

## merge

September 10, 2023

```
[1]: print("hh")
```

hh

```
[2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.io as pio
```

```
[3]: # since both the files contain same data for a different duration of time, we
      ↪are merging both the files into one file
df1 = pd.read_csv('1.csv')
df2 = pd.read_csv('2.csv')
```

```
[4]: df = pd.concat([df1, df2], axis=0)
df.reset_index(drop=True, inplace=True)
#modified data such that for a state all the data appears in a single unbroken
      ↪chain
```

```
[5]: df.to_csv('merged_data.csv', index=False)
df.
      ↪columns=['State', 'Date', 'freq', 'UR', 'employed', 'LPR', 'Area', 'region', 'lon', 'lat']
```

```
[6]: print(df.head()) #prints top5 rows of csv
```

	State	Date	freq	UR	employed	LPR	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	

	region	lon	lat
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN

```
3    NaN NaN NaN
4    NaN NaN NaN
```

```
[7]: df.info
```

```
[7]: <bound method DataFrame.info of
UR      employed    LPR    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
...
1030    West Bengal  30-06-2020    M  7.29  30726310.0  40.39  NaN
1031    West Bengal  31-07-2020    M  6.83  35372506.0  46.17  NaN
1032    West Bengal  31-08-2020    M 14.87  33298644.0  47.48  NaN
1033    West Bengal  30-09-2020    M  9.35  35707239.0  47.73  NaN
1034    West Bengal  31-10-2020    M  9.98  33962549.0  45.63  NaN

      region    lon    lat
0      NaN    NaN    NaN
1      NaN    NaN    NaN
2      NaN    NaN    NaN
3      NaN    NaN    NaN
4      NaN    NaN    NaN
...
1030    East  22.9868  87.855
1031    East  22.9868  87.855
1032    East  22.9868  87.855
1033    East  22.9868  87.855
1034    East  22.9868  87.855

[1035 rows x 10 columns]>
```

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

```
State      28
Date       28
freq       28
UR         28
employed   28
LPR        28
Area      295
region     768
lon        768
lat        768
dtype: int64
```

```
[9]: un_rate = 'UR'
lp_rate = 'LPR'
ee = 'employed'
print("Mean and Median of various columns")
print("
    Mean
    ↪Median")
print("Estimated Unemployment Rate (%)      =",df[un_rate].mean(),"    ↪
    ↪",df[un_rate].median())
print("Estimated Labour Participation Rate (%) =",df[lp_rate].mean(),"    ↪
    ↪",df[lp_rate].median())

print("Estimated Employed                    =",df[ee].mean(),"    ",df[ee].
    ↪median())
df.columns = df.columns.str.strip()
```

Mean and Median of various columns

	Mean	Median
Estimated Unemployment Rate (%)	= 11.906991062562065	8.89
Estimated Labour Participation Rate (%)	= 42.37861966236346	40.88
Estimated Employed	= 8996209.182720954	5543380.0

```
[10]: grouped=df.groupby('State')
mean_values = grouped['employed'].mean()
median_values = grouped['employed'].median()
umprateme = grouped['UR'].mean()
umprate_m = grouped['UR'].median()
lprm = grouped['LPR'].mean()
lpr_m = grouped['LPR'].median()
summary_df = pd.DataFrame({'Est emp Mean': mean_values, 'Est emp Median':↪
    ↪median_values, 'Est unp % Mean' : umprateme, 'Est unp % Median':umprate_m, 'Est↪
    ↪labour partitipation rate % Mean' : lprm, 'Est labour partitipation rate %↪
    ↪Median':lpr_m})
print(summary_df)
```

State	Est emp Mean	Est emp Median	Est unp % Mean \
Andhra Pradesh	1.006762e+07	11560649.0	7.789474
Assam	6.870190e+06	8693018.0	5.991389
Bihar	1.532425e+07	21441384.5	19.063684
Chandigarh	3.168312e+05	322063.0	15.991667
Chhattisgarh	5.387143e+06	6512548.5	8.866316
Delhi	3.155226e+06	4298930.0	17.000263
Goa	2.898573e+05	278730.0	10.125000
Gujarat	1.438326e+07	13382199.0	6.588158
Haryana	4.422069e+06	4564207.0	26.597368
Himachal Pradesh	1.316156e+06	1907231.5	17.888947
Jammu & Kashmir	2.252962e+06	2419733.0	16.275333
Jharkhand	5.601188e+06	6624275.5	20.309737

Karnataka	1.355051e+07	12720063.5	6.937105
Kerala	5.523504e+06	4661442.0	9.942368
Madhya Pradesh	1.406360e+07	16180873.0	7.261053
Maharashtra	2.504666e+07	23748937.0	7.668421
Meghalaya	8.681364e+05	1080609.0	4.546757
Odisha	8.172349e+06	10576149.5	5.869474
Puducherry	2.547739e+05	276927.0	12.361389
Punjab	5.656118e+06	6023578.0	12.017895
Rajasthan	1.259125e+07	15055910.0	14.534474
Sikkim	1.477263e+05	133399.0	8.063200
Tamil Nadu	1.482701e+07	13457093.0	10.048158
Telangana	1.012504e+07	10898656.0	7.499737
Tripura	8.960261e+05	1026978.5	27.483158
Uttar Pradesh	3.531324e+07	42765132.5	11.810789
Uttarakhand	1.755916e+06	1972074.0	7.818919
West Bengal	2.143712e+07	24482594.5	8.668684

State	Est unp % Median	Est labour participation rate % Mean \
Andhra Pradesh	5.810	39.266842
Assam	4.895	44.487778
Bihar	14.485	37.895789
Chandigarh	17.510	39.336667
Chhattisgarh	8.265	42.376579
Delhi	15.510	38.121053
Goa	10.440	39.247353
Gujarat	5.420	45.940263
Haryana	25.610	42.569474
Himachal Pradesh	17.275	43.177368
Jammu & Kashmir	16.155	40.090000
Jharkhand	16.035	41.324737
Karnataka	3.590	41.517632
Kerala	7.645	34.476842
Madhya Pradesh	4.870	38.848947
Maharashtra	6.155	42.082895
Meghalaya	3.730	57.831622
Odisha	3.725	38.616316
Puducherry	1.750	38.138611
Punjab	10.760	40.833158
Rajasthan	14.130	40.135789
Sikkim	5.700	45.969600
Tamil Nadu	3.045	39.953947
Telangana	5.780	53.520789
Tripura	27.310	60.777632
Uttar Pradesh	10.740	38.952368
Uttarakhand	6.340	34.176486
West Bengal	7.245	45.349211

State	Est labour partitipation rate % Median
Andhra Pradesh	37.995
Assam	44.290
Bihar	38.095
Chandigarh	39.360
Chhattisgarh	43.455
Delhi	38.130
Goa	39.385
Gujarat	46.310
Haryana	42.665
Himachal Pradesh	44.140
Jammu & Kashmir	39.430
Jharkhand	41.105
Karnataka	41.015
Kerala	36.105
Madhya Pradesh	39.220
Maharashtra	41.740
Meghalaya	59.880
Odisha	39.935
Puducherry	36.740
Punjab	41.770
Rajasthan	40.735
Sikkim	45.610
Tamil Nadu	39.025
Telangana	55.185
Tripura	61.340
Uttar Pradesh	39.420
Uttarakhand	34.030
West Bengal	46.070

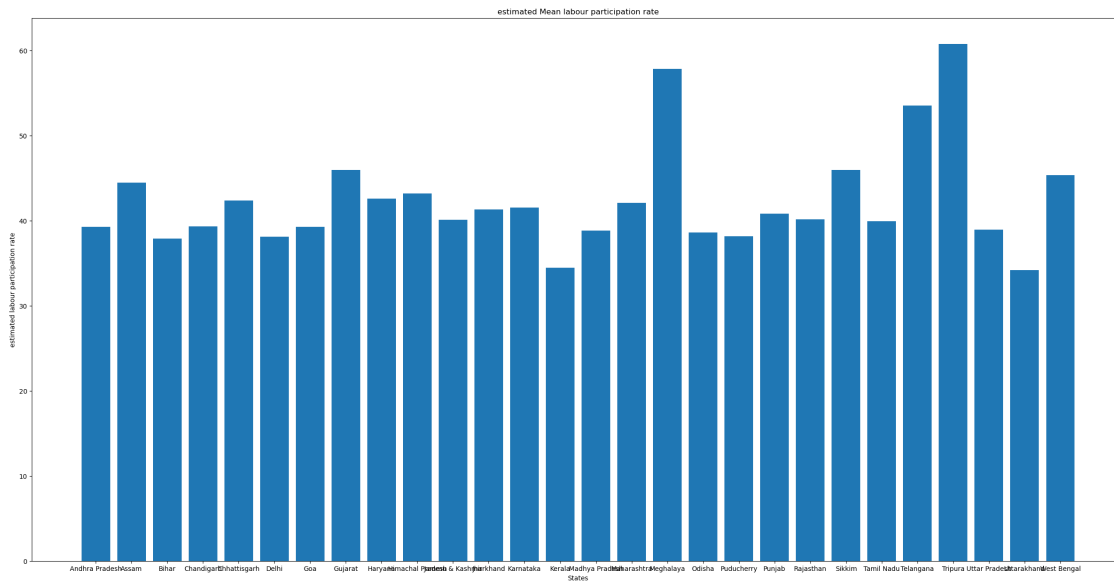
```
[11]: mean_values = grouped['employed'].mean().tolist()
median_values = grouped['employed'].median().tolist()
umpratem = grouped['UR'].mean().tolist()
umprate_m = grouped['UR'].median().tolist()
lprm = grouped['LPR'].mean().tolist()
lpr_m = grouped['LPR'].median().tolist()
gpred = grouped.groups.keys()

#plotting estimated labour participation rate mean

plt.figure(figsize=(30,15))
plt.bar(gpred,lprm)
plt.xlabel('States')
plt.ylabel('estimated labour participation rate')

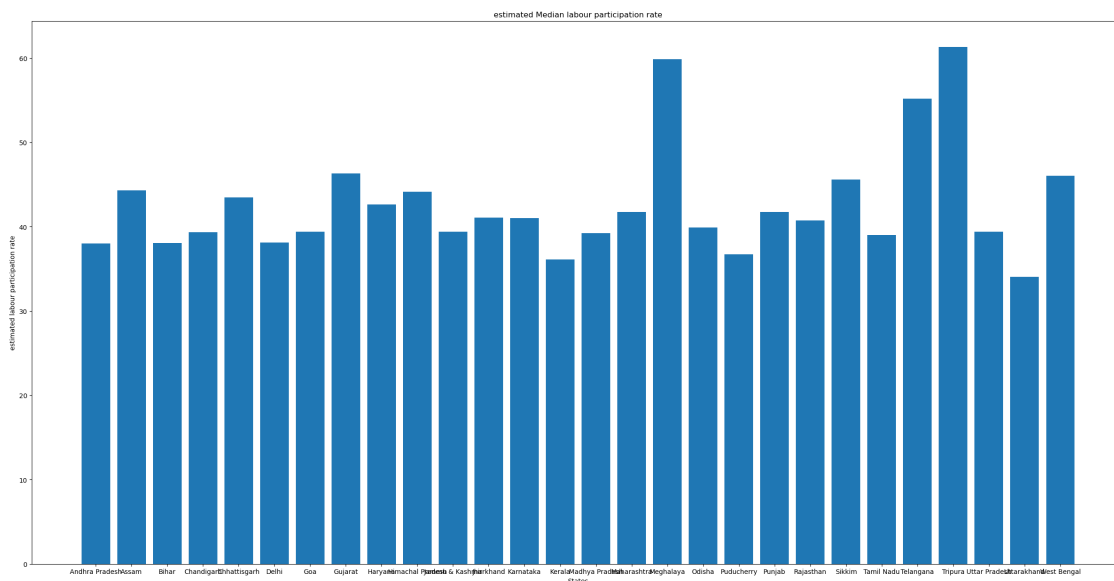
plt.title('estimated Mean labour participation rate')
```

```
plt.show()
```



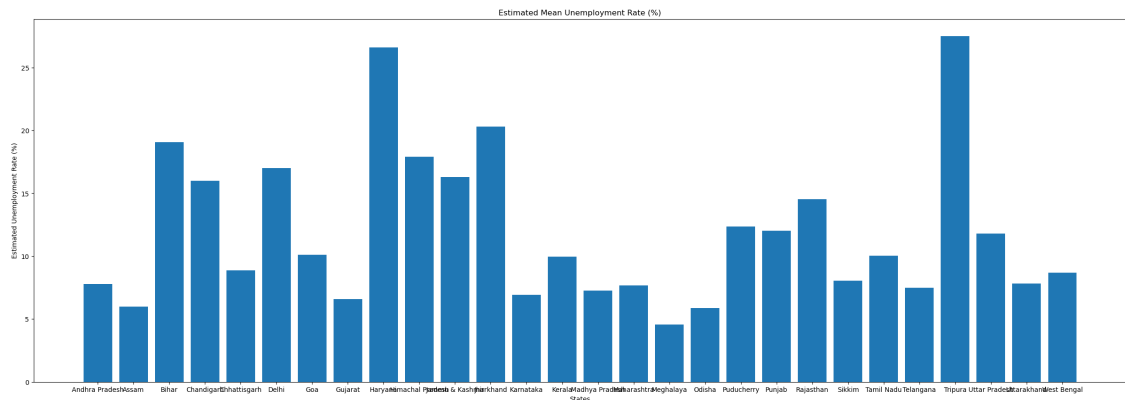
```
[12]: plt.figure(figsize=(30,15))
plt.bar(gpred,lpr_m)
plt.xlabel('States')
plt.ylabel('estimated labour participation rate')

plt.title('estimated Median labour participation rate')
plt.show()
```



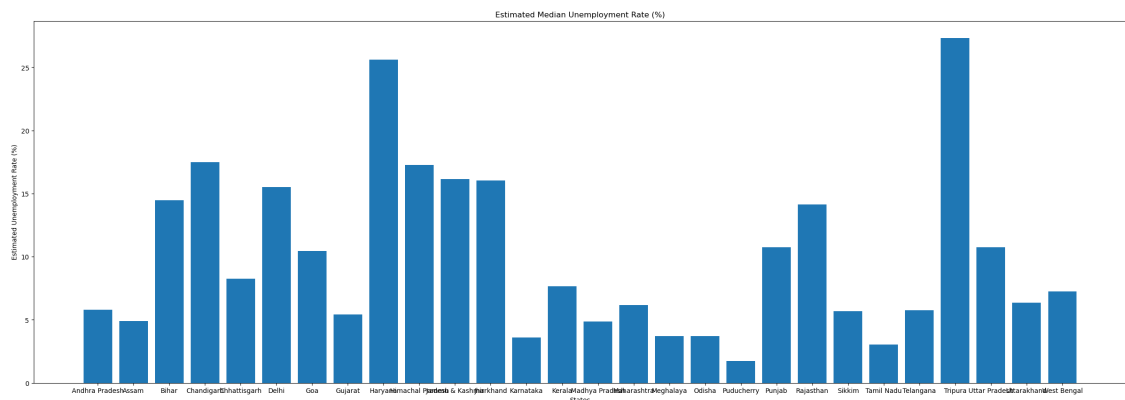
```
plt.figure(figsize=(30,10))
plt.bar(gpred,umpratem)
plt.xlabel('States')
plt.ylabel('Estimated Unemployment Rate (%)')

plt.title('Estimated Mean Unemployment Rate (%)')
plt.show()
```



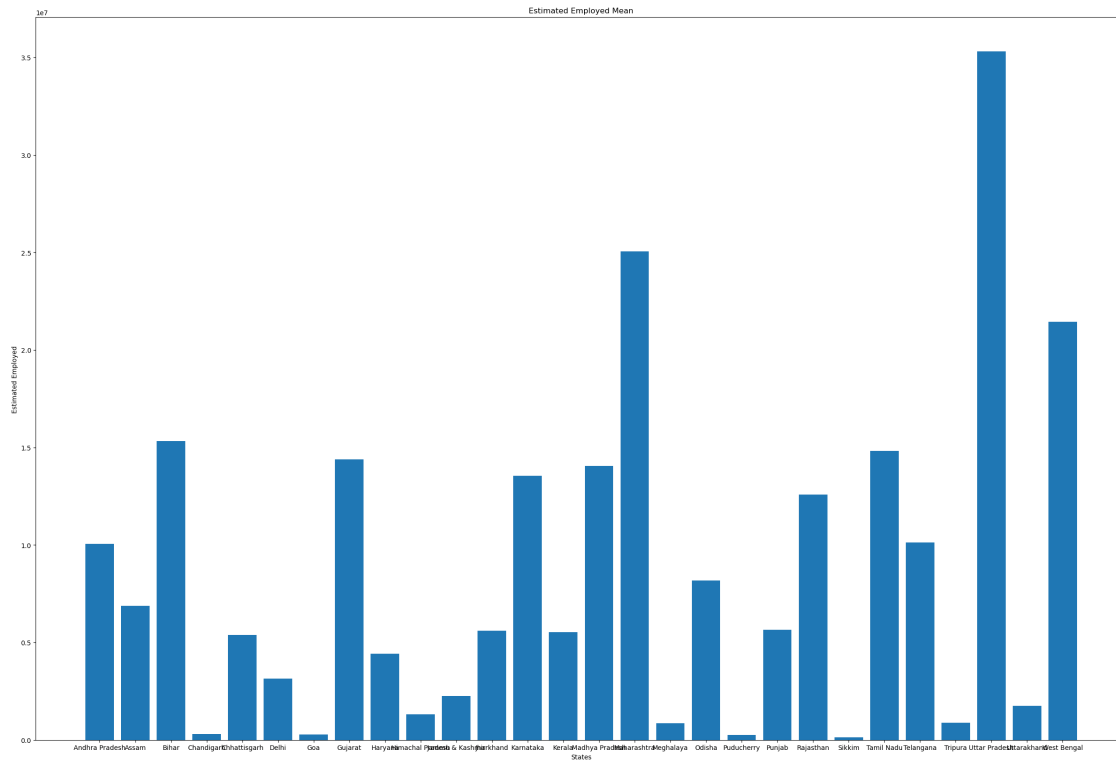
```
plt.figure(figsize=(30,10))
plt.bar(gpred,umprate_m)
plt.xlabel('States')
plt.ylabel('Estimated Unemployment Rate (%)')

plt.title('Estimated Median Unemployment Rate (%)')
plt.show()
```



```
[15]: plt.figure(figsize=(30,20))
plt.bar(gpred,mean_values)
plt.xlabel('States')
plt.ylabel('Estimated Employed ')

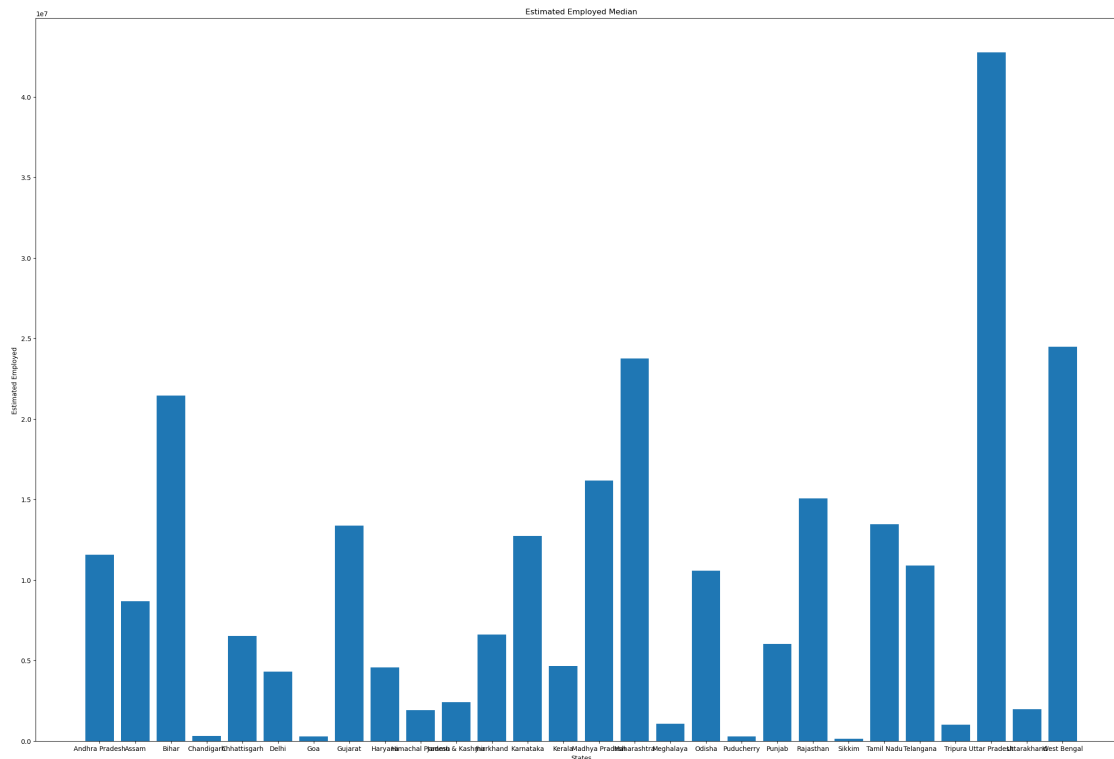
plt.title('Estimated Employed Mean')
plt.show()
```



```
[16]: plt.figure(figsize=(30,20))
plt.bar(gpmed,median_values)
plt.xlabel('States')
plt.ylabel('Estimated Employed ')

plt.title('Estimated Employed Median')
plt.show()
```

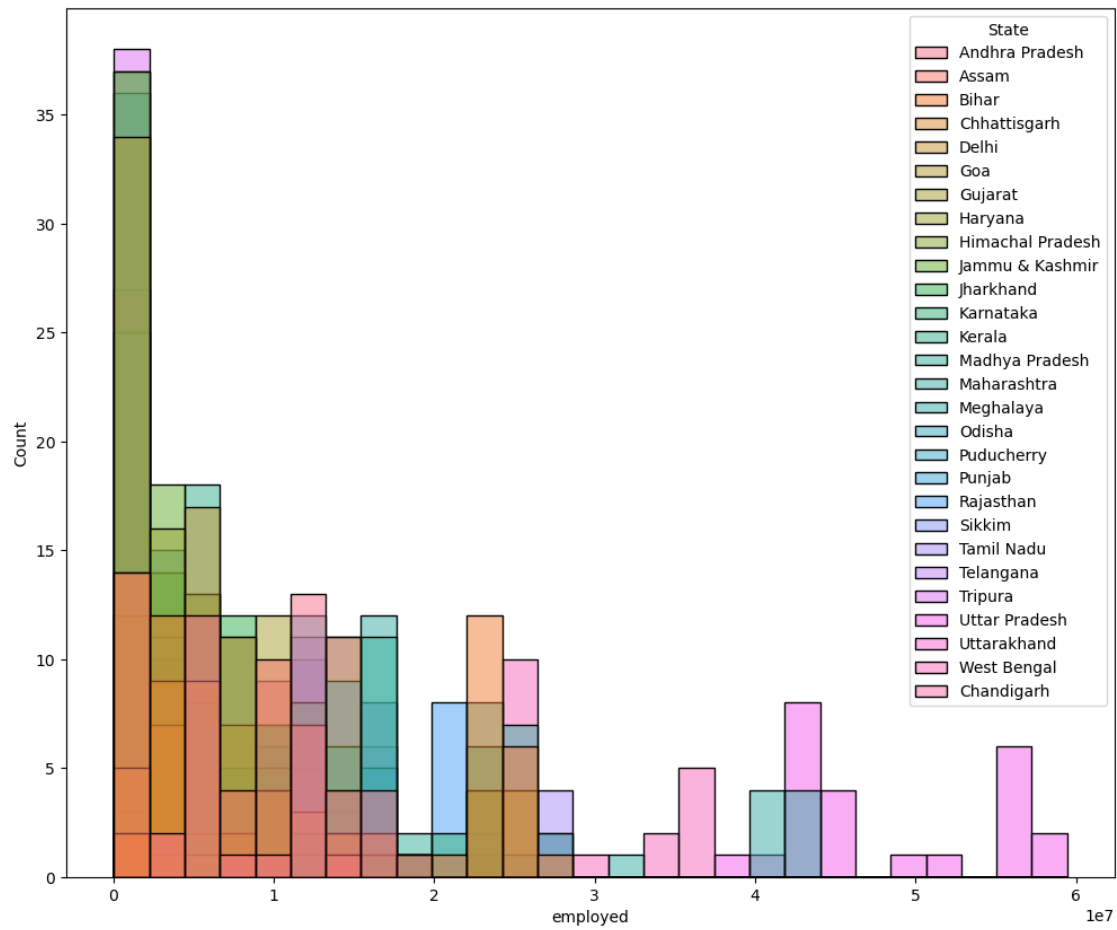




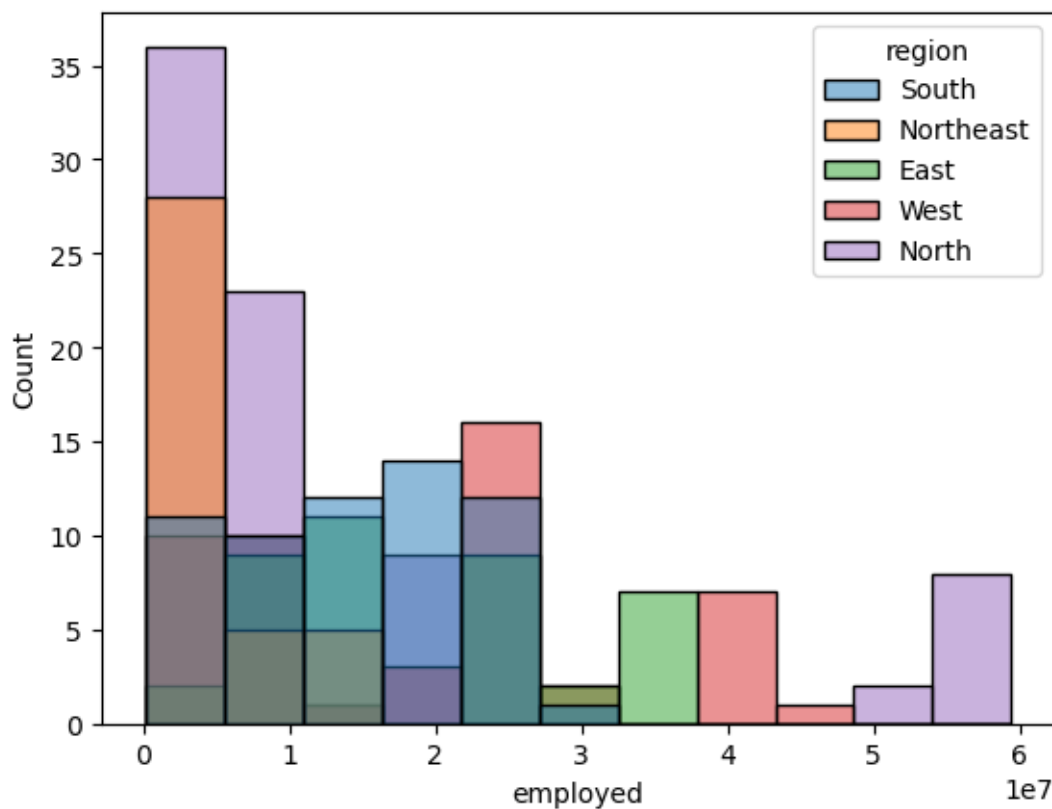
[ ]:

[ ]:

```
[17]: #ploting according to regions and state
plt.figure(figsize=(12,10))
sns.histplot(x="employed", hue="State", data=df)
plt.show()
sns.histplot(x="employed", hue="region", data=df)
```



[17]: <AxesSubplot:xlabel='employed', ylabel='Count'>



```
[18]: df.info
```

```
[18]: <bound method DataFrame.info of
UR    employed    LPR    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
...
1030   West Bengal  30-06-2020    M  7.29  30726310.0  40.39  NaN
1031   West Bengal  31-07-2020    M  6.83  35372506.0  46.17  NaN
1032   West Bengal  31-08-2020    M 14.87  33298644.0  47.48  NaN
1033   West Bengal  30-09-2020    M  9.35  35707239.0  47.73  NaN
1034   West Bengal  31-10-2020    M  9.98  33962549.0  45.63  NaN

region    lon    lat
0        NaN    NaN    NaN
1        NaN    NaN    NaN
2        NaN    NaN    NaN
3        NaN    NaN    NaN
```

4		NaN	NaN	NaN
...	...	...	...	
1030	East	22.9868	87.855	
1031	East	22.9868	87.855	
1032	East	22.9868	87.855	
1033	East	22.9868	87.855	
1034	East	22.9868	87.855	

[1035 rows x 10 columns]>

```
[21]: une = df[["State", 'region', "UR"]]
      une.dropna(subset=["region"], inplace=True)
      figure = px.sunburst (une, path = ["region", "State"], values="UR",
      ↪width=700,height=700, color_continuous_scale="RdY1Gn",title="Unemployment_
      ↪Rate in Indian")
      pio.write_image(figure, "sunburst_plot.png")

      from IPython.display import Image
      Image("sunburst_plot.png")
```

/tmp/ipykernel\_29008/2291208178.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

[21]:

## Unemployment Rate in Indian



[20]: *#thanks proejct done*

[ ]:

[ ]: