**8. Create an ARIMA model for time series forecasting.**

|  |  |
| --- | --- |
| **EX.N0 : 8** | **Create an ARIMA model for time series forecasting.** |
| **DATE : 07/04/2025** |

**AIM:**

To Create an ARIMA model for time series forecasting.

**PROGRAM:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

from pandas.plotting import register\_matplotlib\_converters

register\_matplotlib\_converters()

# Sample population dataset (Year vs Population in millions)

data = {

'Year': pd.date\_range(start='2000', periods=21, freq='Y'),

'Population': [1000, 1020, 1045, 1070, 1090, 1110, 1140, 1170, 1200, 1235,

1260, 1290, 1320, 1350, 1385, 1410, 1440, 1470, 1500, 1530, 1560]

}

df = pd.DataFrame(data)

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

# Fit ARIMA model (p=1, d=1, q=1) – you can tune this

model = ARIMA(df['Population'], order=(1, 1, 1))

model\_fit = model.fit()

# Summary of the model

print(model\_fit.summary())

# Forecast the next 5 years

forecast\_steps = 5

forecast = model\_fit.forecast(steps=forecast\_steps)

forecast\_years = pd.date\_range(start=df.index[-1] + pd.DateOffset(years=1), periods=forecast\_steps, freq='Y')

# Plot original data and forecast

plt.figure(figsize=(10, 5))

plt.plot(df.index, df['Population'], label='Historical Population', marker='o')

plt.plot(forecast\_years, forecast, label='Forecasted Population', marker='x', linestyle='--', color='red')

plt.xlabel('Year')

plt.ylabel('Population (millions)')

plt.title('ARIMA Forecast of Population')

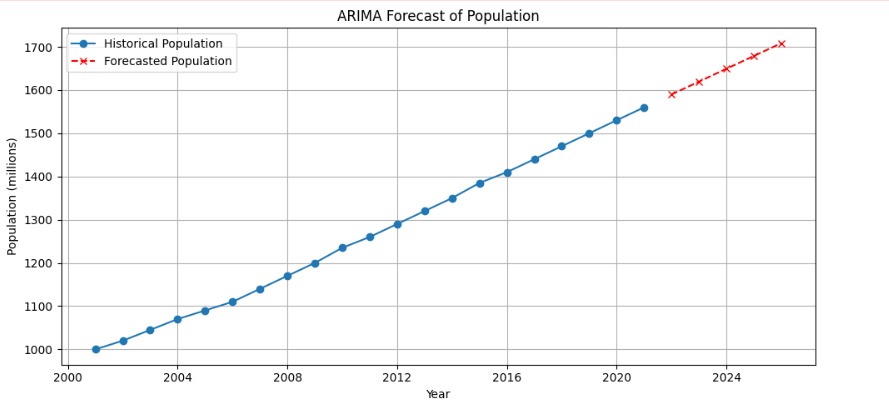
plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**OUTPUT:**



**RESULT:**

Thus, the program for Create an ARIMA model for time series forecasting is executed successfully.