Question 1

[CM1]Data Pre-Processing and Preparation

1.

Removing comma and converting Resident Population 2020 Census,Population Density 2020 Census columns to numeric type

```
cols=['Resident Population 2020 Census','Population Density 2020 Census']
original_data[cols] = original_data[cols].replace(',', '',regex=True)
original_data['Resident Population 2020 Census'] = pd.to_numeric(original_data['Resident Population 2020 Census'])
original_data['Population Density 2020 Census'] = pd.to_numeric(original_data['Population Density 2020 Census'])
original_data
```

2. Normalization - Z-score Normalization is applied on the data as:

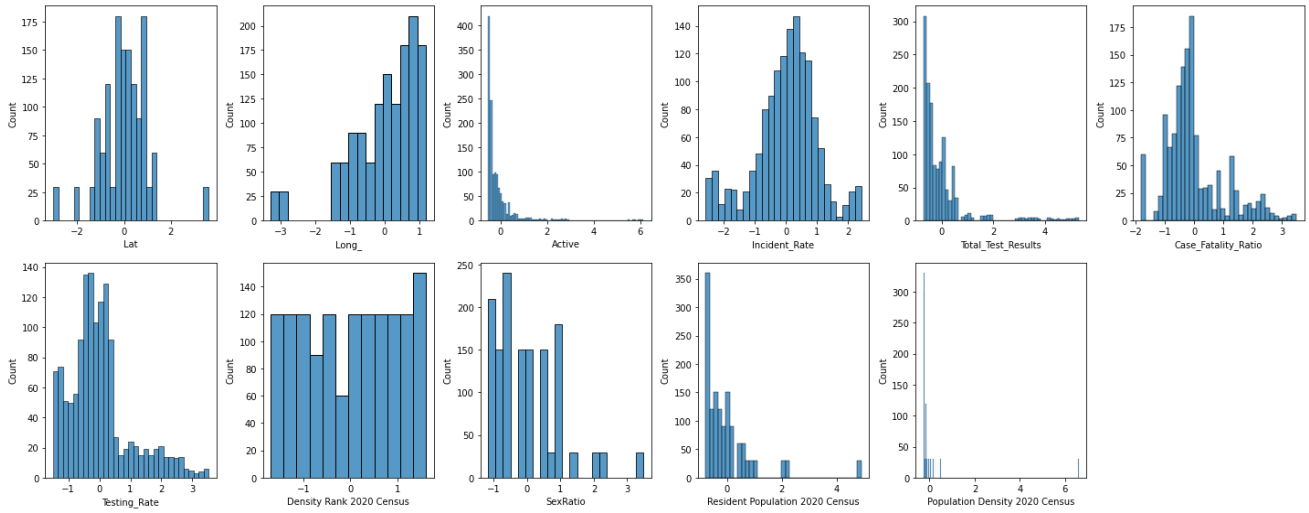
- It takes the points roughly in the same scale on both features and handles the squashing.
- Handles the outliers more effectively as compared to other normalization methods.

Z-score Normalization

```
[239] from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
      Original_data_copy.iloc[:, 3:14] = sc.fit_transform(Original_data_copy.iloc[:, 3:14])
      Original_data_copy
```

	Lat	Long_	Active	Incident_Rate	Total_Test_Results	Case_Fatality_Ratio	Testing_Rate	Resident Population 2020 Census	Population Density 2020 Census	Density Rank 2020 Census	SexRatio
	-1.178670	0.304586	-0.200701	0.144028	-0.483568	-0.476902	-1.302217	-0.128626	-0.217091	0.118788	-1.168679
	3.608919	-3.033032	-0.449116	-0.290315	-0.569578	-1.800519	2.071905	-0.754447	-0.276853	1.614954	3.492526
	-0.946051	-0.945268	0.389194	0.088543	-0.007494	0.074416	-1.269412	0.181627	-0.239250	0.509092	0.385056
	-0.741727	0.025825	-0.482520	0.202252	-0.456622	-0.031311	-0.559356	-0.422184	-0.242302	0.574143	-0.547185
	-0.552794	-1.365662	4.276989	-0.502599	4.023547	-0.792949	-0.177608	4.905194	-0.122779	-0.922023	0.385056
	...	...	...	...	...	...	...	...	...	...	...
	0.753949	1.027710	-0.524201	-2.293948	-0.625809	-0.270412	1.274271	-0.767619	-0.235038	0.378991	-0.236438
	-0.280378	0.749535	0.400942	-0.559558	-0.005286	-0.534982	-0.745315	0.397468	-0.144206	-0.726871	-0.236438
	-0.161416	0.607653	-0.488254	-0.195930	-0.475825	0.062389	0.411281	-0.599800	-0.232108	0.248890	0.074309

Histograms are plotted for the numeric features after normalization to visualize the distributions



3. Outliers: Boxplots are used on the numeric features to detect outliers

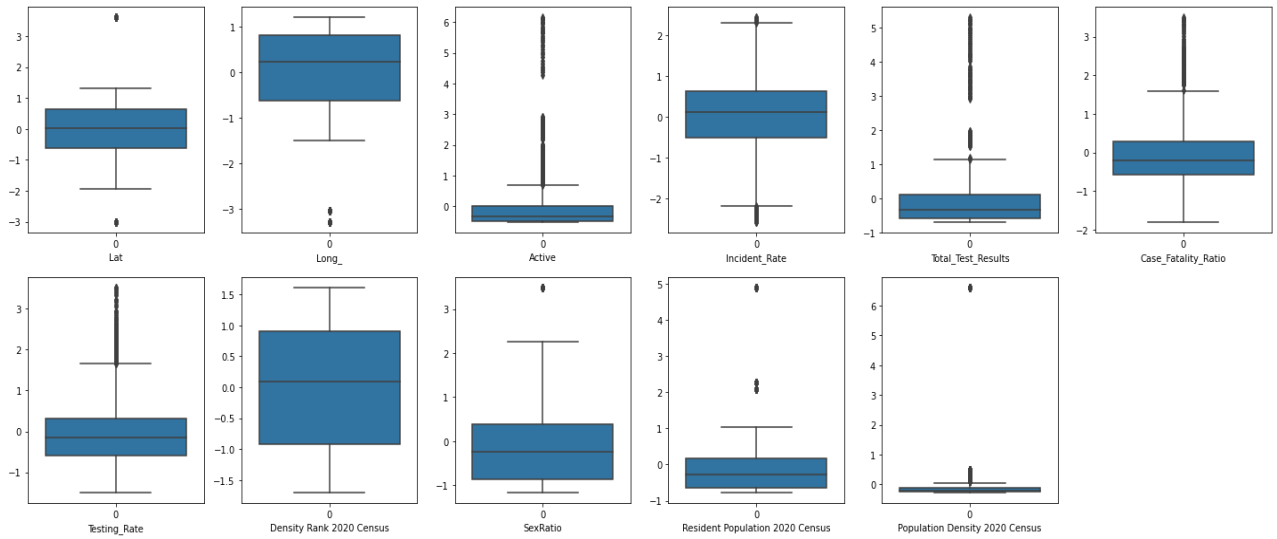
Outlier Detection-Boxplot

```
a = 2  # number of rows
b = 6  # number of columns
c = 1  # initialize plot counter

fig=plt.figure(figsize=(20,8))

for col in plot_cols:
    plt.subplot(a, b, c)
    plt.tight_layout()
    sns.boxplot(data=Original_data_copy[col])
    plt.xlabel(col)
    c = c + 1
plt.show()
```

From the above method, we can observe significant outliers in Active, Testing_Rate and Case_Fatality _Ratio and other columns below:



From the histogram, we can observe most of the features follow normal distribution. Hence z-score is used to remove outliers from these features.

The z-score is the number of standard deviations a data point deviates from the mean. But, in more technical terms, it's a measure of how many standard deviations a raw score is below or above the population mean. Z-scores range from -3 standard deviations (which corresponds to the extreme left of the normal distribution curve) to +3 standard deviations (which corresponds to the extreme right of the normal distribution curve). Data points that fall outside of these ranges are discarded.

Since there are lot of outliers in columns line Active, Case_fatality_ratio using IQR to remove outliers leads to significant data loss. Hence, z-score was chosen instead.

Outlier Removal using z-score

```
threshold = 3

for col in plot_cols:
  Original_data_copy['outliers'] = np.where((Original_data_copy[col] - threshold > 0), True, np.where(Original_data_copy[col] + threshold < 0, True, False))
  Original_data_copy.drop(Original_data_copy[Original_data_copy['outliers'] == True].index,inplace=True)

Original_data_copy.shape
```

```
(1209, 18)
```

Through this method, the shape of our data reduces from 1308 to 1209 after outlier removal.

- State and State ID columns convey the same information
- State is already available in encoded format in State ID
- Hence State column is dropped and State ID and Day columns are retained.