# **EX: 4 IMPLEMENT PROGRAM TO CHECK**

# **DATE: STATIONARY OF TIME SERIES DATA**

**AIM:**

To analyze the stationarity of a time series dataset using the **Augmented Dickey-Fuller (ADF) test** and visualize rolling statistics for trend detection.

# **ALGORITHM:**

# Load Data – Import the dataset and convert the Month column to datetime format.

# Preprocess Data – Set the Month column as the index to represent time series data.

# Compute Rolling Statistics – Calculate rolling mean and rolling standard deviation to observe trends.

# Perform ADF Test – Apply the Augmented Dickey-Fuller (ADF) **test** to check stationarity.

# Interpret Results – Analyze the test statistic and p-value to determine if the data is stationary.

# Apply Differencing (if needed) – If the series is non-stationary, apply first-order differencing.

# Re-evaluate Stationarity – Perform the ADF test again on the differenced series to confirm stationarity.

# **PROGRAM:**

# # Install necessary libraries (if needed)

# !pip install pandas numpy matplotlib statsmodels --quiet

# 

# # Import required libraries

# import numpy as np

# import pandas as pd

# import matplotlib.pyplot as plt

# import statsmodels.api as sm

# from statsmodels.tsa.stattools import adfuller

# 

# # Load the airline-passenger dataset

# file\_path = "/content/airline-passengers.csv" # File uploaded in Google Colab

# df = pd.read\_csv(file\_path)

# 

# # Convert the 'Month' column to datetime format

# df['Month'] = pd.to\_datetime(df['Month'])

# df.set\_index('Month', inplace=True)

# 

# # Extract time series data

# time\_series = df['Passengers']

# 

# # Function to check stationarity using ADF test

# def check\_stationarity(timeseries):

# # Rolling statistics

# rolling\_mean = timeseries.rolling(window=12).mean()

# rolling\_std = timeseries.rolling(window=12).std()

# 

# # Plot original data with rolling mean and standard deviation

# plt.figure(figsize=(12, 6))

# plt.plot(timeseries, label='Original Data', color='blue')

# plt.plot(rolling\_mean, label='Rolling Mean', color='red')

# plt.plot(rolling\_std, label='Rolling Std Dev', color='green')

# plt.legend()

# plt.title('Rolling Mean & Standard Deviation')

# plt.show()

# 

**OUTPUT:**

**Checking Stationarity of Original Data:**

**Results of Augmented Dickey-Fuller Test:**

**Test Statistic 0.815369**

**p-value 0.991880**

**#Lags Used 13.000000**

**Number of Observations Used 130.000000**

**Critical Value (1%) -3.481682**

**Critical Value (5%) -2.884042**

**Critical Value (10%) -2.578770**

**dtype: float64**

**The data is likely non-stationary (p-value >= 0.05)**

**# Augmented Dickey-Fuller test**

**print("Results of Augmented Dickey-Fuller Test:")**

**adf\_test = adfuller(timeseries, autolag='AIC')**

**adf\_results = pd.Series(adf\_test[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number of Observations Used'])**

**for key, value in adf\_test[4].items():**

**adf\_results[f'Critical Value ({key})'] = value**

**print(adf\_results)**

**# Conclusion**

**if adf\_test[1] <= 0.05:**

**print("\n✅ The data is likely stationary (p-value < 0.05)")**

**else:**

**print("\n❌ The data is likely non-stationary (p-value >= 0.05)")**

**# Check stationarity of the original data**

**print("\n📊 Checking Stationarity of Original Data:")**

**check\_stationarity(time\_series)**

**# Apply first-order differencing if non-stationary**

**diff\_series = time\_series.diff().dropna()**

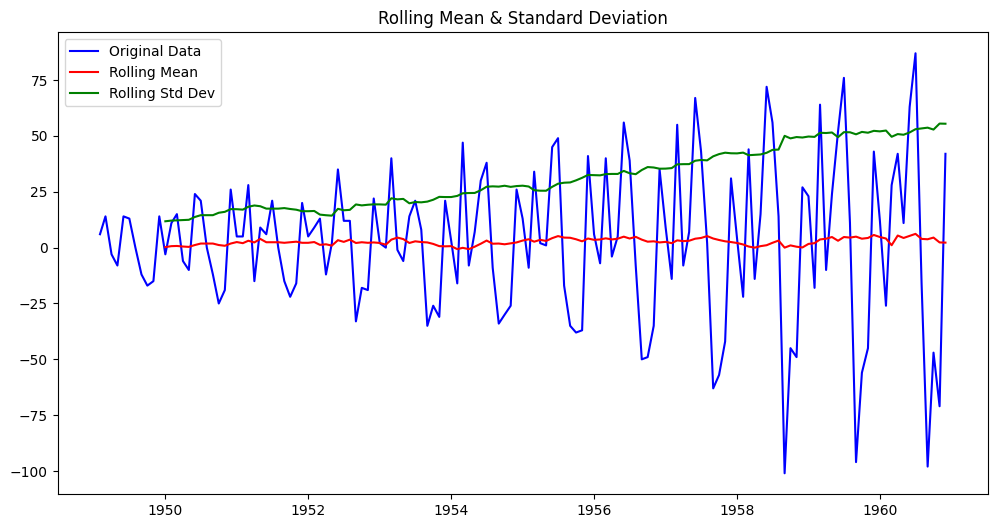
**# Check stationarity after differencing**

**print("\n📊 Checking Stationarity After First Differencing:")**

**check\_stationarity(diff\_series)**

**OUTPUT:**

**Checking Stationarity After First Differencing:**

****

**Results of Augmented Dickey-Fuller Test:**

**Test Statistic -2.829267**

**p-value 0.054213**

**#Lags Used 12.000000**

**Number of Observations Used 130.000000**

**Critical Value (1%) -3.481682**

**Critical Value (5%) -2.884042**

**Critical Value (10%) -2.578770**

**dtype: float64**

**The data is likely non-stationary (p-value >= 0.05)**

**RESULT:**

Thus, this program analyzes the stationarity of a time series using the Augmented Dickey-Fuller (ADF) test.