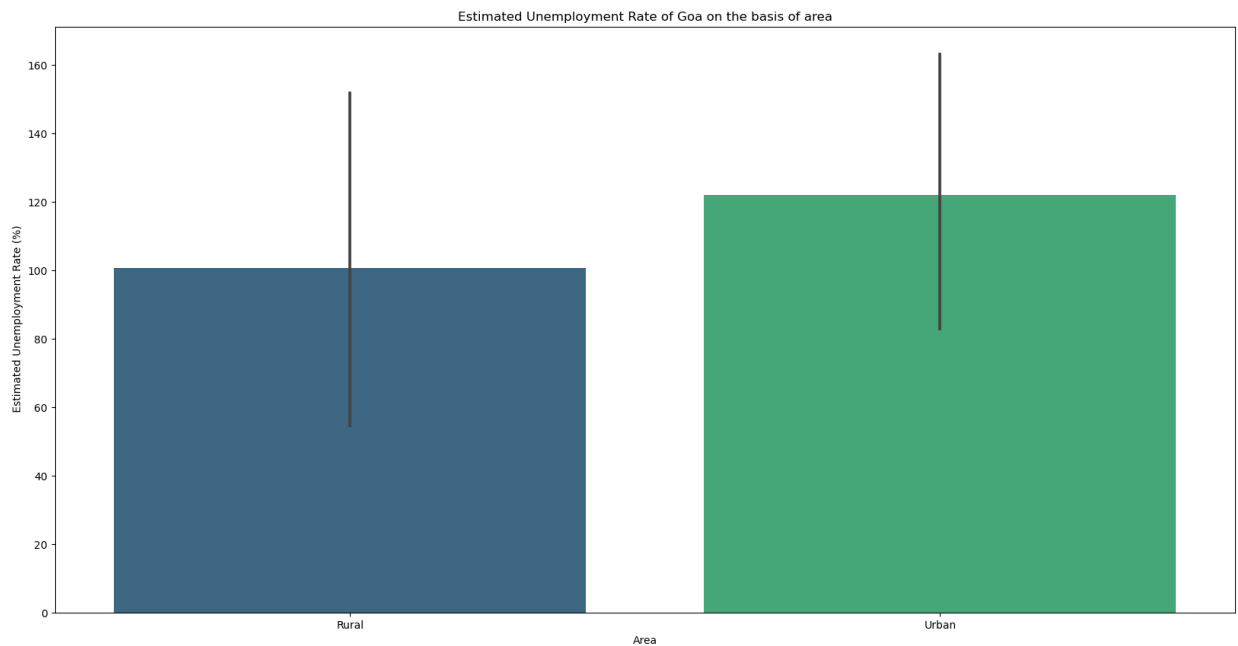
In [46]: `pq2 = df.loc[df['Region']=='Goa']  
plt.figure(figsize=(20,10))  
plt.title("Estimated Unemployment Rate of Goa on particular dates")  
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x=" Date",estimator=sum,pale`

Out[46]: <AxesSubplot:title={'center': 'Estimated Unemployment Rate of Goa on particular date s'}, xlabel=' Date', ylabel=' Estimated Unemployment Rate (%)'>



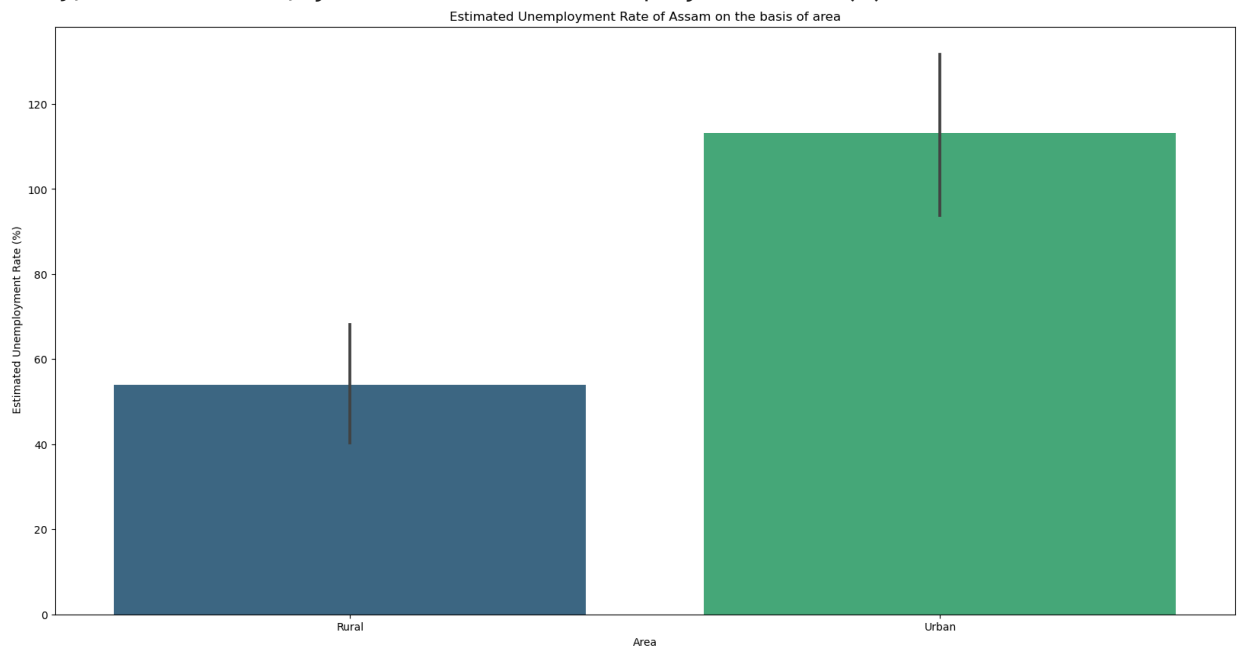
In [42]: `pq2 = df.loc[df['Region']=='Goa']  
plt.figure(figsize=(20,10))  
plt.title("Estimated Unemployment Rate of Goa on the basis of area")  
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x="Area",estimator=sum,pale`

Out[42]: <AxesSubplot:title={'center': 'Estimated Unemployment Rate of Goa on the basis of area'}, xlabel='Area', ylabel=' Estimated Unemployment Rate (%)'>



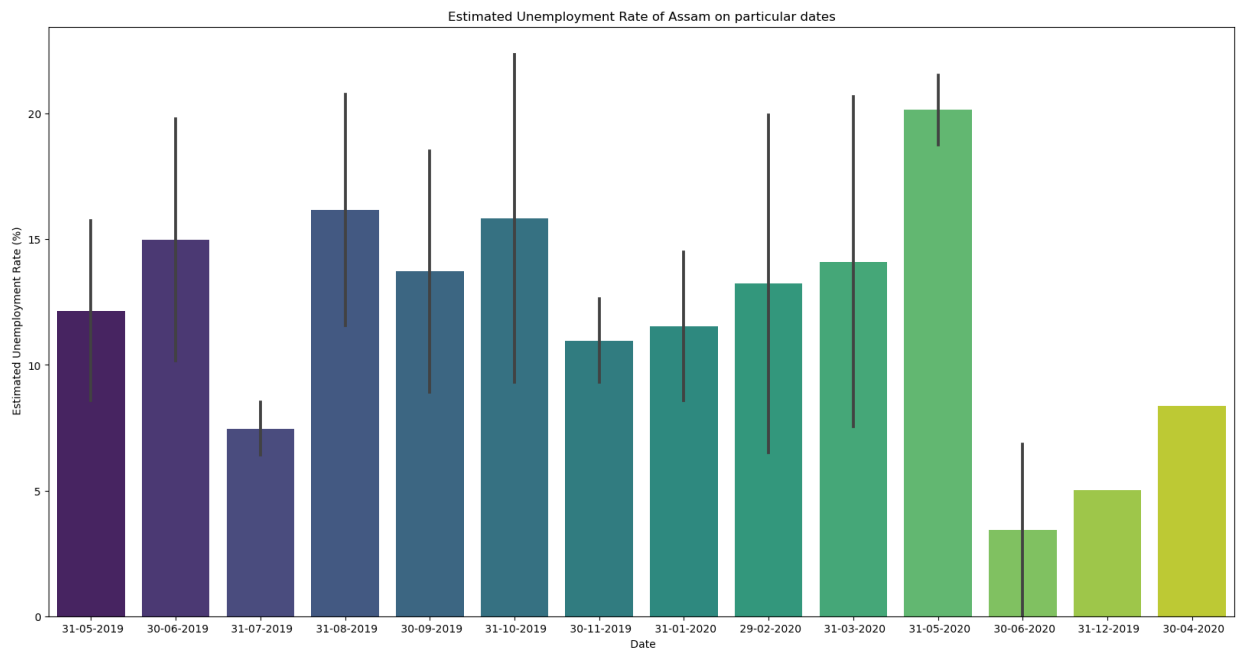
```
In [44]: pq2 = df.loc[df['Region']=='Assam']
plt.figure(figsize=(20,10))
plt.title("Estimated Unemployment Rate of Assam on the basis of area")
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x="Area",estimator=sum,palette=
```

```
Out[44]: <AxesSubplot:title={'center':'Estimated Unemployment Rate of Assam on the basis of area'}, xlabel='Area', ylabel=' Estimated Unemployment Rate (%)'>
```



```
In [38]: pq2 = df.loc[df['Region']=='Assam']
plt.figure(figsize=(20,10))
plt.title("Estimated Unemployment Rate of Assam on particular dates")
sns.barplot(data=pq2,y=" Estimated Unemployment Rate (%)",x=" Date",estimator=sum,palette=
```

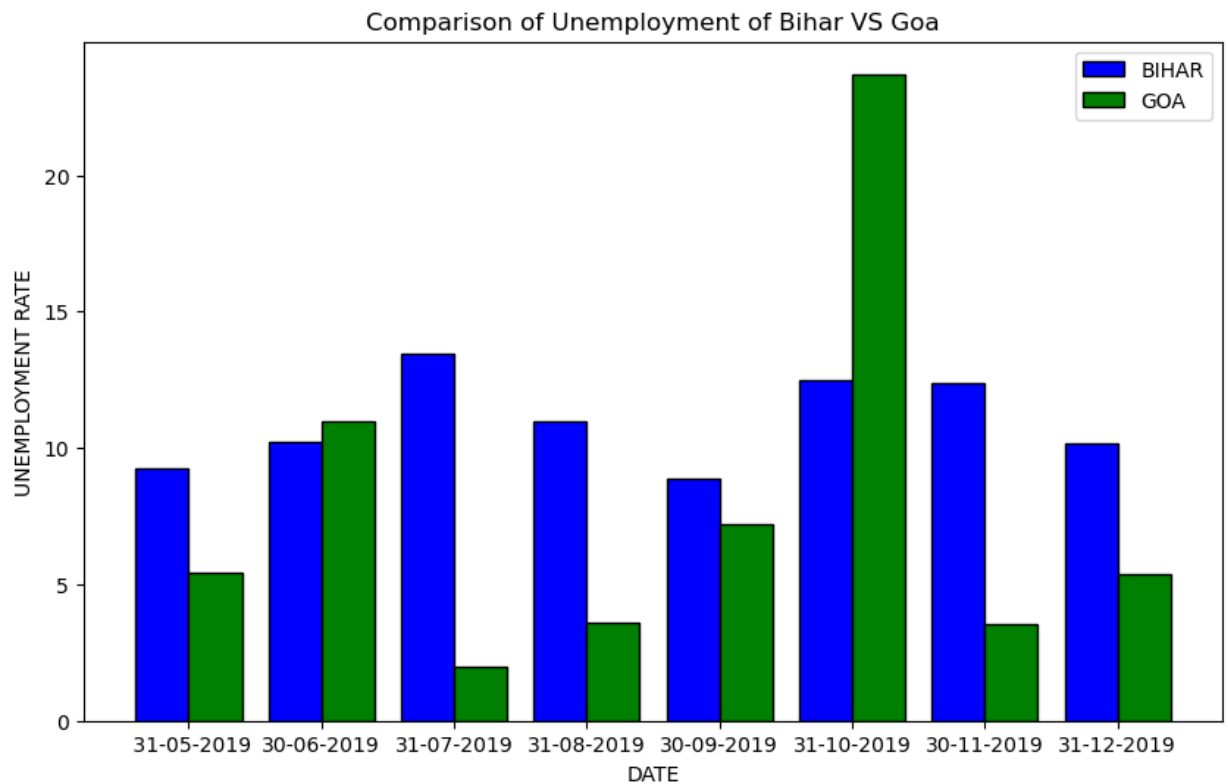
```
Out[38]: <AxesSubplot:title={'center':'Estimated Unemployment Rate of Assam on particular dates'}, xlabel=' Date', ylabel=' Estimated Unemployment Rate (%)'>
```



```
In [82]: Bihar = [9.27,10.2,13.44,11,8.87,12.47,12.4,10.16]
Goa = [5.45,10.98,1.98,3.61,7.21,23.71,3.54,5.38]
n=8
r = np.arange(n)
width =0.4
plt.figure(figsize=(10, 6))
plt.bar(r,Bihar, color = 'b',width = width, edgecolor = 'black',label='BIHAR')
plt.bar(r + width, Goa, color = 'g',width = width, edgecolor = 'black',label='GOA')

plt.xlabel("DATE")
plt.ylabel("UNEMPLOYMENT RATE")
plt.title("Comparison of Unemployment of Bihar VS Goa")

# plt.grid(linestyle='--')
plt.xticks(r + width/2,[' 31-05-2019',
'30-06-2019',
'31-07-2019',
'31-08-2019',
'30-09-2019',
'31-10-2019',
'30-11-2019',
'31-12-2019',])
plt.legend()
plt.show()
```



In [108...

```

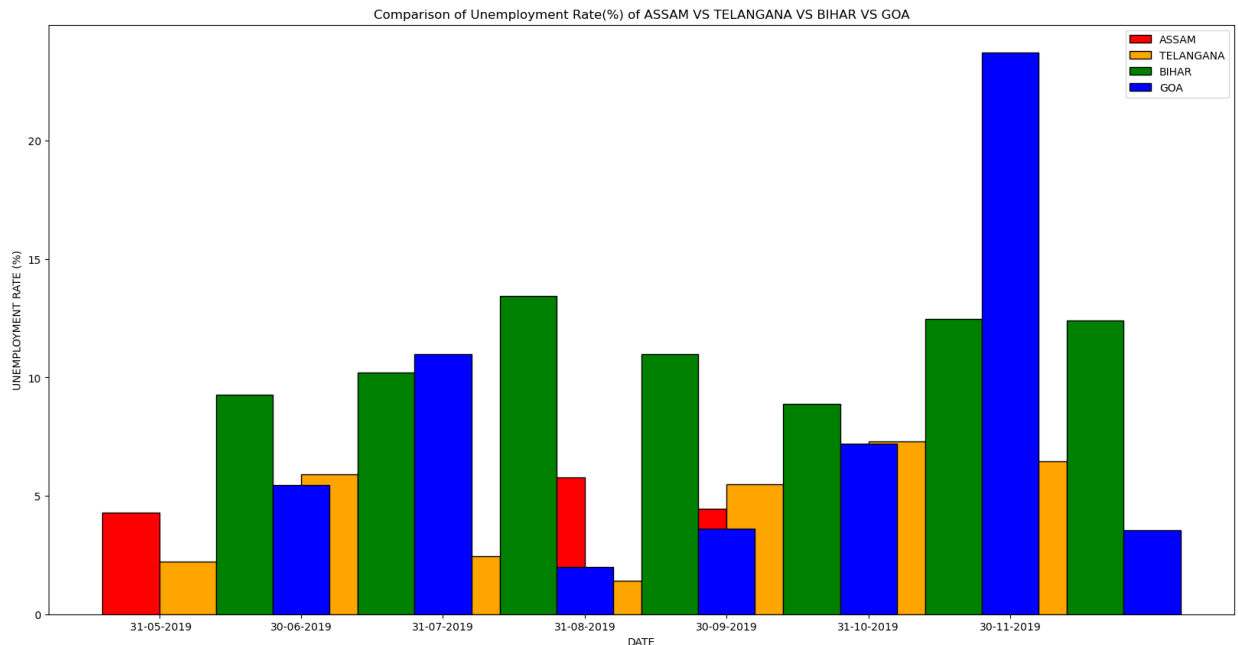
Assam = [4.29,
5.08,
4.26,
5.79,
4.46,
4.65,
4.66]
Telangana = [2.23,
5.92,
2.45,
1.4,
5.49,
7.29,
6.47]
Bihar = [9.27,10.2,13.44,11,8.87,12.47,12.4]
Goa = [5.45,10.98,1.98,3.61,7.21,23.71,3.54]
n=7
r = np.arange(n)
width =0.4
plt.figure(figsize=(20, 10))
plt.bar(r,Assam, color = 'r',width = width, edgecolor = 'black',label='ASSAM')
plt.bar(r + width,Telangana, color = 'orange',width = width, edgecolor = 'black',label='TELANGANA')
plt.bar(r + width+width, Bihar, color = 'g',width = width, edgecolor = 'black',label='BIHAR')
plt.bar(r + width+width+width, Goa, color = 'b',width = width, edgecolor = 'black',label='GOA')

plt.xlabel("DATE")
plt.ylabel("UNEMPLOYMENT RATE (%)")
plt.title("Comparison of Unemployment Rate(%) of ASSAM VS TELANGANA VS BIHAR VS GOA")

# plt.grid(linestyle='--')
plt.xticks(r + width/2,[' 31-05-2019',
'30-06-2019',
'31-07-2019',
'31-08-2019',

```

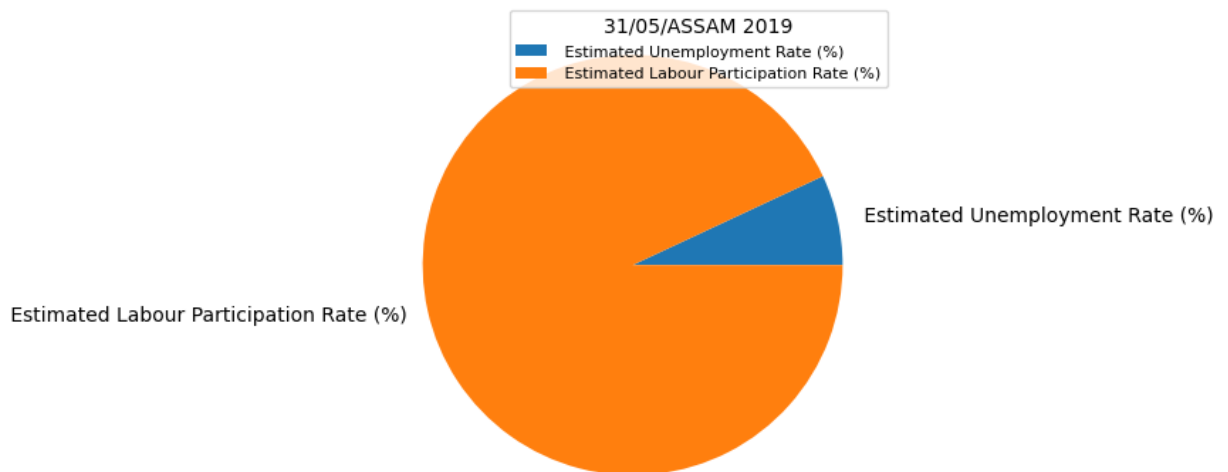
```
'30-09-2019',
'31-10-2019',
'30-11-2019'])
plt.legend()
plt.show()
```



In [117...

```
y = np.array([4.29,57.39])
mylabels = [" Estimated Unemployment Rate (%)"," Estimated Labour Participation Rate (%)"]

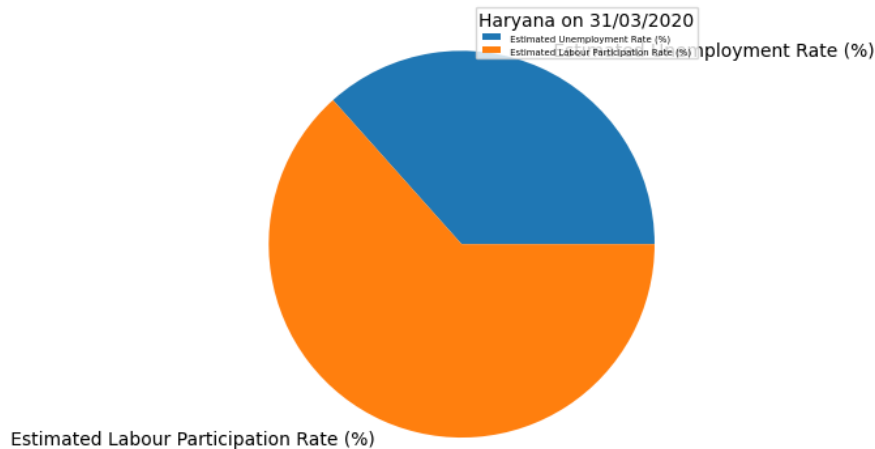
plt.pie(y, labels= mylabels)
plt.legend(title="31/05/ASSAM 2019", fontsize=8)
plt.show()
```



In [120...

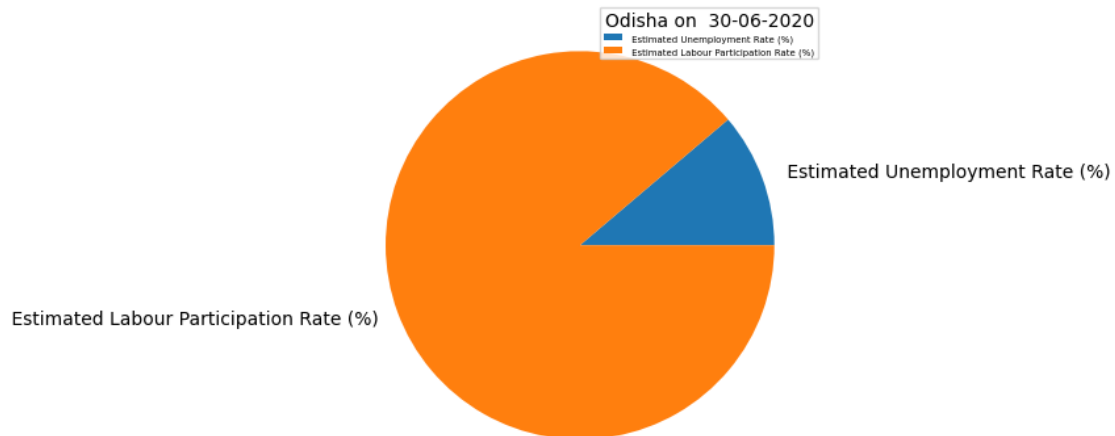
```
y = np.array([23.92,41.4])
mylabels = [" Estimated Unemployment Rate (%)"," Estimated Labour Participation Rate (%)"]
plt.title("Pie Chart Analysis of Estimated Unemployment and Estimated Labour Particip")
plt.pie(y, labels= mylabels)
plt.legend(title="Haryana on 31/03/2020", fontsize=5)
plt.show()
```

Pie Chart Analysis of Estimated Unemployment and Estimated Labour Participation of Haryana on 31/05/2019



```
In [121... y = np.array([4.59,36.36])
mylabels = [" Estimated Unemployment Rate (%)"," Estimated Labour Participation Rate (
plt.title("Pie Chart Analysis of Estimated Unemployment and Estimated Labour Particip
plt.pie(y, labels= mylabels)
plt.legend(title="Odisha on 30-06-2020", fontsize=5)
plt.show()
```

Pie Chart Analysis of Estimated Unemployment and Estimated Labour Participation of Odisha on 30-06-2020



```
In [122... y = np.array([26.53,63.02])
mylabels = [" Estimated Unemployment Rate (%)"," Estimated Labour Participation Rate (
plt.title("Pie Chart Analysis of Estimated Unemployment and Estimated Labour Particip
plt.pie(y, labels= mylabels)
plt.legend(title="Tripura on 31-12-2019", fontsize=5)
plt.show()
```

Pie Chart Analysis of Estimated Unemployment and Estimated Labour Participation of Tripura on 31-12-2019

