8. CREATE AN ARIMA MODEL FOR TIME SERIES FORECASTING

AIM:

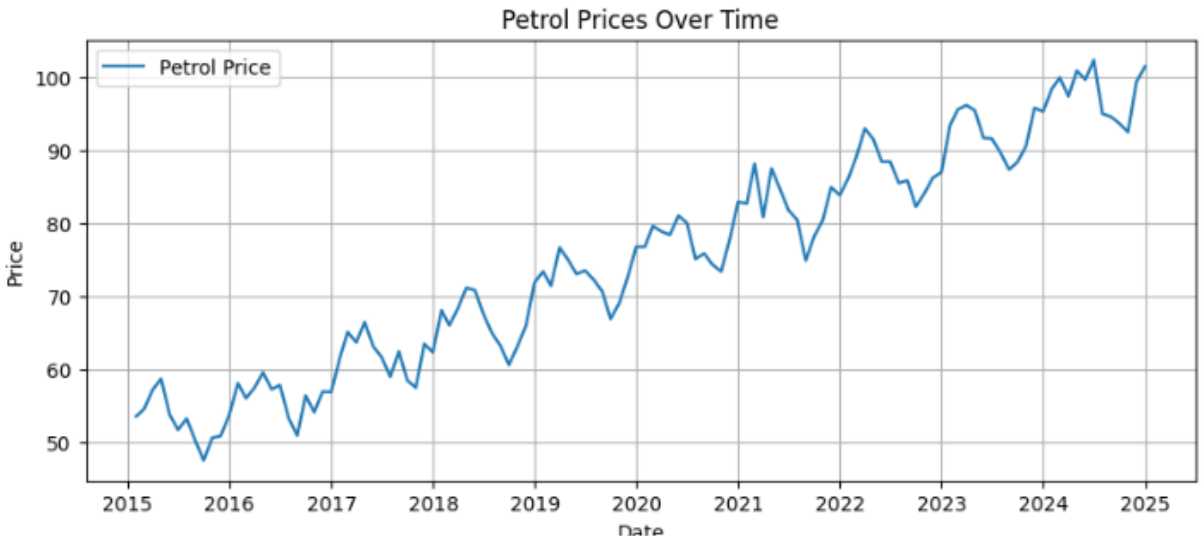
To create an ARIMA model for time series forecasting.

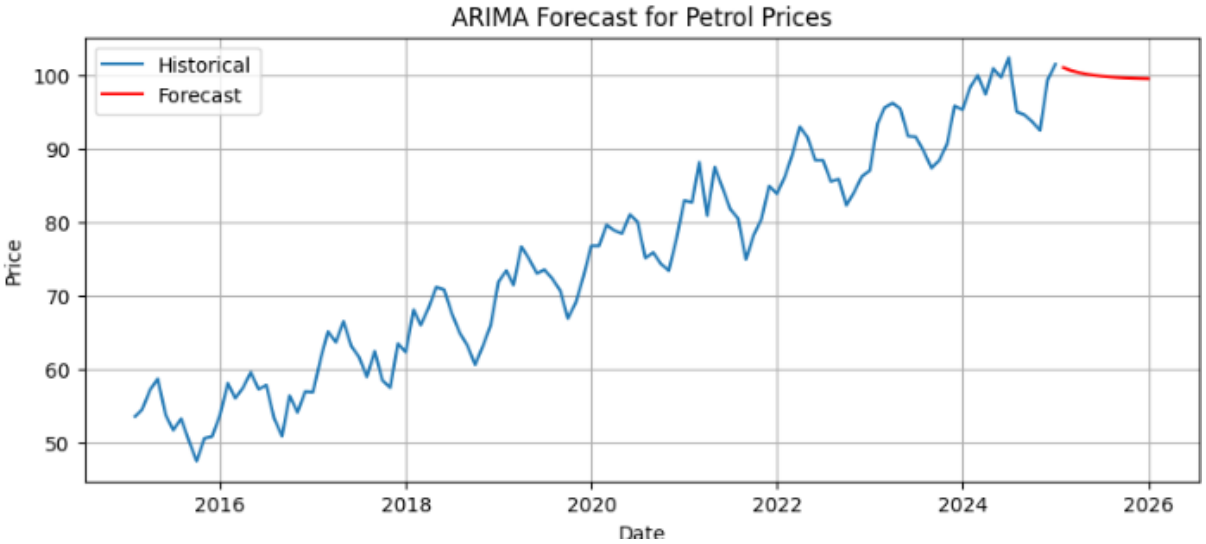
PROCEDURE:

```
# Step 1: Generate synthetic petrol price data
np.random.seed(42)
dates = pd.date_range(start='2015-01-01', periods=120, freq='M')  # 10 years monthly
trend = np.linspace(50, 100, 120)
seasonal = 5 * np.sin(2 * np.pi * dates.month / 12)
noise = np.random.normal(0, 2, 120)
prices = trend + seasonal + noise

df = pd.DataFrame({'Date': dates, 'Petrol_Price': prices})
df.set_index('Date', inplace=True)

# Step 2: Plot original data
plt.figure(figsize=(10, 4))
plt.plot(df, label='Petrol Price')
plt.title("Petrol Prices Over Time")
plt.xlabel("Date")
plt.ylabel("Price")
```

```
plt.grid(True)
plt.legend()
plt.show()

# Step 3: Fit ARIMA model (order can be tuned)
model = ARIMA(df['Petrol_Price'], order=(2, 1, 2))  # (p,d,q)
model_fit = model.fit()

# Step 4: Forecast next 12 months
forecast = model_fit.forecast(steps=12)
forecast_index = pd.date_range(df.index[-1] + pd.DateOffset(months=1),
periods=12, freq='M')
forecast_series = pd.Series(forecast, index=forecast_index)

# Step 5: Plot forecast
plt.figure(figsize=(10, 4))
plt.plot(df['Petrol_Price'], label='Historical')
plt.plot(forecast_series, label='Forecast', color='red')
plt.title("ARIMA Forecast for Petrol Prices")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT:





OUTPUT:

The program to execute an ARIMA model for time series forecasting has been executed successfully