**Exp no: 4 Implement program to check stationary of a time series data**

**Date: 25/3/25**

**Aim:**

The aim of this analysis is to implement program to check stationary of a time series data in the Air Passengers dataset.

**Objectives:**

The primary objective of this analysis is to check the stationarity of the time series data of monthly international airline passengers using the Augmented Dickey-Fuller (ADF) test and visual inspection. Stationarity is crucial for many time series forecasting methods (like ARIMA) that assume the underlying data is stationary. By performing the ADF test and visualizing the data, we aim to determine whether the series exhibits trends or seasonality that would require transformation (such as differencing) to make it stationary.

**Background/Scope:**

Time series data often exhibits patterns such as trends, seasonality, and cyclic behavior. For time series models like ARIMA, stationarity is a key assumption. A stationary time series has a constant mean, variance, and autocovariance over time. If the time series is non-stationary, transformations such as differencing may be required to make it suitable for modeling. The Air Passenger dataset, which provides monthly counts of international airline passengers from 1949 to 1960, is a widely used example for studying time series analysis. The scope of this analysis is to test the stationarity of this dataset using both statistical (ADF test) and visual methods (rolling statistics), and to interpret the results.

**Steps for Time Series Sales Data Preprocessing:**

**Step 1: Load the Dataset**

The first step is to load the Air Passenger dataset into a pandas DataFrame. The dataset contains monthly international airline passenger data indexed by date.

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller

# Load the Air Passenger dataset

url = '/content/AirPassengers.csv'  # Update with the correct path

data = pd.read\_csv(url, header=0, index\_col=0, parse\_dates=True)

# Display first few rows of data

print(data.head())

**Step 2: Visualize the Time Series Data**

Plot the time series data to visualize the trend of airline passengers over time.

import matplotlib.pyplot as plt

# Plot the time series

plt.figure(figsize=(10, 6))

plt.plot(df.index, df['#Passengers'], label='Passengers')

plt.title('Monthly International Airline Passengers (1949-1960)')

plt.xlabel('Date')

plt.ylabel('Number of Passengers')

plt.legend()

plt.show()

A graph showing the growth of passengers

AI-generated content may be incorrect.

**Step 3: Plot Rolling Statistics (Moving Mean and Standard Deviation):**

Rolling statistics can help us identify if the mean and variance are constant over time. If they change, it suggests the data is non-stationary.

# Rolling Statistics (Moving Mean and Moving Standard Deviation)

rolling\_mean = data.rolling(window=12).mean()  # 12-month moving average

rolling\_std = data.rolling(window=12).std()    # 12-month moving standard deviation

# Plot the rolling statistics

plt.figure(figsize=(10,6))

plt.plot(data, label="Original Series")

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

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

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

plt.xlabel("Year")

plt.ylabel("Passengers")

plt.legend(loc='best')

plt.show()

A graph of a number of years

AI-generated content may be incorrect.

**Step 4: Perform Augmented Dickey-Fuller (ADF) Test:**

Apply the ADF test to statistically test if the time series is stationary. The test’s p-value helps determine whether we can reject the null hypothesis of a unit root (non-stationarity).

# Function to perform Augmented Dickey-Fuller test

def adf\_test(series):

    result = adfuller(series)

    print("ADF Statistic: ", result[0])

    print("p-value: ", result[1])

    print("Critical Values: ")

    for key, value in result[4].items():

        print(f"\t{key}: {value}")

    if result[1] <= 0.05:

        print("The series is stationary.")

    else:

        print("The series is not stationary.")

# Perform ADF test on the dataset to check for stationarity

adf\_test(data)

**Step 5: Interpret Results:**

ADF Test: The p-value and critical values from the ADF test will help you determine

stationarity. If the p-value is less than 0.05, the series is likely stationary.

Rolling Statistics: If the rolling mean and standard deviation vary significantly over time, the series is likely non-stationary.

**OUTPUT:**

ADF Statistic: 0.8153688792060498

p-value: 0.991880243437641

Critical Values:

1%: -3.4816817173418295

5%: -2.8840418343195267

10%: -2.578770059171598

The series is not stationary.

**Step 6: Transformation:**

The given code applies a **log transformation** to stabilize variance, followed by **first-order differencing** to remove trend, and **seasonal differencing (12-month lag)** to eliminate yearly seasonality. These transformations help convert the non-stationary time series into a stationary one, making it suitable for time series modeling and forecasting using methods like **ARIMA** or **SARIMA**.

# Apply log transformation to stabilize variance

data\_log = np.log(data)

# Apply first-order differencing to remove trend

data\_log\_diff = data\_log.diff().dropna()

# Apply seasonal differencing (12-month lag) to handle seasonality

data\_seasonal\_diff = data\_log\_diff.diff(12).dropna()

# Plot the seasonally differenced data

plt.figure(figsize=(10,6))

plt.plot(data\_seasonal\_diff, label="Seasonally Differenced Air Passengers")

plt.title("Seasonally Differenced Monthly International Airline Passengers")

plt.xlabel("Year")

plt.ylabel("Seasonal Differenced Log(Passengers)")

plt.legend(loc='best')

plt.show()

# Perform ADF test on the seasonally differenced data

print("Seasonally Differenced Series ADF Test:")

adf\_test(data\_seasonal\_diff['#Passengers'])

A graph showing a number of different types of flight

AI-generated content may be incorrect.

Seasonally Differenced Series ADF Test:

ADF Statistic: -4.4433249418311425

p-value: 0.00024859123113838495

Critical Values:

1%: -3.4870216863700767

5%: -2.8863625166643136

10%: -2.580009026141913

The series is stationary.

**Result:**

Thus program to check stationary of air passenger time series data has been implemented successfully.