```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
import calendar


import datetime as dt

import plotly.io as pio
pio.templates
```

```
Templates configuration
-----------------------
    Default template: 'plotly'
    Available templates:
        ['ggplot2', 'seaborn', 'simple_white', 'plotly',
         'plotly_white', 'plotly_dark', 'presentation', 'xgridoff',
         'ygridoff', 'gridon', 'none']
```

```
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from IPython.display import HTML
```

Data Import

```
df = pd.read_csv('/content/Unemployment in India.csv')
```

```
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

```
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
```

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

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

```
df.Columns =['States','Date','Frequency','Estimated Unemployment Rate','Estimated Employed','Estimated Labour Participation Rate','Region','l
```

```
<ipython-input-8-633a7513f857>:1: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.
  df.Columns =['States','Date','Frequency','Estimated Unemployment Rate','Estimated Employed','Estimated Labour Participation Rate','Reg
```

```
print(df.Columns)
```

```
['States', 'Date', 'Frequency', 'Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate', 'Region', 'l
```

```
df.columns = df.columns.str.strip()


df['Date'] = pd.to_datetime(df['Date'],dayfirst=True)


df['Frequency']= df['Frequency'].astype('category')


df['Month'] =  df['Date'].dt.month


df['Month_int'] = df['Month'].apply(lambda x: int(x) if not pd.isnull(x) else 0)
#df['Month_int'] = df['Month'].apply(lambda x : int(x))


df['Month_name'] =  df['Month_int'].apply(lambda x: calendar.month_abbr[x])


df['Region'] = df['Region'].astype('category')


df.drop(columns='Month',inplace=True)
df.head(3)
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	Month_int	Mon
0	Andhra Pradesh	2019-05-31	Monthly	3.65	11999139.0	43.24	Rural	5	
1	Andhra Pradesh	2019-06-30	Monthly	3.05	11755881.0	42.05	Rural	6	
2	Andhra Pradesh	2019-07-31	Monthly	3.75	12086707.0	43.50	Rural	7	

Region

```
# Checking for null values
df['Region'].value_counts(dropna=False)
```

```
Andhra Pradesh      28
Karnataka           28
West Bengal         28
Uttar Pradesh       28
Tripura             28
Telangana           28
Tamil Nadu          28
Rajasthan           28
Punjab              28
Odisha              28
Maharashtra         28
Kerala              28
Madhya Pradesh      28
Jharkhand           28
Himachal Pradesh    28
Haryana             28
Gujarat             28
Delhi               28
Chhattisgarh        28
Bihar               28
NaN                 28
Meghalaya           27
```

```
Uttarakhand         27
Assam               26
Puducherry          26
Goa                 24
Jammu & Kashmir     21
Sikkim              17
Chandigarh          12
Name: Region, dtype: int64
```

```
df['Region'].isnull().sum()
```

```
28
```

```
#Get indices where there is nan
nan_index_Region = df[df['Region'].isnull()].index.tolist()
```

```
# Checking for null values
df['Date'].value_counts(dropna=False)
```

```
2019-10-31    55
2019-11-30    55
2019-05-31    54
2019-06-30    54
2019-07-31    54
2019-08-31    53
2019-12-31    53
2020-01-31    53
2020-02-29    53
2019-09-30    52
2020-03-31    52
2020-04-30    51
2020-05-31    51
2020-06-30    50
NaT           28
Name: Date, dtype: int64
```

```
# Checking for null values
nan_index_Date = df[df['Date'].isnull()].index.tolist()
```

Frequency

```
df['Frequency'] = df['Frequency'].str.replace(' Monthly', 'Monthly')

# Checking for null values
df['Frequency'].value_counts(dropna=False)
```

```
Monthly    740
NaN         28
Name: Frequency, dtype: int64
```

```
# Checking for null values
nan_index_Frequency = df[df['Frequency'].isnull()].index.tolist()
```

Estimated Unemployment Rate (%)

```
# Checking for null values
df['Estimated Unemployment Rate (%)'].value_counts(dropna=False)
```

```
NaN      28
0.00     11
3.31      4
5.35      3
3.66      3
         ..
13.70     1
4.03      1
40.59     1
3.69      1
9.86      1
Name: Estimated Unemployment Rate (%), Length: 625, dtype: int64
```

```
# Checking for null values
nan_index_Unemployment_Rate = df[df['Estimated Unemployment Rate (%)'].isnull()].index.tolist()
```

Estimated Employed

```
# Checking for null values
df['Estimated Employed'].value_counts(dropna=False)
```

```
NaN          28
247210.0      1
233029.0      1
241366.0      1
246596.0      1
             ..
6021921.0     1
6395022.0     1
6164215.0     1
6189471.0     1
9088931.0     1
Name: Estimated Employed, Length: 741, dtype: int64
```

```
# Checking for null values
nan_index_Employed = df[df['Estimated Employed'].isnull()].index.tolist()
```

Estimated Labour Participation Rate (%)

```
# Checking for null values
df['Estimated Labour Participation Rate (%)'].value_counts(dropna=False)
```

```
NaN      28
40.43     3
42.82     3
39.92     3
43.25     3
         ..
44.08     1
46.50     1
45.79     1
44.79     1
40.67     1
Name: Estimated Labour Participation Rate (%), Length: 627, dtype: int64
```

```
# Checking for null values
nan_index_Labour = df[df['Estimated Labour Participation Rate (%)'].isnull()].index.tolist()
```

Area

```
# Checking for null values
df['Area'].value_counts(dropna=False)
```

```
Urban    381
Rural    359
NaN       28
Name: Area, dtype: int64
```

```
# Checking for null values
nan_index_Area = df[df['Area'].isnull()].index.tolist()
```

Checking if all null rows are same?

```
Null = [nan_index_Region, nan_index_Date, nan_index_Frequency, nan_index_Unemployment_Rate, nan_index_Employed, nan_index_Labour, nan_index_A

#check if there is intersection between the indices
for x in range(0, len(Null)):
    for y in range(x+1, len(Null)):
```

```
        if not set(Null[x]).intersection(Null[y]):
            print(f'No intersection between {x} and {y}')
```

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

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Area	Month_int	Month_name
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
2	Andhra Pradesh	2019-07-31	Monthly	3.75	12086707.0	43.50	Rural	7	Jul
3	Andhra Pradesh	2019-08-31	Monthly	3.32	12285693.0	43.97	Rural	8	Aug
4	Andhra Pradesh	2019-09-30	Monthly	5.17	12256762.0	44.68	Rural	9	Sep
...	...	...	...	...	...	...	...	...	...
749	West Bengal	2020-02-29	Monthly	7.55	10871168.0	44.09	Urban	2	Feb
750	West Bengal	2020-03-31	Monthly	6.67	10806105.0	43.34	Urban	3	Mar
751	West Bengal	2020-04-30	Monthly	15.63	9299466.0	41.20	Urban	4	Apr
752	West Bengal	2020-05-31	Monthly	15.22	9240903.0	40.67	Urban	5	May
753	West Bengal	2020-06-30	Monthly	9.86	9088931.0	37.57	Urban	6	Jun

740 rows × 9 columns

Dependencies

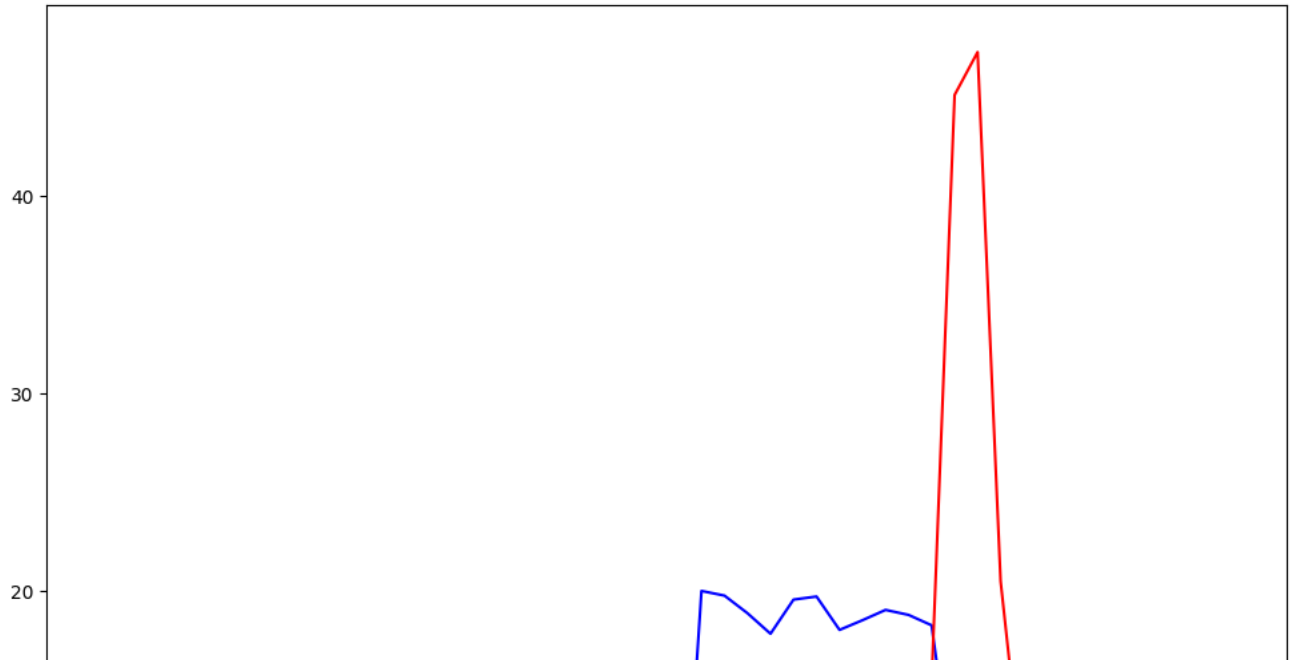
Employement vs UnEmployement

```
#checking relation using just first 50 rows to keep the plot readable
Employed = df['Estimated Employed'].iloc[:50]
Unemployed = df['Estimated Unemployment Rate (%)'].iloc[:50]

#normalize the employed in range 1 to 20
Employed = (Employed - Employed.min())/(Employed.max() - Employed.min()) * 19 + 1

plt.figure(figsize=(12,10))
plt.plot(Employed, label='Employed', color='blue')
plt.plot(Unemployed, label='Unemployed', color='red')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
#get correlation of df['Estimated Employed'] and  df['Estimated Unemployment Rate (%)']
print(df['Estimated Employed'].corr(df['Estimated Unemployment Rate (%)']))

'''This shows that the correlation between the two is too less hence dropping employement.'''
```

```
-0.22287639952214783
'This shows that the correlation between the two is too less hence dropping employement.'
```

```
df = df.drop(['Estimated Employed'], axis=1)
df
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Labour Participation Rate (%)	Area	Month_int	Month_name
0	Andhra Pradesh	2019-05-31	Monthly	3.65	43.24	Rural	5	May
1	Andhra Pradesh	2019-06-30	Monthly	3.05	42.05	Rural	6	Jun
2	Andhra Pradesh	2019-07-31	Monthly	3.75	43.50	Rural	7	Jul
3	Andhra Pradesh	2019-08-31	Monthly	3.32	43.97	Rural	8	Aug
4	Andhra Pradesh	2019-09-30	Monthly	5.17	44.68	Rural	9	Sep
...	...	...	...	...	...	...	...	...
749	West Bengal	2020-02-29	Monthly	7.55	44.09	Urban	2	Feb
750	West Bengal	2020-03-31	Monthly	6.67	43.34	Urban	3	Mar

Unemployement vs Dates

```
#group by region and area
group = df.groupby(['Region','Area']).agg({'Estimated Unemployment Rate (%)': 'mean'})
group
```

Region	Area	Estimated Unemployment Rate (%)
Andhra Pradesh	Rural	5.526429
	Urban	9.427857
Assam	Rural	4.490833
	Urban	8.088571
Bihar	Rural	16.770000
	Urban	21.066429
Chandigarh	Rural	NaN
	Urban	15.991667
Chhattisgarh	Rural	6.628571
	Urban	11.852143
Delhi	Rural	15.258571
	Urban	17.732143
Goa	Rural	8.390000
	Urban	10.158333
Gujarat	Rural	5.917143
	Urban	7.410714
Haryana	Rural	25.012857
	Urban	27.553571
Himachal Pradesh	Rural	15.504286
	Urban	21.576429
Jammu & Kashmir	Rural	14.951818
	Urban	17.549000
Jharkhand	Rural	15.221429
	Urban	25.948571
Karnataka	Rural	7.224286
	Urban	6.127857
Kerala	Rural	10.341429
	Urban	9.906429
Madhya Pradesh	Rural	5.653571
	Urban	9.159286
Maharashtra	Rural	6.810000
	Urban	8.305000
Meghalaya	Rural	2.475000
	Urban	7.301538
Odisha	Rural	6.612857
	Urban	4.702857
Puducherry	Rural	7.263333
	Urban	12.745000
Punjab	Rural	11.925000
	Urban	12.137143
Rajasthan	Rural	10.927143