EX:No.3 DATE:1/02/25	Implement programs to check stationarity of a time series data.

AIM:

To Implement programs to check stationarity of a time series data.

OBJECTIVE:

To analyze whether the air pollution time-series data is stationary using statistical tests and visualizations.

BACKGROUND:

- A stationary time series has a constant mean, variance, and no seasonality.
- Stationarity is important for forecasting and modeling.
- Non-stationary data needs transformations like differencing.
- Statistical tests like ADF (Augmented Dickey-Fuller) test help detect stationarity. Visual methods like rolling statistics help identify trends and variance changes.

SCOPE OF THE PROGRAM:

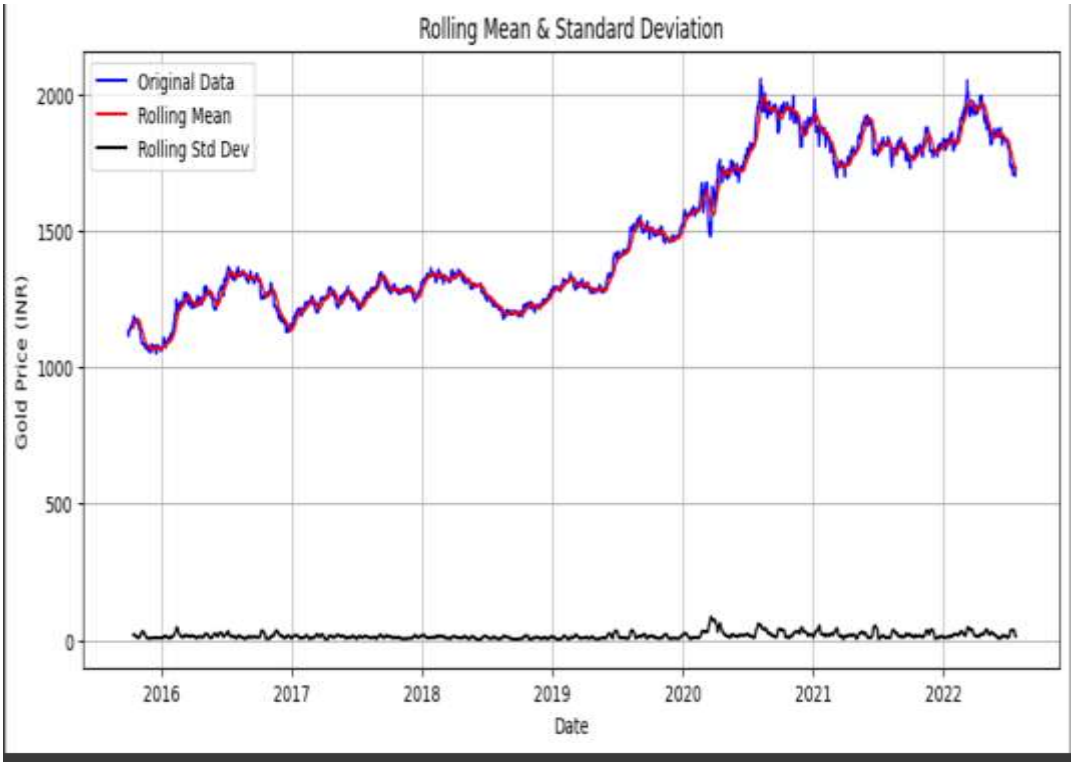
- Load and clean air pollution time-series data.
- Check for missing values and handle them.
- Use rolling mean and standard deviation to check stationarity.
- Apply Augmented Dickey-Fuller (ADF) test for statistical confirmation.
- Apply differencing if the data is non-stationary.

CODE:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.stattools import adfuller

df = pd.read_csv("/content/gold_data.csv")

print("Column names in dataset:", df.columns)

date_col = "Date"  # Change if needed
gold_price_col = "Price"  # Update based on actual column name

df[date_col] = pd.to_datetime(df[date_col], errors='coerce')
df.set_index(date_col, inplace=True)

df.dropna(subset=[gold_price_col], inplace=True)
```

```
def check_stationarity(timeseries):
    #
    rolling_mean = timeseries.rolling(window=12).mean()
    rolling_std = timeseries.rolling(window=12).std()

    # Plot rolling statistics
    plt.figure(figsize=(12, 5))
    plt.plot(timeseries, color='blue', label="Original Data")
    plt.plot(rolling_mean, color='red', label="Rolling Mean")
    plt.plot(rolling_std, color='black', label="Rolling Std Dev")
    plt.xlabel("Date")
    plt.ylabel("Gold Price (INR)")
    plt.title("Rolling Mean & Standard Deviation")
    plt.legend()
    plt.grid()
    plt.show()

    # Perform Augmented Dickey-Fuller test
    print("\nDickey-Fuller Test Results:")
    adf_test = adfuller(timeseries, autolag='AIC')
     adf_results = pd.Series(adf_test[:4], index=["Test Statistic", "p-value", "# Lags
Used", "# Observations Used"])
    for key, value in adf_test[4].items():
        adf_results[f'Critical Value ({key})'] = value
    print(adf_results)

    # Interpret the test results
    p_value = adf_test[1]
    if p_value <= 0.05:
        print("\n☑ The data is stationary (Reject H0)")
    else:
        print("\n❌ The data is NOT stationary (Fail to reject H0)")

# Check stationarity of gold price data
check_stationarity(df[gold_price_col])
```

OUTPUT:



RESULT:

Thus, the program to check whether the give data is stationary or not is implemented Successfully