EXPERIMENT 4

AIM:

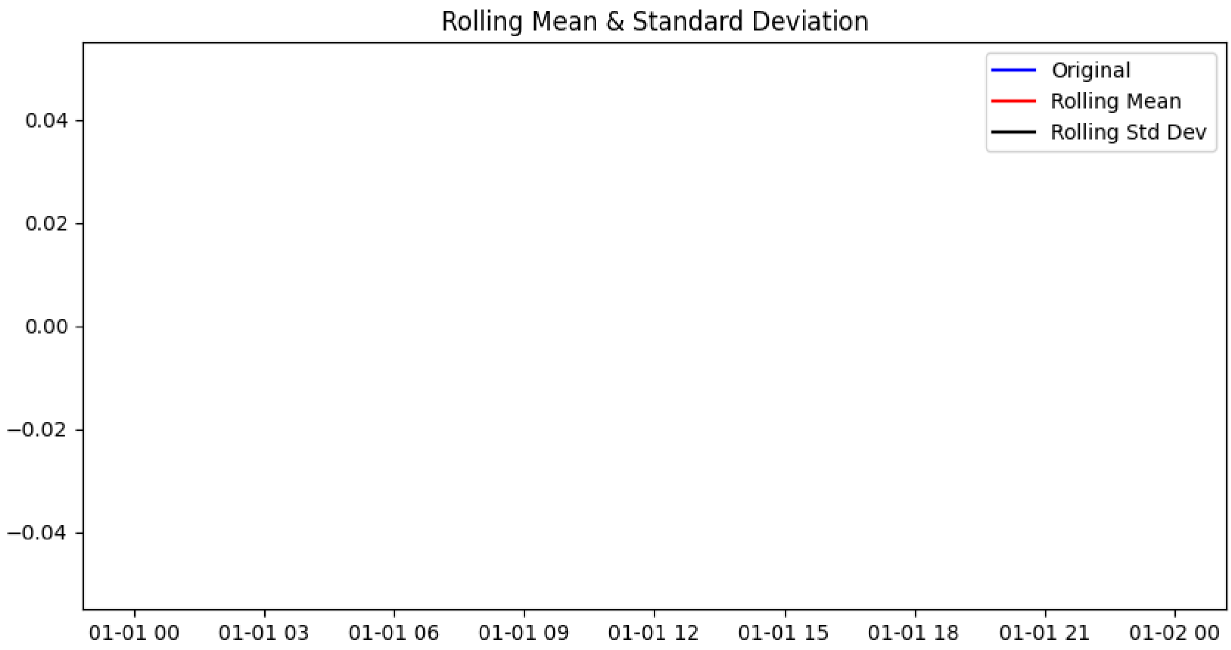
To check whether the given time series data is **stationary or non-stationary** using the **Augmented Dickey-Fuller (ADF) test** in Python. The program allows users to **upload a dataset, visualize the time series, and analyze stationarity.**

PROCEDURE AND CODE:

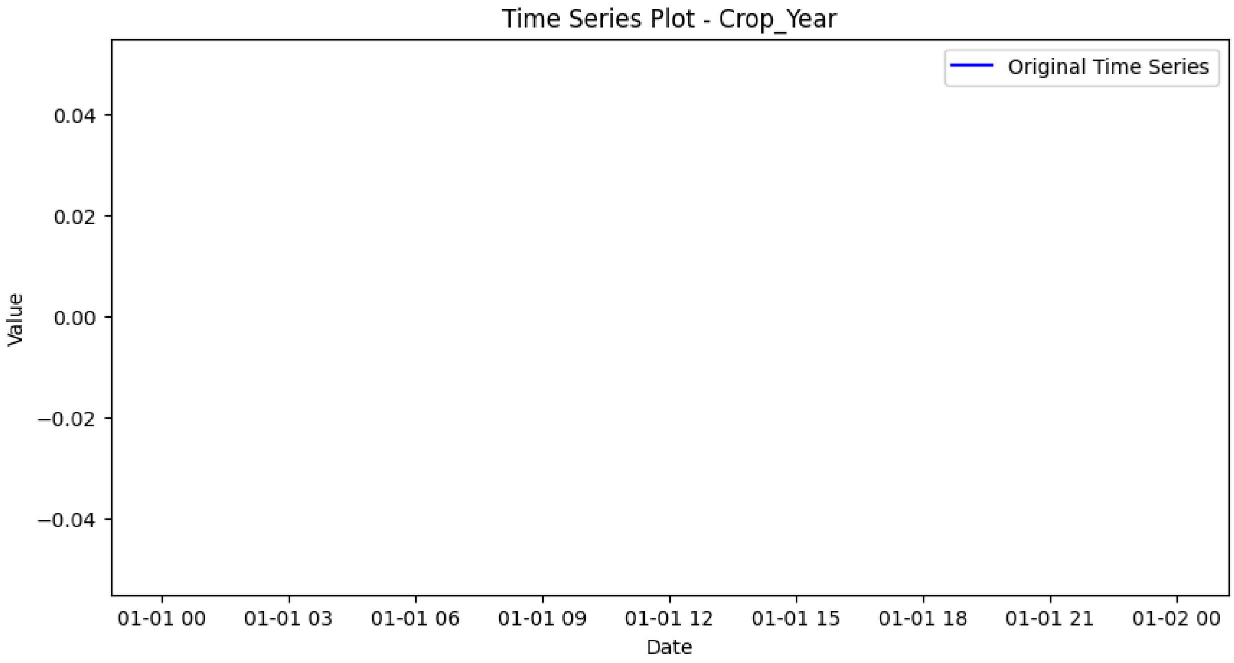
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.stattools import adfuller
from google.colab import files
import io

# Step 1: Upload file manually
print("📂 Please upload your dataset CSV file:")
uploaded = files.upload()

# Step 2: Read the uploaded file
filename = list(uploaded.keys())[0]  # Get uploaded filename
df = pd.read_csv(io.BytesIO(uploaded[filename]))  # Read CSV file

# Step 3: Display first few rows
print("\n📊 Dataset Preview:")
print(df.head())

# Step 4: Check column names
print("\n📝 Column Names:", df.columns)

# Step 5: Convert the first column to datetime (if possible)
df[df.columns[0]] = pd.to_datetime(df[df.columns[0]], errors="coerce")
df.set_index(df.columns[0], inplace=True)  # Set as index

# Step 6: Automatically detect a numerical column
numeric_cols = df.select_dtypes(include=["number"]).columns

if len(numeric_cols) == 0:
    print("\n❌ No numerical columns found in the dataset. Please check your file.")
else:
    selected_col = numeric_cols[0]  # Select first numerical column
    print(f"\n🔍 Using column for ADF test: {selected_col}")
```

```
# Step 7: Plot the time series data
plt.figure(figsize=(10, 5))
plt.plot(df[selected_col], label="Original Time Series", color="blue")
plt.title(f"Time Series Plot - {selected_col}")
plt.xlabel("Date")
plt.ylabel("Value")
plt.legend()
plt.show()

# Step 8: Define function to check stationarity
def check_stationarity(timeseries):
    rolmean = timeseries.rolling(window=12).mean()
    rolstd = timeseries.rolling(window=12).std()

    # Plot rolling mean & standard deviation
    plt.figure(figsize=(10, 5))
    plt.plot(timeseries, color="blue", label="Original")
    plt.plot(rolmean, color="red", label="Rolling Mean")
    plt.plot(rolstd, color="black", label="Rolling Std Dev")
    plt.legend(loc="best")
    plt.title("Rolling Mean & Standard Deviation")
    plt.show()

    # Perform Augmented Dickey-Fuller test
    print("\n📉 Augmented Dickey-Fuller Test Results:")
    result = adfuller(timeseries.dropna(), autolag="AIC")
    labels = ["Test Statistic", "p-value", "# Lags Used", "Number of Observations Used"]
    for label, value in zip(labels, result[:4]):
        print(f"{label}: {value}")

    # Interpret the results
    if result[1] <= 0.05:
        print("\n✅ Conclusion: The data is **stationary** (Reject H0)")
    else:
        print("\n❌ Conclusion: The data is **non-stationary** (Fail to Reject H0)")

# Step 9: Run the stationarity test
check_stationarity(df[selected_col])
```

CONCLUSION:



```
📉 Augmented Dickey-Fuller Test Results:
Test Statistic: -4.183812405864463
p-value: 0.0007017290683872245
# Lags Used: 28
Number of Observations Used: 19660

✅ Conclusion: The data is **stationary** (Reject H0)
```



Result:

The Augmented Dickey-Fuller (ADF) test determines whether the given time series data is **stationary or non-stationary** based on the **p-value**. If **p-value ≤ 0.05**, the data is **stationary**, otherwise, it is **non-stationary**. ✅📉