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

**EX:No.3**

**DATE:28/02/25**

**AIM:**

To check the stationarity of a time series using rolling statistics and the Augmented Dickey-Fuller (ADF) test.

**ALGORITHM:**

1. Load the Data – Read the time series dataset and set the 'Date' column as the index.
2. Extract Time Series – Select the 'Close' price or relevant column for analysis.
3. Compute Rolling Statistics – Calculate the rolling mean and standard deviation.
4. Plot Rolling Statistics – Visualize changes in mean and standard deviation over time.
5. Perform ADF Test – Apply the Augmented Dickey-Fuller (ADF) test to check stationarity.
6. Analyze ADF Results – Compare the p-value with 0.05 significance level.
7. Interpret Outcome – If p ≤ 0.05, the series is stationary; otherwise, it's non-stationary.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.tsa.stattools import adfuller

# Load the dataset

file\_path = 'AAPL.csv' # Replace with your dataset path

data = pd.read\_csv(file\_path)

# Convert 'Date' to datetime and set it as index

data['Date'] = pd.to\_datetime(data['Date'])

data.set\_index('Date', inplace=True)

# Selecting only the 'Close' price for analysis

ts = data['Close']

# Function to check stationarity

def check\_stationarity(timeseries):

# Rolling statistics

rolling\_mean = timeseries.rolling(window=30).mean()

rolling\_std = timeseries.rolling(window=30).std()

# Plot rolling statistics

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

plt.plot(timeseries, label='Original Time Series', color='blue', alpha=0.5)

plt.plot(rolling\_mean, label='Rolling Mean (30-day)', color='red', linewidth=2)

plt.plot(rolling\_std, label='Rolling Std Dev (30-day)', color='green', linewidth=2)

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

plt.legend()

plt.show()

# Augmented Dickey-Fuller Test

print("\nAugmented Dickey-Fuller Test Results:")

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

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

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

results[f'Critical Value ({key})'] = value

print(results)

# Conclusion

if adf\_test[1] <= 0.05:

print("\n The time series is stationary (p-value <= 0.05).")

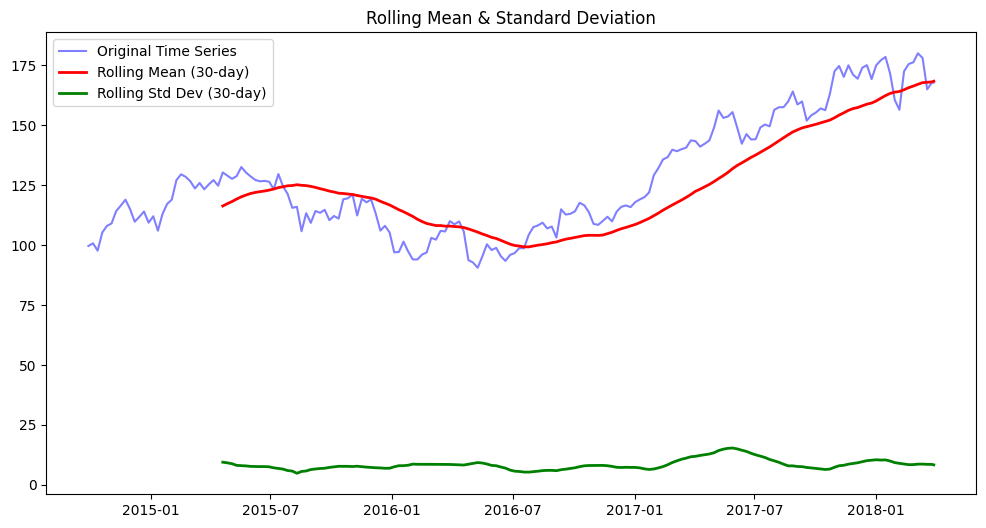
else:

print("\n The time series is NOT stationary (p-value > 0.05).")

# Run stationarity check

check\_stationarity(ts)

**OUTPUT:**



Augmented Dickey-Fuller Test Results:

ADF Test Statistic -0.868202

p-value 0.798264

# Lags Used 0.000000

# Observations Used 183.000000

Critical Value (1%) -3.466598

Critical Value (5%) -2.877467

Critical Value (10%) -2.575260

dtype: float64

The time series is NOT stationary (p-value > 0.05).

**RESULT:**

Thus the program has been completed and verified successfully.