Data science Task 2

Name : Danthamala Balaji Preetham

UNEMPLOYEMENT ANALYSIS WITH PYTHON

Unemployment is measured by the unemployment rate which is the number of people who are unemployed as a percentage of the total labour force. We have seen a sharp increase in the unemployment rate during Covid-19, so analyzing the unemployment rate can be a good data science project.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.figure_factory as ff
import plotly.graph_objects as gg
import plotly.express as exp
%matplotlib inline
```

```
df = pd.read_excel("/content/Unemployment in India.xlsx")
df_11_2020 = pd.read_excel("/content/Unemployment_Rate_upto_11_2020.xlsx")
```

```
df.head()
```

	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

```
df_11_2020.head()
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	Longitude	Latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.74
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.74
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.74
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.74

Next steps: Generate code with df_11_2020 View recommended plots

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 754 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    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: 41.4+ KB
```

```
df_11_2020.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
```

```
df.shape
```

```
(754, 7)
```

```
df_11_2020.shape
```

```
(267, 9)
```

```
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

```
df_11_2020.describe()
```

	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

```
df.isnull().sum()
```

```
Region                                   14
Date                                     14
Frequency                                14
Estimated Unemployment Rate (%)          14
Estimated Employed                       14
Estimated Labour Participation Rate (%)  14
Area                                     14
dtype: int64
```

```
df_11_2020.isnull().sum()
```

```
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
```

```
df.isna().sum()
```

```
Region                                   14
Date                                     14
Frequency                                14
Estimated Unemployment Rate (%)          14
Estimated Employed                       14
Estimated Labour Participation Rate (%)  14
Area                                     14
dtype: int64
```

```
df_11_2020.isna().sum()
```

```
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
```

```
df = df.dropna()
```

```
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
```

```
df.isna().sum()
```

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

```
df.shape
```

```
(740, 7)
```

```
df.duplicated().sum()
```

```
0
```

```
df_11_2020.duplicated().sum()
```

```
0
```

```
df.columns = ['State' , 'Date' , 'Frequency' , 'Estimated Unemployment Rate',
              'Estimated Employed', 'Estimated Labour Participation Rate', 'Area']
```

```
df_11_2020.columns = ['State' , 'Date' , 'Frequency' , 'Estimated Unemployment Rate','Estimated Employed', 'Estimated Labour Participation Rate','Region','Longitude','Latitude']
```

```
df.columns
```

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

```
df_11_2020.columns
```

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

```
df.head(2)
```

	State	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

Next steps: Generate code with df View recommended plots

```
df_11_2020.head(2)
```

	State	Date	Frequency	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Region	Longitude	Latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74

Next steps: Generate code with df_11_2020 View recommended plots

```
#State with highest unemployment rate

df['State'].value_counts().idxmax()
```

```
'Andhra Pradesh'
```

```
df['State'].value_counts()
```

```
State
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: count, dtype: int64
```

```
df_11_2020['State'].value_counts().idxmax()
```

```
'Andhra Pradesh'
```

```
# State with lowest unemployment rate

df['State'].value_counts().idxmin()
```

```
'Chandigarh'
```

```
df_11_2020['State'].value_counts().idxmin()
```

```
'Sikkim'
```

```
# Month of Employment
# This code converts the 'Date' coloumn to a datetime type,extracts months as integer, and adds a new coloumn with the corresponding three-letter month abbreviation

import datetime as dt
import calendar as cal

df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)  #This line converts the 'Date' coloumn in dataframe to datetime type.
df['month_int'] = df['Date'].dt.month              #This line extracts month component from 'Date' coloumn and assigns it to a new coloumn called 'month_int' in the dataframe df.
df['month'] = df['month_int'].apply(lambda x: cal.month_abbr[x])  #This line creates a mew coloumn 'month' in the dataframe df.
```

```
df_11_2020['Date'] = pd.to_datetime(df_11_2020['Date'], dayfirst=True)
df_11_2020['month_int'] = df_11_2020['Date'].dt.month
df_11_2020['month'] = df_11_2020['month_int'].apply(lambda x: cal.month_abbr[x])
```

```
# month with the highest unemployment

df['month'].value_counts().idxmax()
```

```
'May'
```

```
df_11_2020['month'].value_counts().idxmax()
```

```
'Mar'
```

```
# Month with the lowest employment

df['month'].value_counts().idxmin()
```

```
'Apr'
```

```
df_11_2020['month'].value_counts().idxmin()
```

```
'Jan'
```

```
df.head(3)
```

	State	Date	Frequency	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Area	month_int	month
0	Andhra Pradesh	2019-05-31	Monthly	3.65	11999139.0	43.24	Rural	5	May
1	Andhra Pradesh	2019-06-30	Monthly	3.05	11755881.0	42.05	Rural	6	Jun

Next steps: Generate code with `df` View recommended plots

```
df_11_2020.head(3)
```

	State	Date	Frequency	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Region	Longitude	Latitude	month_int	month
0	Andhra Pradesh	2020-01-31	M	5.48	16635535	41.02	South	15.9129	79.74	1	Jan
1	Andhra	2020-									

Next steps: Generate code with `df_11_2020` View recommended plots

```
df.drop(columns=['Frequency','month_int'])
```

	State	Date	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Area	month
0	Andhra Pradesh	2019-05-31	3.65	11999139.0	43.24	Rural	May
1	Andhra Pradesh	2019-06-30	3.05	11755881.0	42.05	Rural	Jun
2	Andhra Pradesh	2019-07-31	3.75	12086707.0	43.50	Rural	Jul
3	Andhra Pradesh	2019-08-31	3.32	12285693.0	43.97	Rural	Aug
4	Andhra Pradesh	2019-09-30	5.17	12256762.0	44.68	Rural	Sep
...	...	...	...	...	...	...	...
749	West Bengal	2020-02-29	7.55	10871168.0	44.09	Urban	Feb
750	West Bengal	2020-03-31	6.67	10806105.0	43.34	Urban	Mar
751	West Bengal	2020-04-30	15.63	9299466.0	41.20	Urban	Apr
752	West Bengal	2020-05-31	15.22	9240903.0	40.67	Urban	May
753	West Bengal	2020-06-30	9.86	9088931.0	37.57	Urban	Jun

740 rows × 7 columns

```
df_11_2020.drop(columns=['Frequency','month_int'])
```

	State	Date	Estimated Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Region	Longitude	Latitude	month
0	Andhra Pradesh	2020-01-31	5.48	16635535	41.02	South	15.9129	79.740	Jan
1	Andhra Pradesh	2020-02-29	5.83	16545652	40.90	South	15.9129	79.740	Feb
2	Andhra Pradesh	2020-03-31	5.79	15881197	39.18	South	15.9129	79.740	Mar
3	Andhra Pradesh	2020-04-30	20.51	11336911	33.10	South	15.9129	79.740	Apr
4	Andhra Pradesh	2020-05-31	17.43	12988845	36.46	South	15.9129	79.740	May
...	...	...	...	...	...	...	...	...	...
262	West Bengal	2020-06-30	7.29	30726310	40.39	East	22.9868	87.855	Jun
263	West Bengal	2020-07-31	6.83	35372506	46.17	East	22.9868	87.855	Jul

```
# Top 10 states with the highest unemployment
df1 = df[['State','Estimated Unemployment Rate']].groupby('State').sum().sort_values('Estimated Unemployment Rate' , ascending=False)
```

```
df1_11_2020 = df_11_2020[['State','Estimated Unemployment Rate']].groupby('State').sum().sort_values('Estimated Unemployment Rate' , ascending=False)
```

```
df1.head(10)
```

State	Estimated Unemployment Rate
Tripura	793.81
Haryana	735.93
Jharkhand	576.38
Bihar	529.71
Himachal Pradesh	519.13
Delhi	461.87
Rajasthan	393.63
Uttar Pradesh	351.44
Jammu & Kashmir	339.96
Punjab	336.87

Next steps: Generate code with df1 View recommended plots

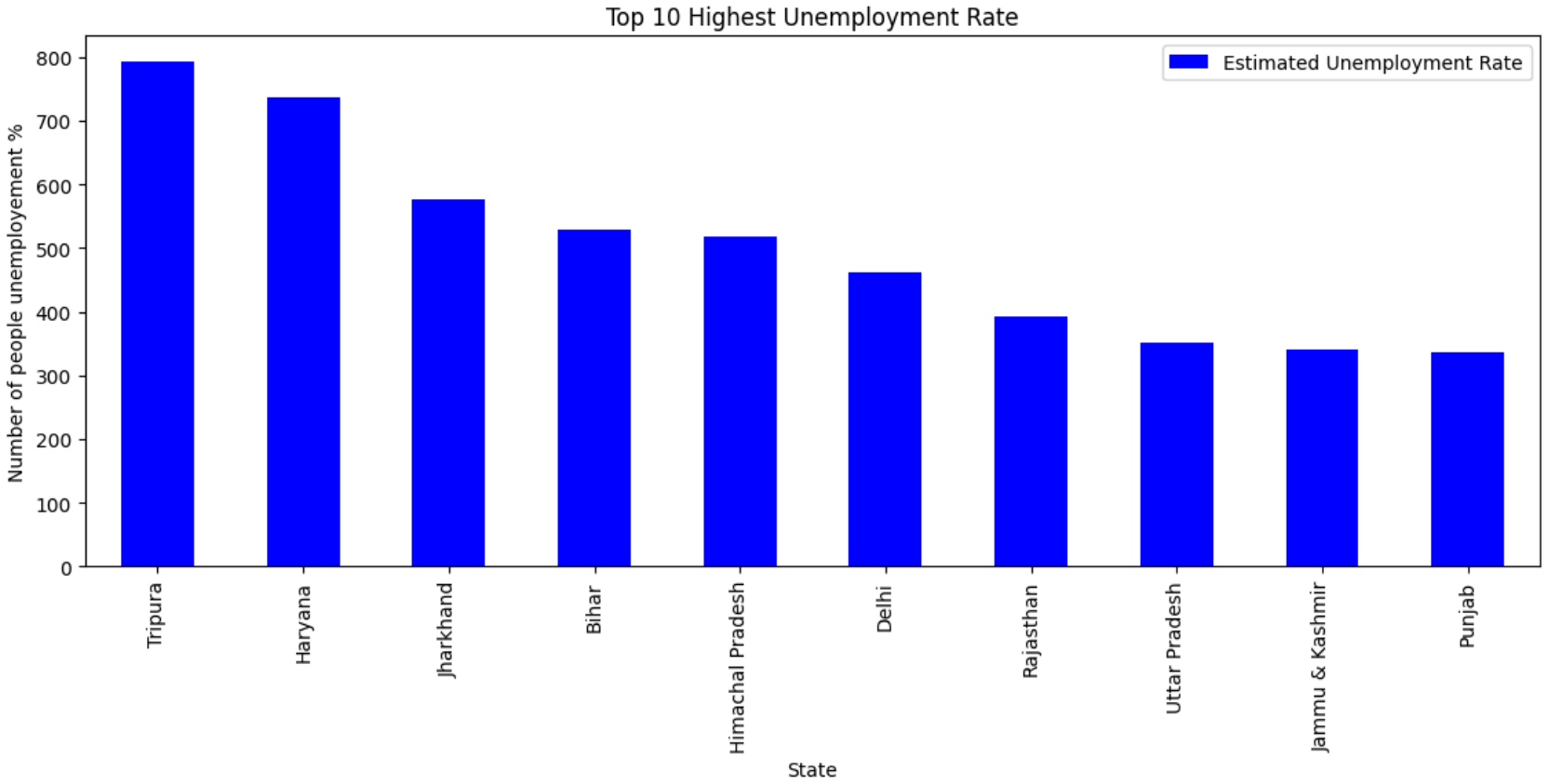
```
df1_11_2020.head(10)
```

State	Estimated Unemployment Rate
Haryana	274.77
Tripura	250.55
Jharkhand	195.39
Bihar	194.71
Delhi	184.14
Puducherry	179.42
Himachal Pradesh	160.65
Rajasthan	158.68
Jammu & Kashmir	148.30
Tamil Nadu	121.87

Next steps: Generate code with df1_11_2020 View recommended plots

```
# Visualisation of this top 10 highest unemployment

fig = plt.figure()
axb = fig.add_subplot(1,2,1)
df1[:10].plot(kind='bar' , color = 'blue' , figsize=(30,5), ax=axb)
axb.set_title('Top 10 Highest Unemployment Rate')
axb.set_xlabel('State')
axb.set_ylabel('Number of people unemployement %')
```

```
Text(0, 0.5, 'Number of people unemployement %')
```




```
# Month with the highest unemployment rate
df2 = df[['month','Estimated Unemployment Rate']].groupby('month').sum().sort_values('Estimated Unemployment Rate',ascending = False)
df2.head(12)
```

month	Estimated Unemployment Rate
May	1747.85
Apr	1205.72
Jun	1097.56
Mar	556.43
Oct	544.55
Nov	542.76
Feb	528.13
Jan	527.39
Aug	510.81
Dec	503.36
Jul	487.83
Sep	470.69

Next steps: Generate code with df2 View recommended plots

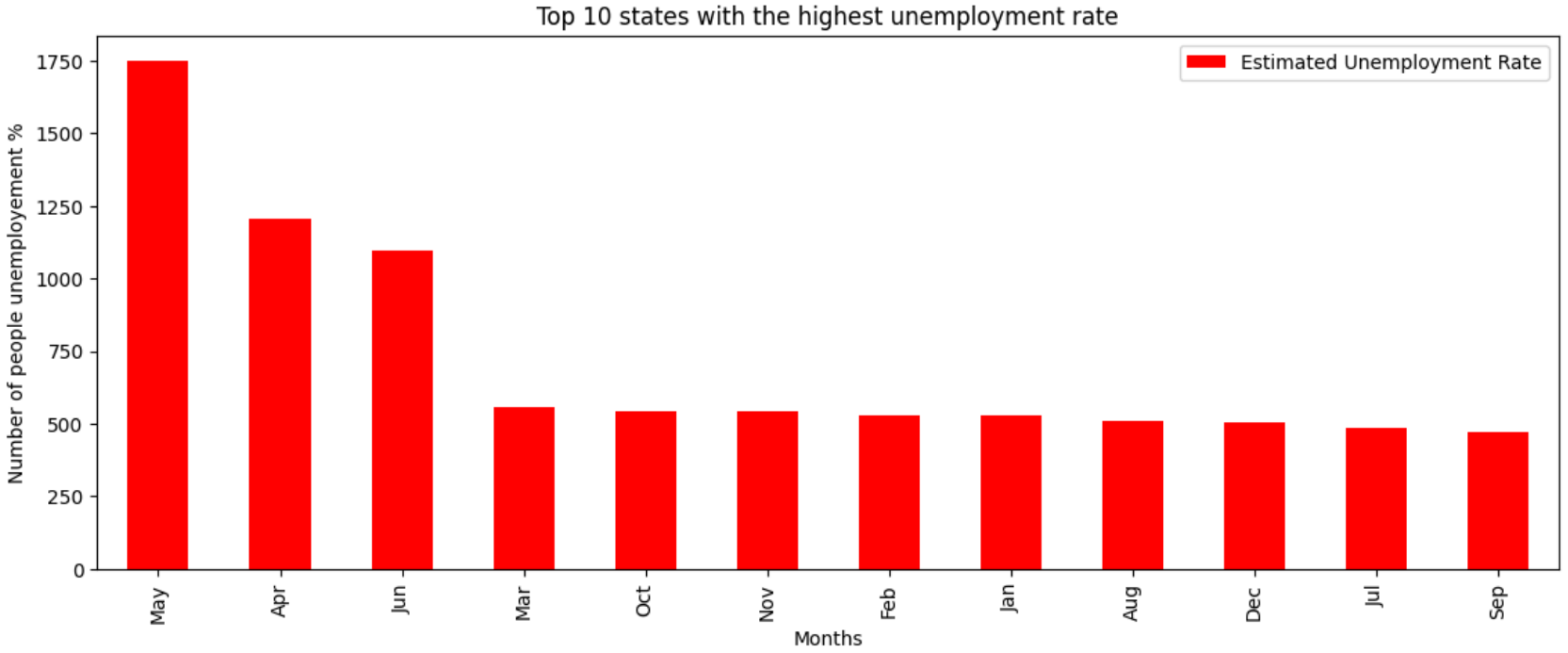
```
df2_11_2020 = df_11_2020[['month','Estimated Unemployment Rate']].groupby('month').sum().sort_values('Estimated Unemployment Rate',ascending = False)
df2.head(12)
```

month	Estimated Unemployment Rate
May	1747.85
Apr	1205.72
Jun	1097.56
Mar	556.43
Oct	544.55
Nov	542.76
Feb	528.13
Jan	527.39
Aug	510.81
Dec	503.36
Jul	487.83
Sep	470.69

Next steps: Generate code with df2 View recommended plots

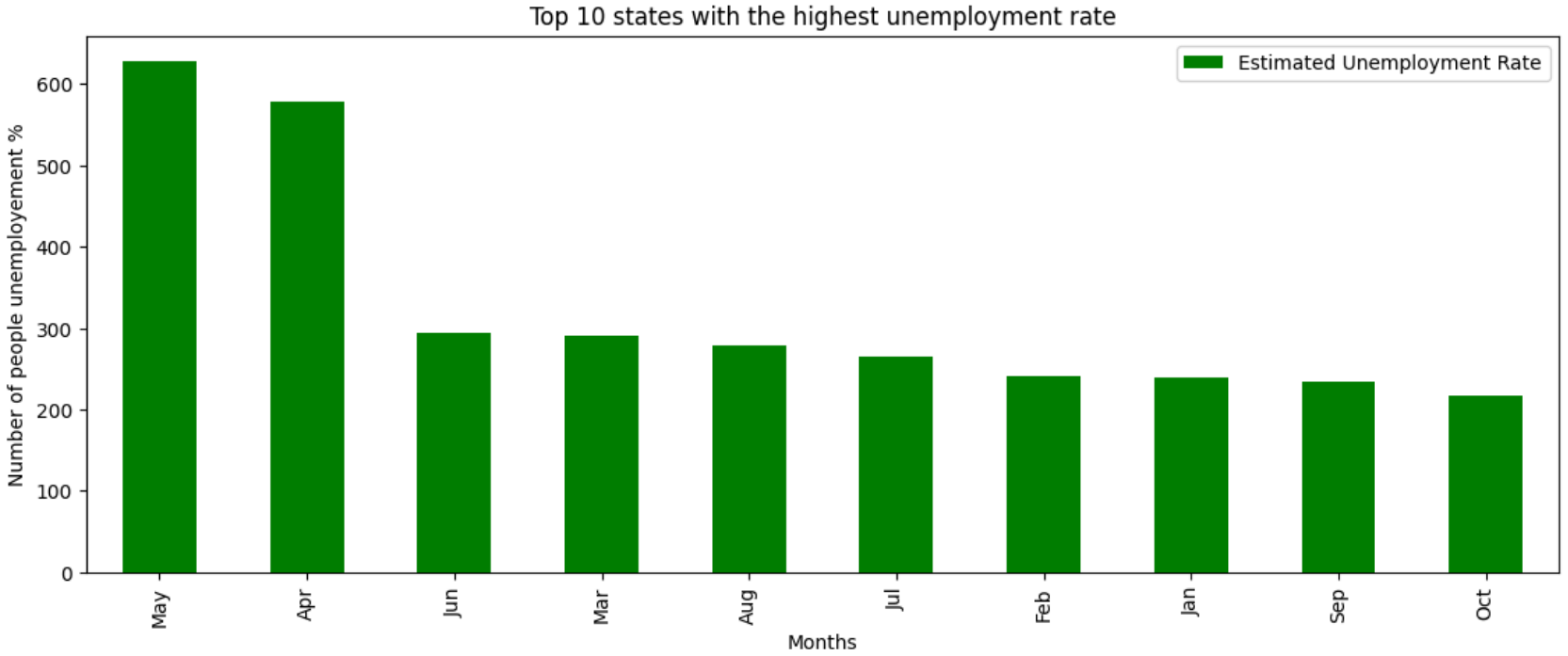
```
fig = plt.figure()
ax0 = fig.add_subplot(1,2,1)
df2[:12].plot(kind='bar' , color = 'red' , figsize = (30,5), ax = ax0)
ax0.set_title('Top 10 states with the highest unemployment rate')
ax0.set_xlabel('Months')
ax0.set_ylabel('Number of people unemployement %')
```

```
Text(0, 0.5, 'Number of people unemployement %')
```

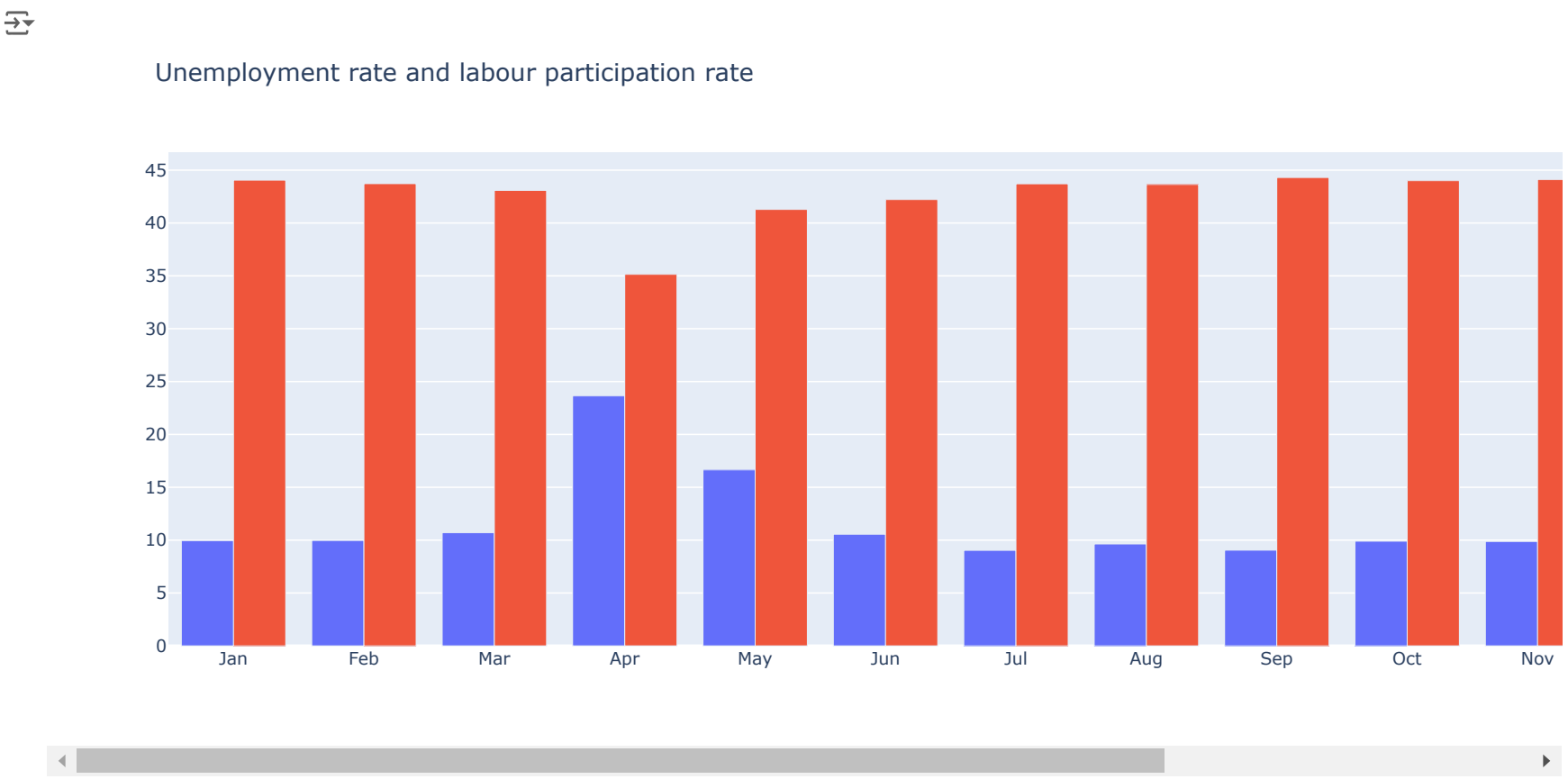



```
fig = plt.figure()
ax0 = fig.add_subplot(1,2,1)
df2_11_2020[:12].plot(kind='bar' , color = 'green' , figsize = (30,5), ax = ax0)
ax0.set_title('Top 10 states with the highest unemployment rate')
ax0.set_xlabel('Months')
ax0.set_ylabel('Number of people unemployement %')
```

```
Text(0, 0.5, 'Number of people unemployement %')
```

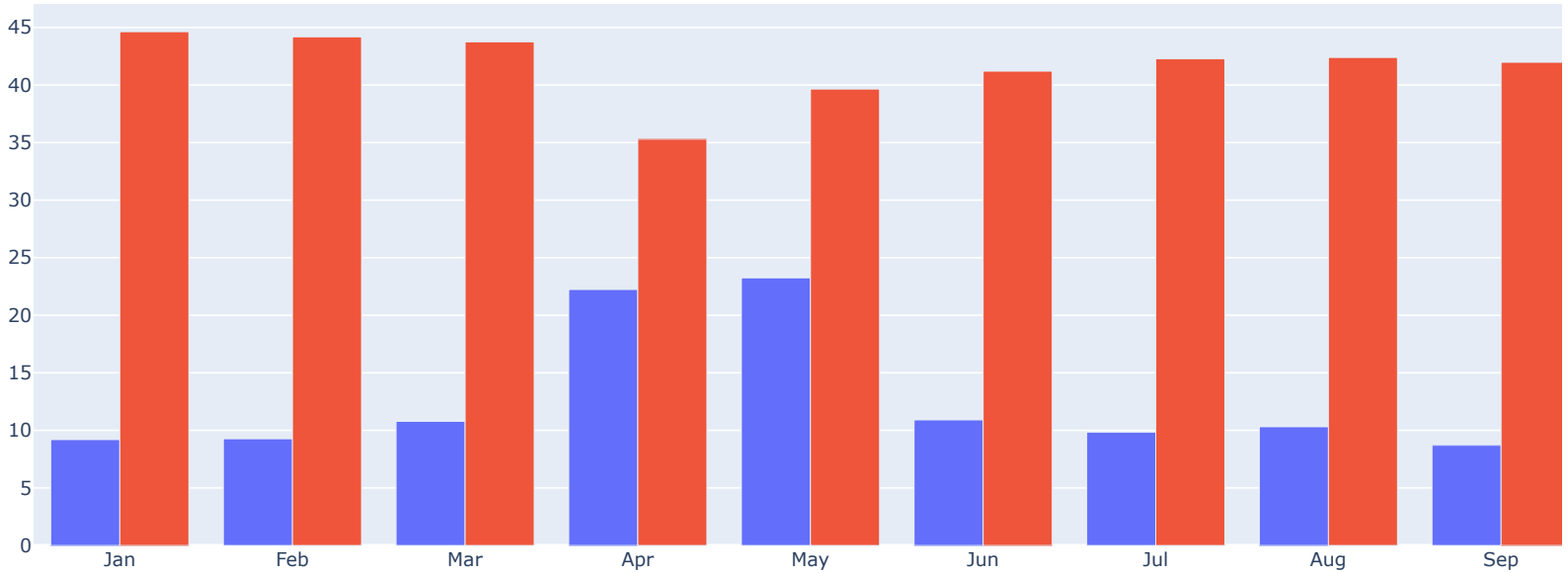



```
#Visualize labour participation rate & unemployment rate in each month

df_EE = df.groupby(['month'])[['Estimated Unemployment Rate','Estimated Employed','Estimated Labour Participation Rate']].mean()
df_EE = pd.DataFrame(df_EE).reset_index()
month = df_EE.month
unemployment_rate = df_EE['Estimated Unemployment Rate']
labour_participation_rate = df_EE['Estimated Labour Participation Rate']

fig = gg.Figure()

fig.add_trace(gg.Bar(x = month, y = unemployment_rate , name='Unemployment_Rate'))
fig.add_trace(gg.Bar(x = month , y = labour_participation_rate , name='Labour_Participation_Rate'))

fig.update_layout(title = 'Unemployment rate and labour participation rate' , xaxis = {'categoryorder':'array', 'categoryarray':['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','S

fig.show()
```




```
df_EE_11_2020 = df_11_2020.groupby(['month'])[['Estimated Unemployment Rate','Estimated Employed','Estimated Labour Participation Rate']].mean()
df_EE_11_2020 = pd.DataFrame(df_EE_11_2020).reset_index()
month = df_EE_11_2020.month
unemployment_rate = df_EE_11_2020['Estimated Unemployment Rate']
labour_participation_rate = df_EE_11_2020['Estimated Labour Participation Rate']

fig = gg.Figure()

fig.add_trace(gg.Bar(x = month, y = unemployment_rate , name='Unemployment_Rate'))
fig.add_trace(gg.Bar(x = month , y = labour_participation_rate , name='Labour Participation Rate'))

fig.update_layout(title = 'Unemployment rate and labour participation rate for upto 11/2020' , xaxis = {'categoryorder':'array', 'categoryarray':['Jan','Feb','Mar','Apr','May','Ju

fig.show()
```

Unemployment rate and labour participation rate for upto 11/2020



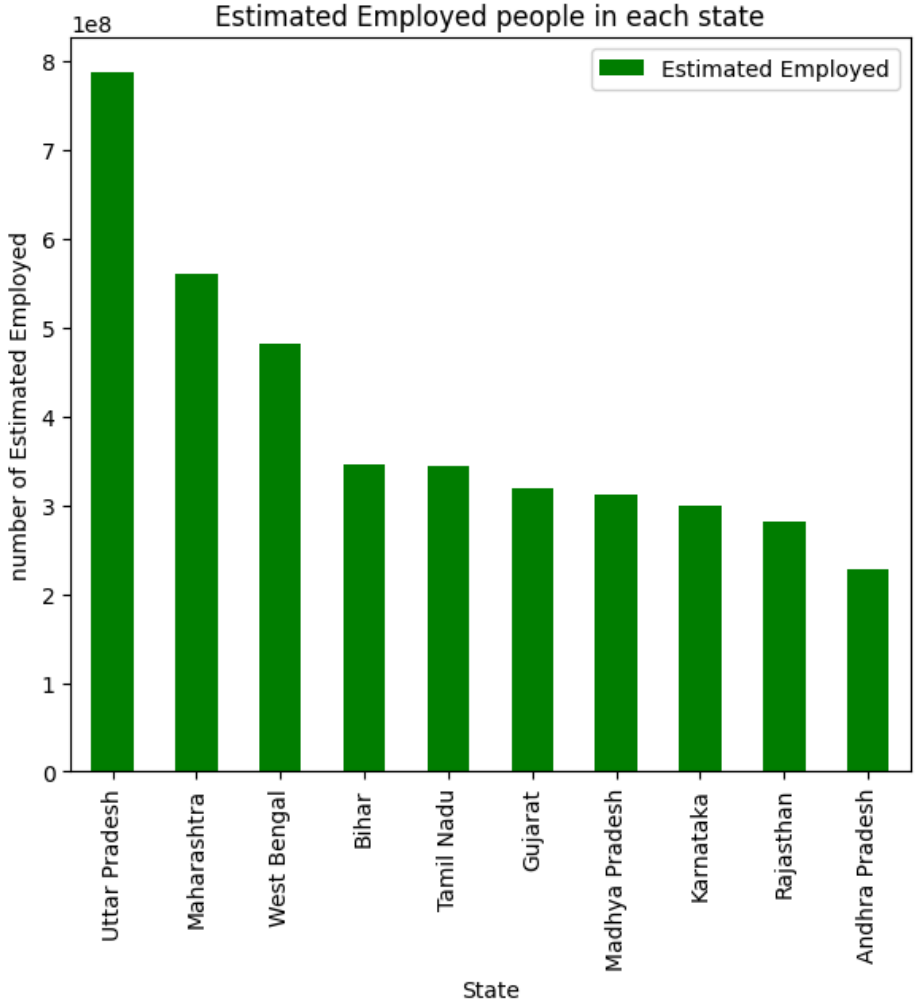

```
# state wise estimated employed

df3 = df[['State','Estimated Employed']].groupby('State').sum().sort_values('Estimated Employed', ascending=False)
df3
```

State	Estimated Employed
Uttar Pradesh	786655301.0
Maharashtra	559725484.0
West Bengal	481559064.0
Bihar	346253296.0
Tamil Nadu	343547309.0
Gujarat	319256358.0
Madhya Pradesh	311233561.0
Karnataka	298679340.0
Rajasthan	281149813.0
Andhra Pradesh	228314609.0
Telangana	222310557.0
Odisha	183280915.0
Assam	139224076.0
Punjab	127102136.0
Jharkhand	125138732.0
Kerala	123925186.0
Chhattisgarh	120497960.0
Haryana	99598029.0
Delhi	73570360.0
Jammu & Kashmir	37798565.0
Uttarakhand	37536159.0
Himachal Pradesh	29675064.0
Tripura	20076074.0
Meghalaya	18622894.0
Puducherry	5519230.0
Goa	5431400.0
Chandigarh	3801975.0
Sikkim	1816972.0

Next steps: Generate code with df3 View recommended plots

```
#State wise estimated employed visualization

fig = plt.figure()
ax1 = fig.add_subplot(1,2,1) # second subplot (ax1) will be positioned in the second coloumn

#Employed
df3[:10].plot(kind = 'bar', color='green', figsize=(15,6), ax = ax1)
ax1.set_title('Estimated Employed people in each state')
ax1.set_xlabel('State')
ax1.set_ylabel('number of Estimated Employed')
```

```
Text(0, 0.5, 'number of Estimated Employed')
```

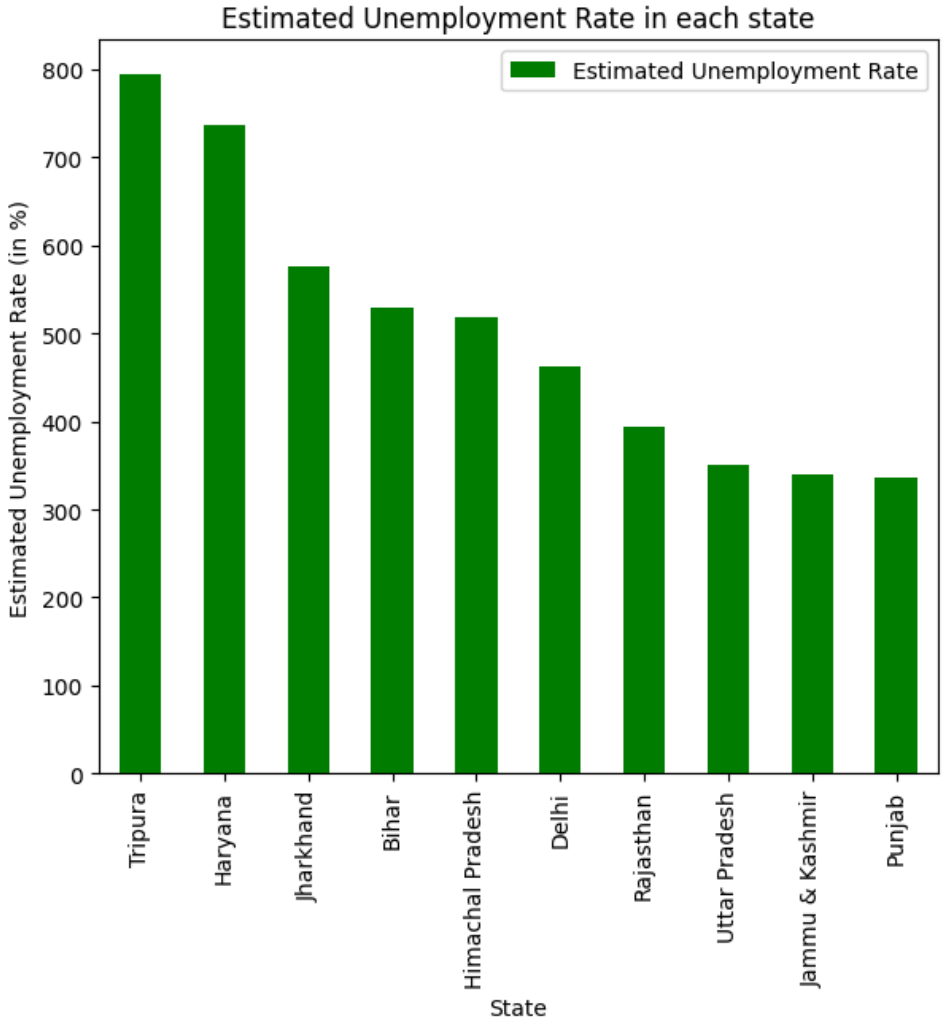


```
# Estimated unemployment rate State wise
# Estimated Unemployment rate (%) = (Number of Unemployed / Labour force ) * 100

df3_a = df[['State', 'Estimated Unemployment Rate']].groupby('State').sum().sort_values('Estimated Unemployment Rate', ascending=False)
df3_a
```

State	Estimated Unemployment Rate
Tripura	793.81
Haryana	735.93
Jharkhand	576.38
Bihar	529.71
Himachal Pradesh	519.13
Delhi	461.87
Rajasthan	393.63
Uttar Pradesh	351.44
Jammu & Kashmir	339.96
Punjab	336.87
Kerala	283.47
Puducherry	265.59
Tamil Nadu	259.96
Chhattisgarh	258.73
West Bengal	227.49
Goa	222.58
Telangana	216.66
Maharashtra	211.61
Andhra Pradesh	209.36
Madhya Pradesh	207.38
Chandigarh	191.90
Karnataka	186.93
Gujarat	186.59
Uttarakhand	177.74
Assam	167.13
Odisha	158.42
Meghalaya	129.57
Sikkim	123.24

```
# Estimated unemployment rate state wise visualization

fig = plt.figure()
ax1 = fig.add_subplot(1,2,2)

df3_a[:10].plot(kind='bar',color='green', figsize=(15,6), ax = ax1)
ax1.set_title('Estimated Unemployment Rate in each state')
ax1.set_xlabel('State')
ax1.set_ylabel('Estimated Unemployment Rate (in %)')
```

```
Text(0, 0.5, 'Estimated Unemployment Rate (in %)')
```



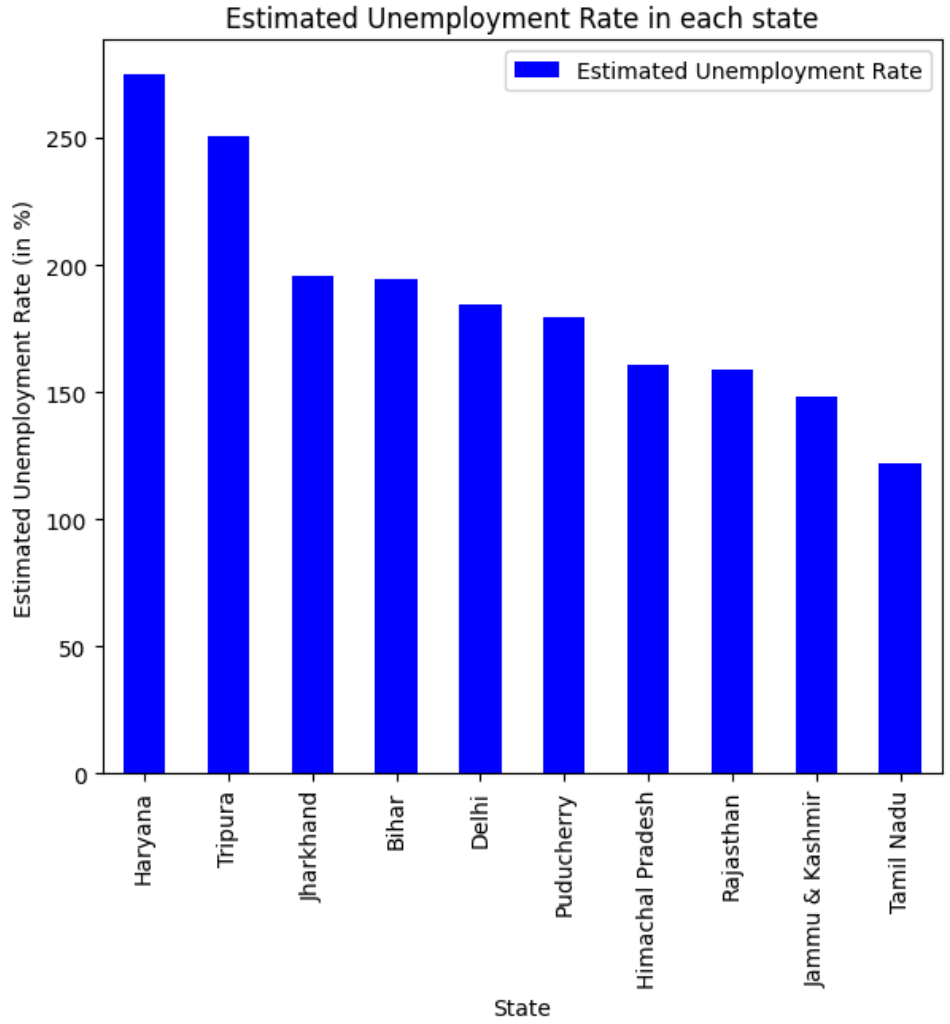

```
df3_a_11_2020 = df_11_2020[['State', 'Estimated Unemployment Rate']].groupby('State').sum().sort_values('Estimated Unemployment Rate', ascending=False)
df3_a_11_2020
```

State	Estimated Unemployment Rate
Haryana	274.77
Tripura	250.55
Jharkhand	195.39
Bihar	194.71
Delhi	184.14
Puducherry	179.42
Himachal Pradesh	160.65
Rajasthan	158.68
Jammu & Kashmir	148.30
Tamil Nadu	121.87
Goa	121.67
Punjab	119.81
Uttarakhand	111.56
West Bengal	101.92
Uttar Pradesh	97.37
Kerala	94.34
Andhra Pradesh	86.64
Maharashtra	79.79
Sikkim	78.34
Chhattisgarh	78.19
Karnataka	76.68
Madhya Pradesh	68.54
Telangana	68.33
Odisha	64.62
Gujarat	63.76
Assam	48.56
Meghalaya	38.66

Next steps: Generate code with df3_a_11_2020 View recommended plots

```
fig = plt.figure()
ax0 = fig.add_subplot(1,2,1)

df3_a_11_2020[:10].plot(kind='bar',color='blue', figsize=(15,6), ax = ax0)
ax0.set_title('Estimated Unemployment Rate in each state')
ax0.set_xlabel('State')
ax0.set_ylabel('Estimated Unemployment Rate (in %)')
```

```
Text(0, 0.5, 'Estimated Unemployment Rate (in %)')
```




```
df3_11_2020 = df_11_2020[['State', 'Estimated Employed']].groupby('State').sum().sort_values('Estimated Employed', ascending=False)
df3_11_2020
```

State	Estimated Employed
Uttar Pradesh	555247990
Maharashtra	392047582
West Bengal	333051643
Bihar	236068280
Gujarat	227307461
Madhya Pradesh	223183353
Tamil Nadu	219878981
Karnataka	216240176
Rajasthan	197317522
Telangana	162440825
Andhra Pradesh	154254800
Odisha	127268329
Assam	108102755
Punjab	87830342
Jharkhand	87706424
Kerala	85967949
Chhattisgarh	84213492
Haryana	68440590
Delhi	46328219
Jammu & Kashmir	29790285
Uttarakhand	27432749
Himachal Pradesh	20338849
Tripura	13972916
Meghalaya	13498153
Goa	4423748
Puducherry	3652629
Sikkim	1876186

Next steps: Generate code with df3_11_2020 View recommended plots

```
fig = plt.figure()
ax1 = fig.add_subplot(1,2,1) # second subplot (ax1) will be positioned in the second coloumn

#Employed
df3_11_2020[:10].plot(kind = 'bar', color='blue', figsize=(15,6), ax = ax1)
ax1.set_title('Estimated Employed people in each state')
ax1.set_xlabel('State')
ax1.set_ylabel('number of Estimated Employed')
```

```
Text(0, 0.5, 'number of Estimated Employed')
```