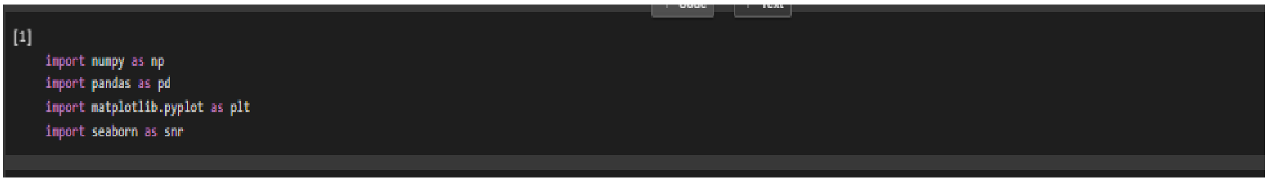
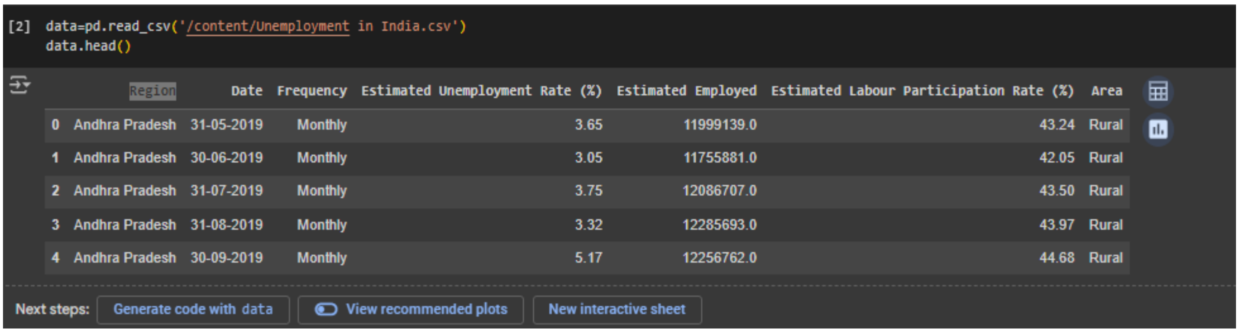
Project on Unemployment analysis using python

- **Aim:-**To create a Data science Project, where we will be predicting 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.
- Steps to be taken in the project is sub-divided into the following sections. These are:
  - Importing the libraries such as 'numpy', 'pandas etc.
  - Loading Dataset as a CSV file for training & testing the models.
  - Checking if still any null values or any other data types other than float and integers are present into the dataset or not.
- Steps of creating unemployment analysis s:-
- Importing numpy as np & pandas as pd for loading and reading the data-set & using matplotlib.pyplot and Seaborn for visualization of data.



```
[1]
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as snr
```

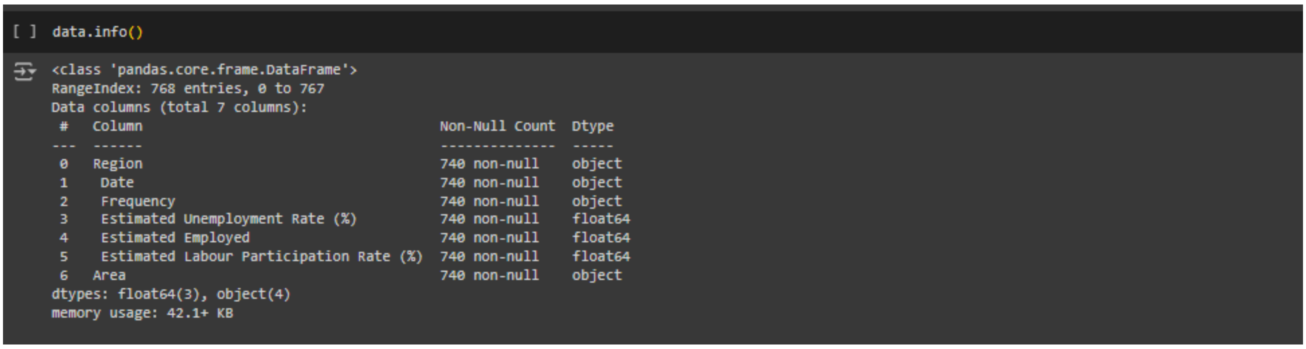
- Loading the csv-dataset in the variable name 'data Then viewing the data with data.head



```
[2] data=pd.read_csv('/content/Unemployment in India.csv')
    data.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

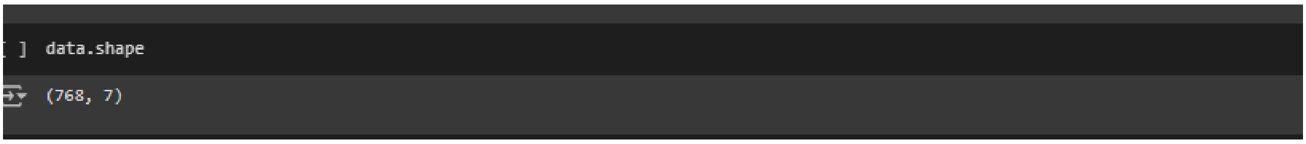
- Checking the data such as number of columns, rows and type of data(float,integer) with help of data_train.info()



```
[ ] data.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
```

We observe that the above data have integer, object and float.

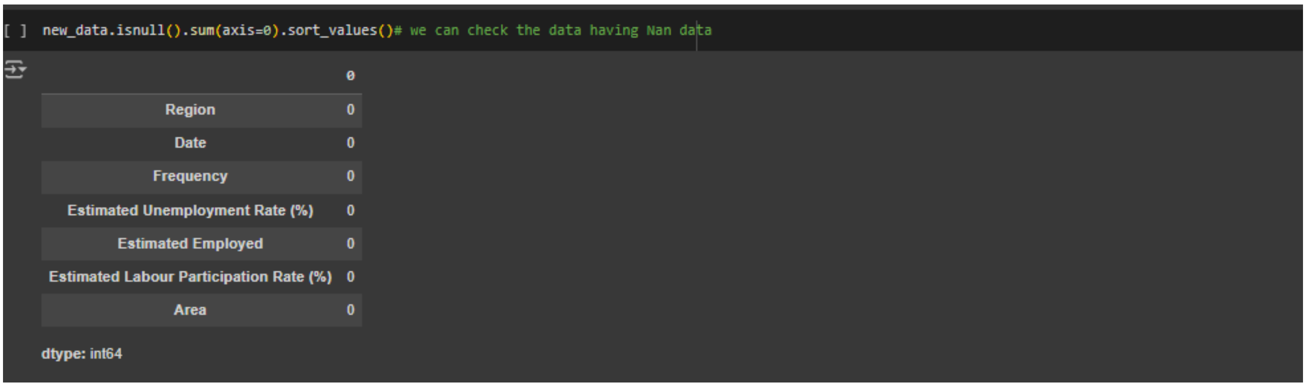


```
[ ] data.shape
```

```
(768, 7)
```

Train data have 768 Rows and 7 columns

- Now checking data have Nan value or not.



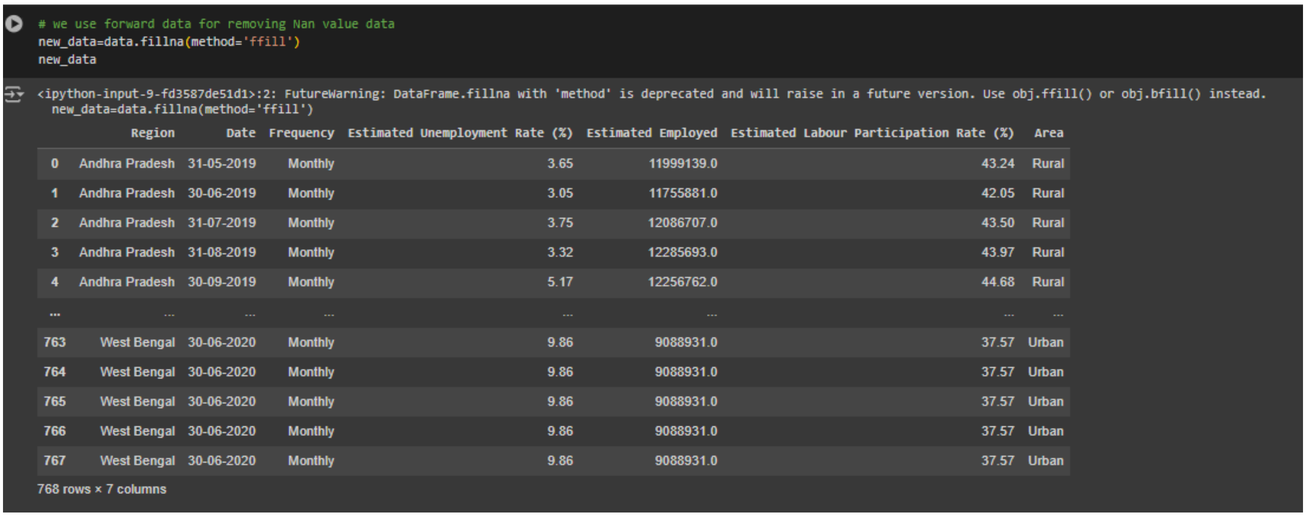
```
[ ] new_data.isnull().sum(axis=0).sort_values()# we can check the data having Nan data
```

	0
Region	0
Date	0
Frequency	0
Estimated Unemployment Rate (%)	0
Estimated Employed	0
Estimated Labour Participation Rate (%)	0
Area	0

dtype: int64

We observe that the above data have Nan value..we cleaning the Nan value before working on it.

- We remove the Nan value with help of fillna(method='ffill)



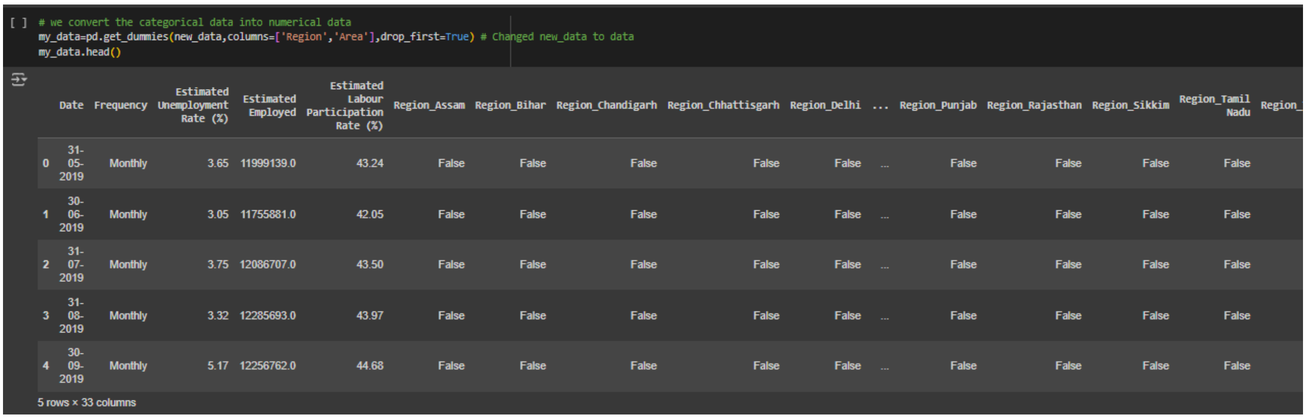
```
# we use forward data for removing Nan value data
new_data=data.fillna(method='ffill')
new_data
```

```
<ipython-input-9-fd3587de51d1>:2: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  new_data=data.fillna(method='ffill')
```

	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
...	...	...	...	...	...	...	...
763	West Bengal	30-06-2020	Monthly	9.86	9088931.0	37.57	Urban
764	West Bengal	30-06-2020	Monthly	9.86	9088931.0	37.57	Urban
765	West Bengal	30-06-2020	Monthly	9.86	9088931.0	37.57	Urban
766	West Bengal	30-06-2020	Monthly	9.86	9088931.0	37.57	Urban
767	West Bengal	30-06-2020	Monthly	9.86	9088931.0	37.57	Urban

768 rows × 7 columns

- Now ,Main focus convert the categorical data into Numerical data with help of one hot encoding method.



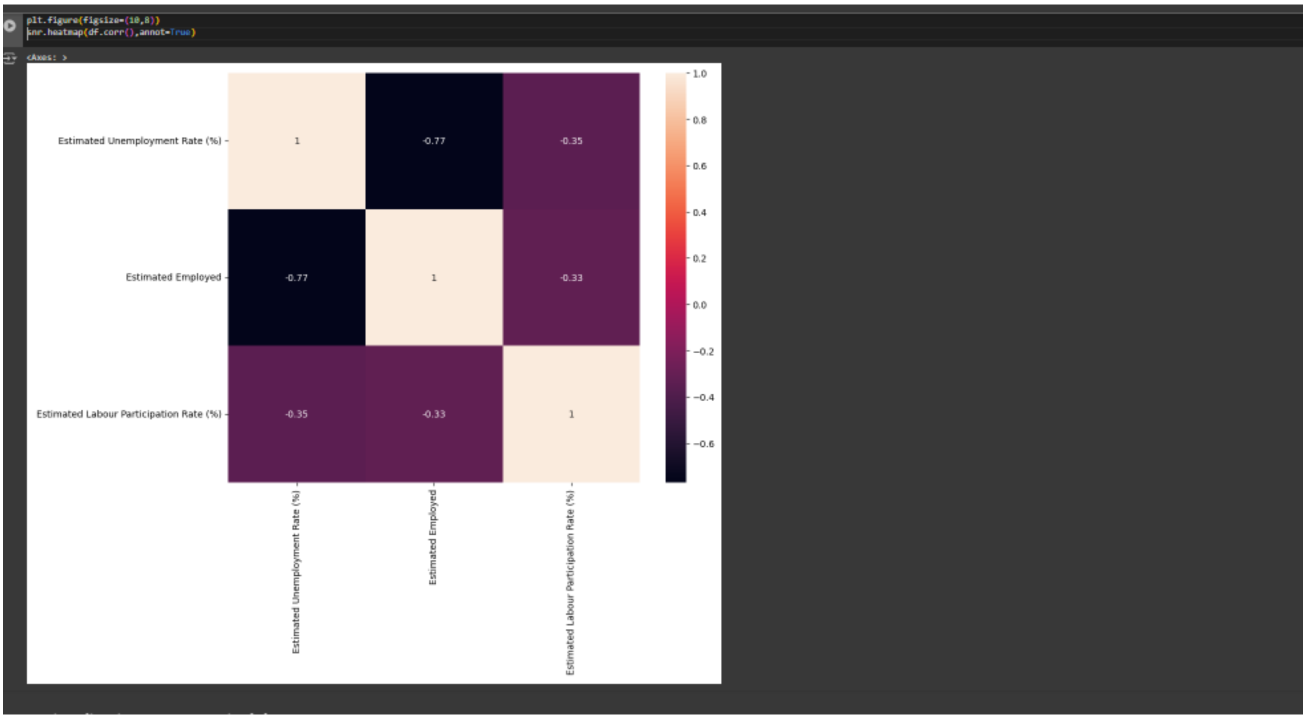
```
[ ] # we convert the categorical data into numerical data
    my_data=pd.get_dummies(new_data,columns=['Region','Area'],drop_first=True) # Changed new_data to data
    my_data.head()
```

	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region_Assam	Region_Bihar	Region_Chandigarh	Region_Chhattisgarh	Region_Delhi	...	Region_Punjab	Region_Rajasthan	Region_Sikkim	Region_Tamil Nadu	Region_
0	31-05-2019	Monthly	3.65	11999139.0	43.24	False	False	False	False	False	...	False	False	False	False	
1	30-06-2019	Monthly	3.05	11755881.0	42.05	False	False	False	False	False	...	False	False	False	False	
2	31-07-2019	Monthly	3.75	12086707.0	43.50	False	False	False	False	False	...	False	False	False	False	
3	31-08-2019	Monthly	3.32	12285693.0	43.97	False	False	False	False	False	...	False	False	False	False	
4	30-09-2019	Monthly	5.17	12256762.0	44.68	False	False	False	False	False	...	False	False	False	False	

5 rows × 33 columns

Finally we observe the data are fully cleaned.

- Now we check the data dependency.



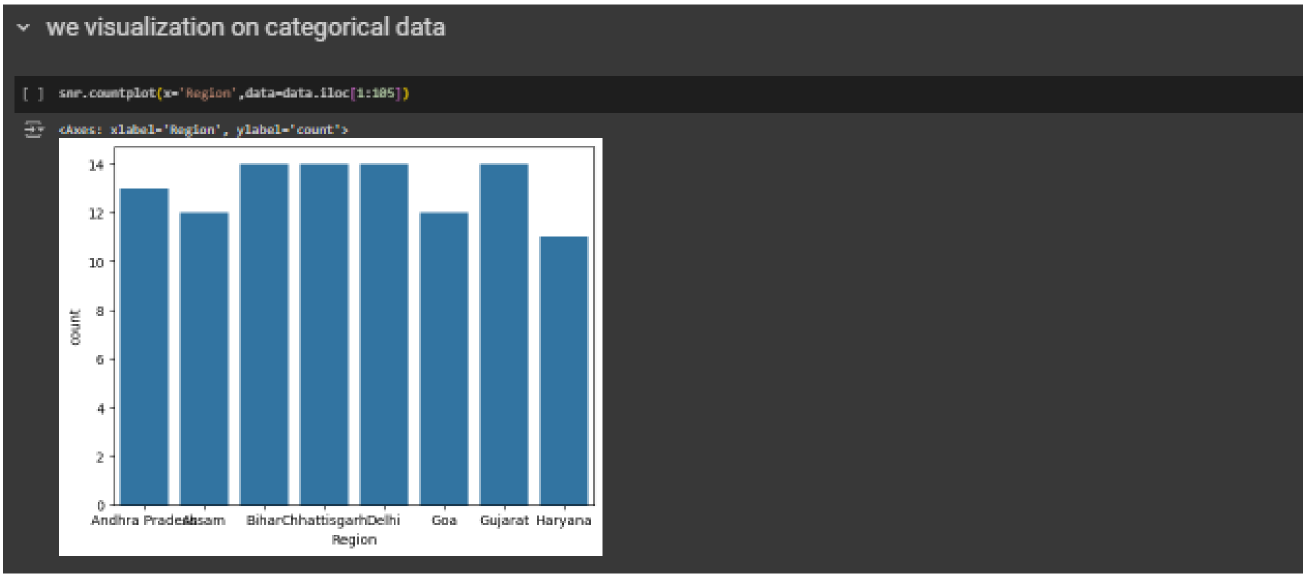
```
plt.figure(figsize=(10,8))
snr.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```

	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)
Estimated Unemployment Rate (%)	1	-0.77	-0.35
Estimated Employed	-0.77	1	-0.33
Estimated Labour Participation Rate (%)	-0.35	-0.33	1

We see that data dependent each other.

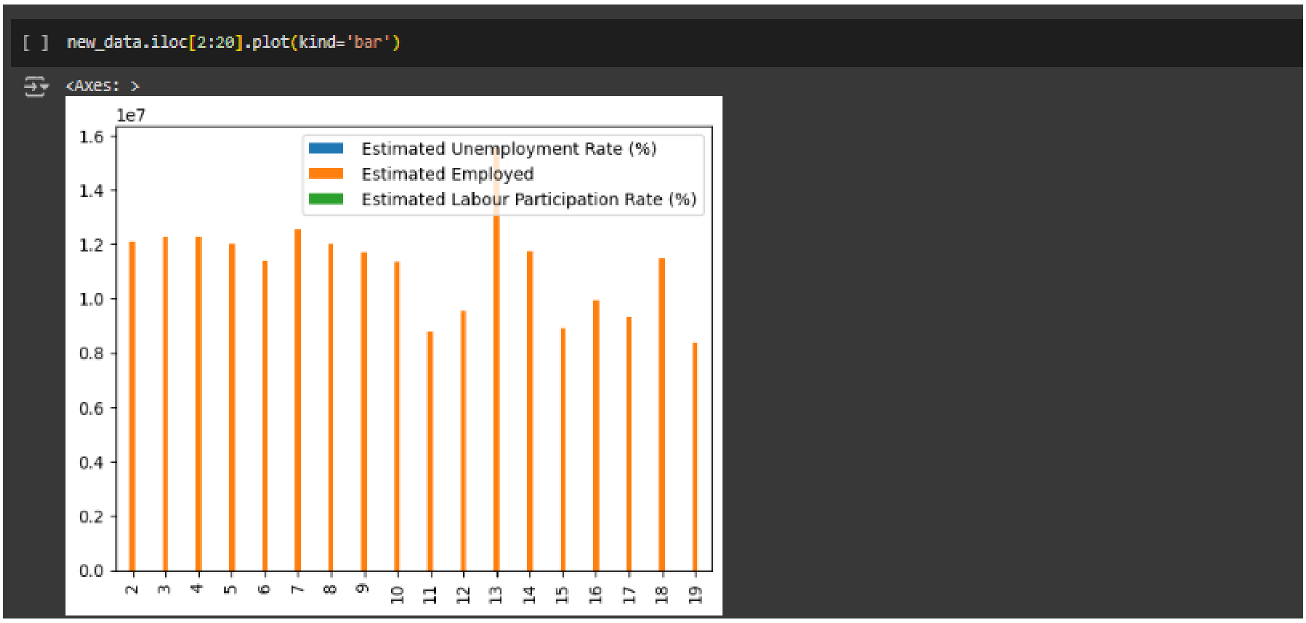
- Visualizing the Unemployment .



we visualization on categorical data

```
[ ] snr.countplot(x='Region',data=data.iloc[1:105])
```

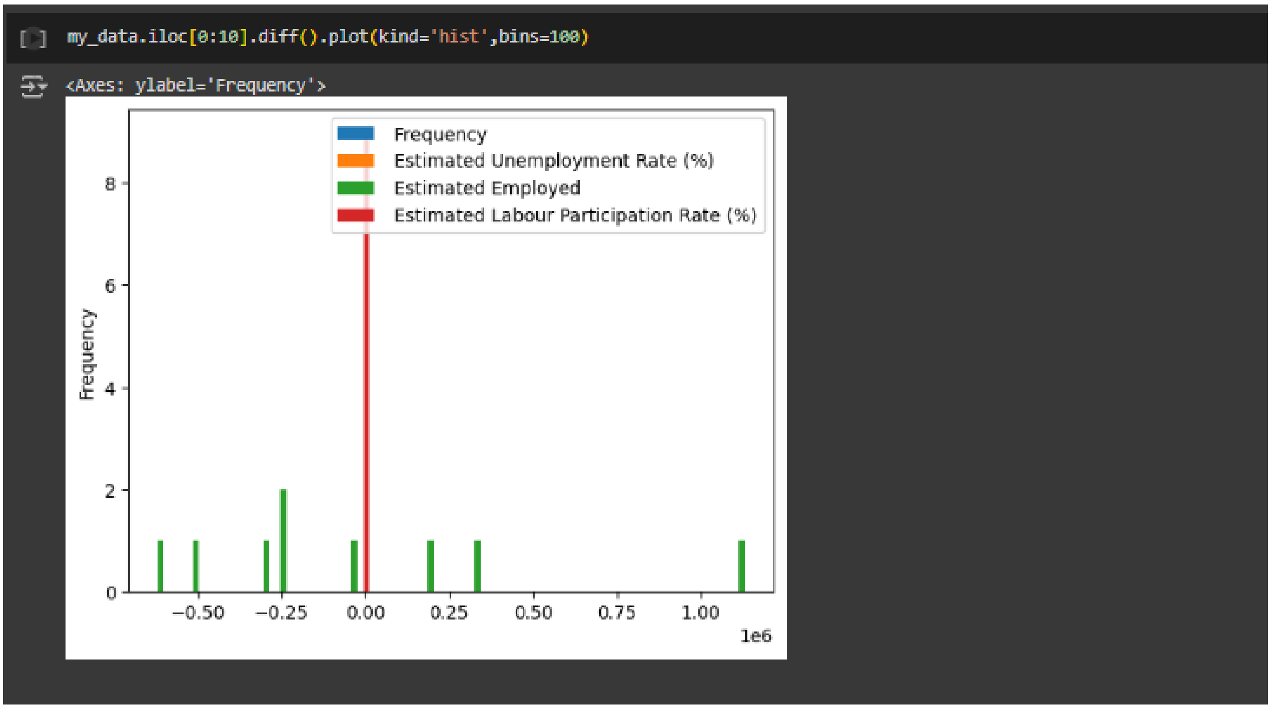
```
<Axes: xlabel='Region', ylabel='count'>
```



```
[ ] new_data.iloc[2:20].plot(kind='bar')
```

```
<Axes: >
```

As per Visualizing the above graph,



```
my_data.iloc[0:10].diff().plot(kind='hist',bins=100)
```

```
<Axes: ylabel='Frequency'>
```

So this is how you can analyze the unemployment rate by using the Python programming language. Unemployment is measured by the unemployment rate which is the number of people who are unemployed as a percentage of the total labour force.

Thank you