# Stationarity Test In a Time Series Dataset

**Aim:**
Implement programs to check stationary of a time series data

**Algorithm:**

1. Import Necessary Libraries:

   Load essential Python libraries: pandas for data manipulation, matplotlib.pyplot for plotting, adfuller from statsmodels.tsa.stattools for the ADF test, and files and io for file handling in Colab.

2. Upload the CSV File:

   Use files.upload() to prompt a file selection dialog in Colab, allowing you to upload your dataset (e.g., 'Electric_Production.csv').

3. Load the Dataset:

   Read the uploaded CSV file into a pandas DataFrame using pd.read_csv().

4. Inspect Column Names:

   Print the column names of the DataFrame to verify they are parsed correctly. This helps ensure that the subsequent steps reference the correct column names.

5. Convert Date Column to Datetime and Set as Index:

   Convert the 'DATE' column to datetime format using pd.to_datetime(). And Set the 'DATE' column as the index of the DataFrame with set_index()

6. Plot the Time Series Data:

   Create a line plot of the time series data using plt.plot(). Label the axes and provide a title to the plot for clarity.

7. Perform the Augmented Dickey-Fuller Test:

   Apply the ADF test to the time series data using adfuller().Extract and print the ADF statistic, p-value, and critical values from the test results.

8. Interpret the Test Results:

Compare the p-value to a significance level (commonly 0.05):

If the p-value is less than 0.05, reject the null hypothesis, indicating that the time series is likely stationary.

If the p-value is greater than or equal to 0.05, fail to reject the null hypothesis, suggesting that the time series is likely non-stationary.

**Program Code:**

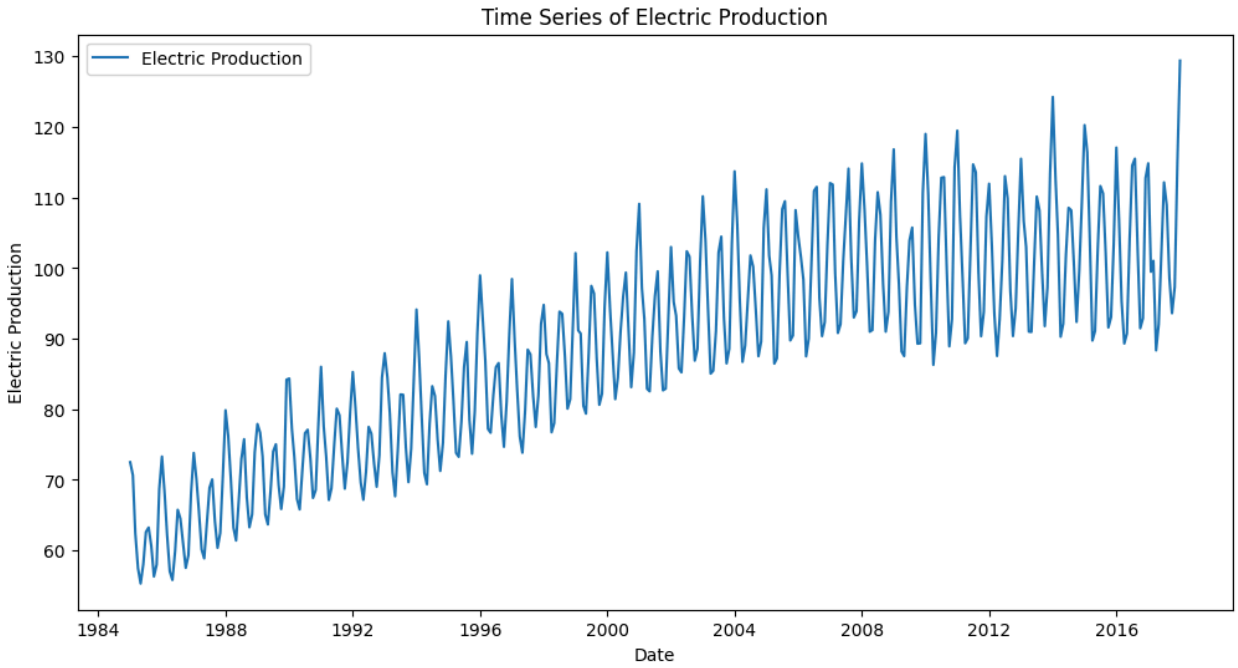
```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from google.colab import files
import io

uploaded = files.upload()

data = pd.read_csv(io.BytesIO(uploaded['Electric_Production.csv']))

data['DATE'] = pd.to_datetime(data['DATE'])
data.set_index('DATE', inplace=True)

plt.figure(figsize=(12, 6))
plt.plot(data.index, data['IPG2211A2N'], label='Electric Production')
plt.xlabel('Date')
plt.ylabel('Electric Production')
plt.title('Time Series of Electric Production')
plt.legend()
plt.show()
```




```
result = adfuller(data['IPG2211A2N'].dropna())
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

```
ADF Statistic: -2.256990
p-value: 0.186215
Critical Values:
        1%: -3.448
        5%: -2.869
        10%: -2.571
```

```
if result[1] < 0.05:
    print("The time series is likely stationary (reject H0).")
else:
    print("The time series is likely non-stationary (fail to reject H0).")
```

```
The time series is likely non-stationary (fail to reject H0).
```

**RESULTS:**

The program has been created and implemented successfully for time series analysis for stationarity check.