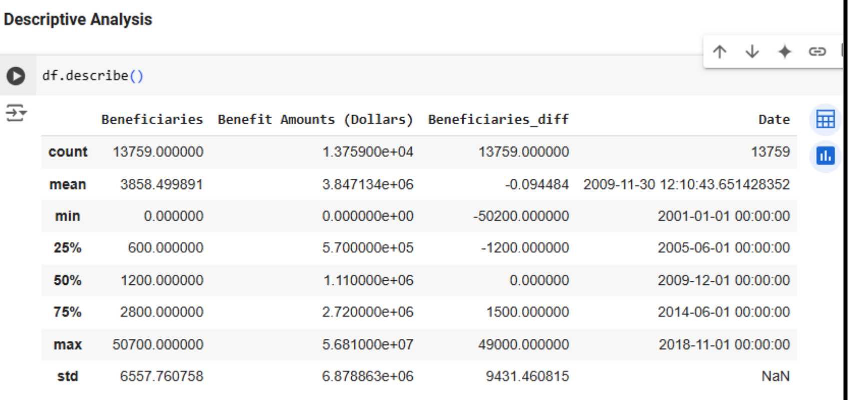
Data Collection and Preprocessing Phase

Date	27 NOVEMBER 2024
Team ID	FACULTY
Project Title	Unemployed Insurance Beneficiary Forecasting.
Maximum Marks	6 Marks

Preprocessing Template

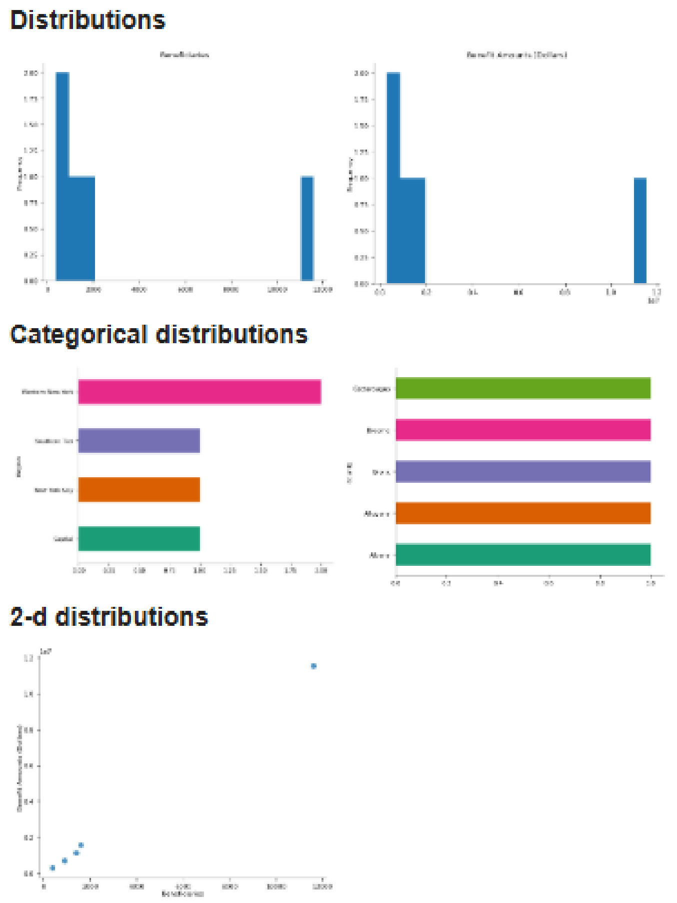
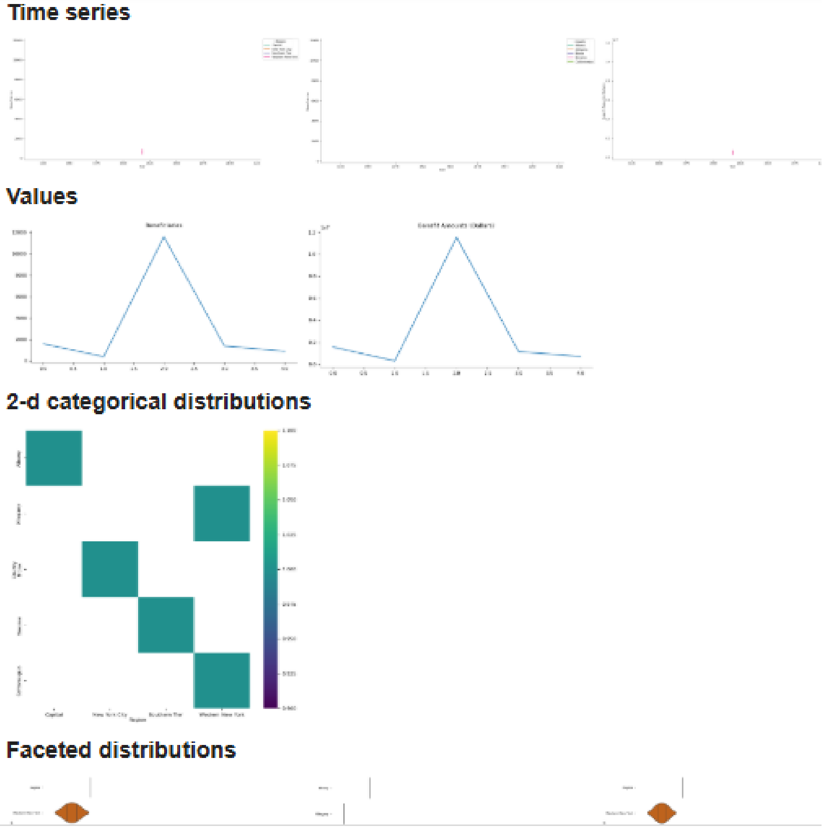
The images will be preprocessed by resizing, normalizing, augmenting,Dataset variables will be statistically analyzed to identify patterns and outliers, with Python,employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	Dimension: 13760 rows × 6 columns Descriptive statistics: 

Descriptive Analysis

```
df.describe()
```

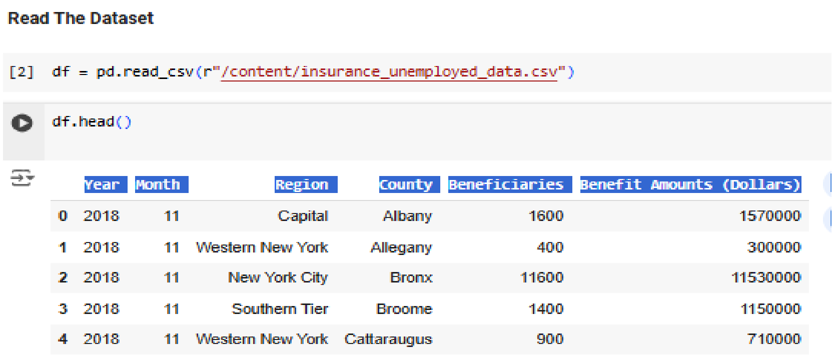
```
       Beneficiaries  Benefit Amounts (Dollars)  Beneficiaries_diff                           Date
count   13759.000000               1.375900e+04        13759.000000                          13759
mean     3858.499891               3.847134e+06           -0.094484  2009-11-30 12:10:43.651428352
min         0.000000               0.000000e+00       -50200.000000            2001-01-01 00:00:00
25%       600.000000               5.700000e+05        -1200.000000            2005-06-01 00:00:00
50%      1200.000000               1.110000e+06            0.000000            2009-12-01 00:00:00
75%      2800.000000               2.720000e+06         1500.000000            2014-06-01 00:00:00
max     50700.000000               5.681000e+07        49000.000000            2018-11-01 00:00:00
std      6557.760758               6.878863e+06         9431.460815                            NaN
```

Distributions	 Distributions Categorical distributions 2-d distributions Beneficiaries Benefit Amounts (Dollars) Frequency Western New York Southern Tier New York City Capital Region
	 Time series Values 2-d categorical distributions Faceted distributions Beneficiaries Benefit Amounts (Dollars) Capital New York City Southern Tier Western New York Region

Outliers and Anomalies	--

Data Preprocessing Code Screenshots

Loading Data



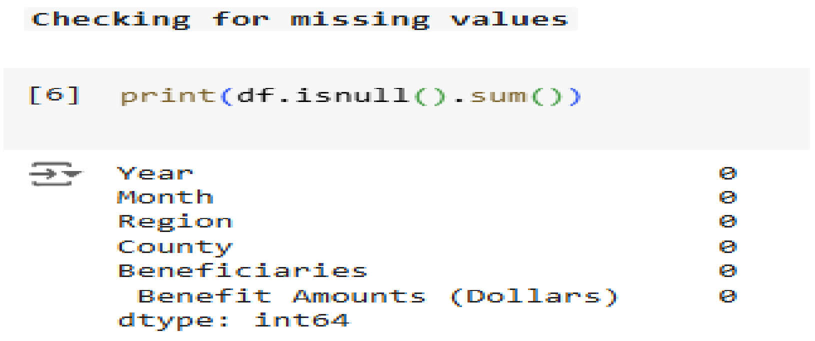
Read The Dataset

```
[2] df = pd.read_csv(r"/content/insurance_unemployed_data.csv")
```

```
df.head()
```

	Year	Month	Region	County	Beneficiaries	Benefit Amounts (Dollars)
0	2018	11	Capital	Albany	1600	1570000
1	2018	11	Western New York	Allegany	400	300000
2	2018	11	New York City	Bronx	11600	11530000
3	2018	11	Southern Tier	Broome	1400	1150000
4	2018	11	Western New York	Cattaraugus	900	710000

Handling Missing Data

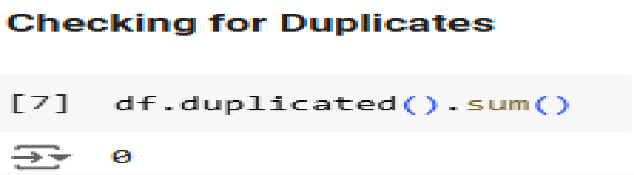


Checking for missing values

```
[6] print(df.isnull().sum())
```

```
Year                          0
Month                         0
Region                        0
County                        0
Beneficiaries                 0
 Benefit Amounts (Dollars)    0
dtype: int64
```

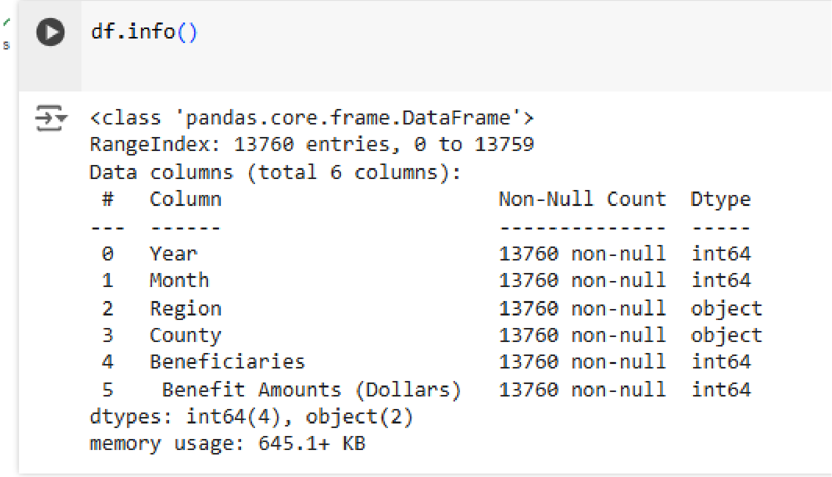
Checking Duplicates



Checking for Duplicates

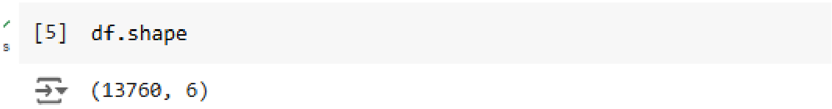
```
[7] df.duplicated().sum()
```

```
0
```



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13760 entries, 0 to 13759
Data columns (total 6 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   Year                       13760 non-null  int64 
 1   Month                      13760 non-null  int64 
 2   Region                     13760 non-null  object
 3   County                     13760 non-null  object
 4   Beneficiaries              13760 non-null  int64 
 5    Benefit Amounts (Dollars)  13760 non-null  int64 
dtypes: int64(4), object(2)
memory usage: 645.1+ KB
```

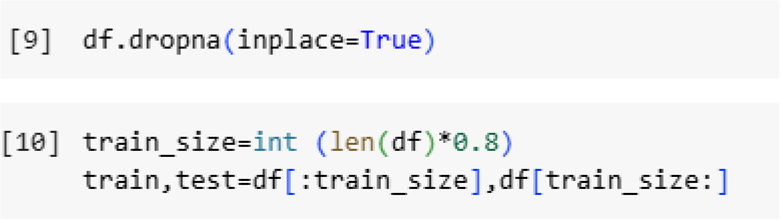


```
[5] df.shape
```

```
(13760, 6)
```

Feature Engineering

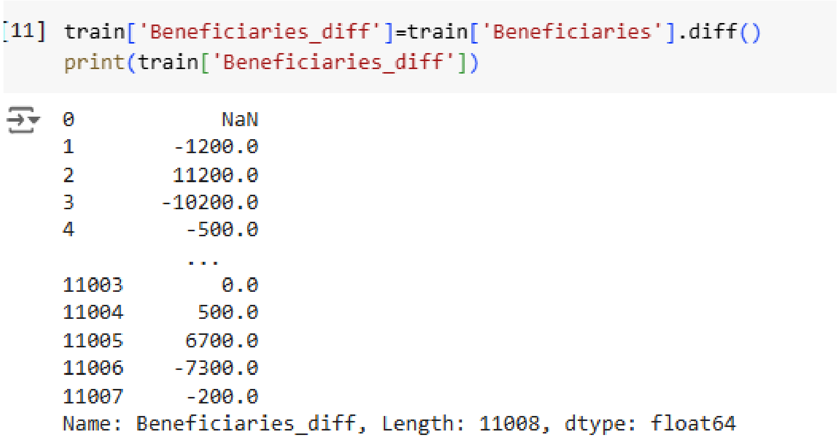
Splitting Dataset into Train and Test Sets



```
[9]  df.dropna(inplace=True)
```

```
[10] train_size=int (len(df)*0.8)
     train,test=df[:train_size],df[train_size:]
```

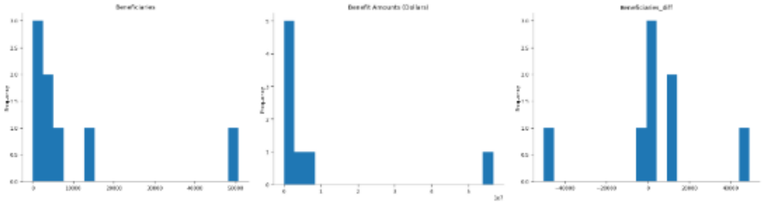
create differenced column

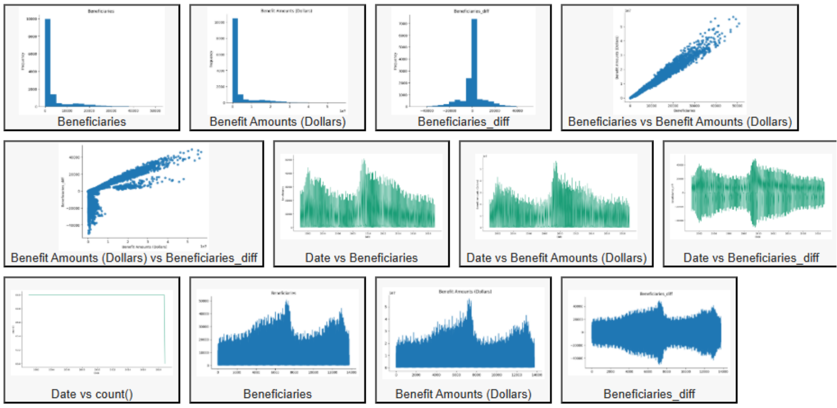


```
[11] train['Beneficiaries_diff']=train['Beneficiaries'].diff()
     print(train['Beneficiaries_diff'])
```

```
0             NaN
1         -1200.0
2         11200.0
3        -10200.0
4          -500.0
           ...
11003         0.0
11004       500.0
11005      6700.0
11006     -7300.0
11007      -200.0
Name: Beneficiaries_diff, Length: 11008, dtype: float64
```

Distributions


Beneficiaries
Frequency
Benefit Amounts (Dollars)
1e7
Beneficiaries_diff


Beneficiaries
Benefit Amounts (Dollars)
Beneficiaries_diff
Beneficiaries vs Benefit Amounts (Dollars)
Benefit Amounts (Dollars) vs Beneficiaries_diff
Date vs Beneficiaries
Date vs Benefit Amounts (Dollars)
Date vs Beneficiaries_diff
Date vs count()
Beneficiaries
Benefit Amounts (Dollars)
Beneficiaries_diff