import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from scipy import stats

# Step 1: Generate Synthetic Time Series Data

np.random.seed(42)

date\_range = pd.date\_range(start='2024-01-01', periods=60, freq='D')

prices = np.random.normal(loc=100, scale=5, size=60)

# Introduce missing values

missing\_indices = np.random.choice(60, 6, replace=False)

prices[missing\_indices] = np.nan

# Introduce outliers

outlier\_indices = np.random.choice(60, 3, replace=False)

prices[outlier\_indices] = [200, 250, 180]

# Create DataFrame

df = pd.DataFrame({'Date': date\_range, 'Price': prices})

# Save to CSV

csv\_path = 'petrol\_prices\_timeseries.csv'

df.to\_csv(csv\_path, index=False)

# Step 2: Load Data

loaded\_df = pd.read\_csv(csv\_path, parse\_dates=['Date'])

# Step 3: Handle Missing Data

loaded\_df['Price'].fillna(method='ffill', inplace=True) # Forward fill

# Step 4: Detect & Remove Outliers

z\_scores = np.abs(stats.zscore(loaded\_df['Price']))

loaded\_df = loaded\_df[z\_scores < 2] # Keep values within 2 standard deviations

# Step 5: Normalize Data

scaler = MinMaxScaler()

loaded\_df['Normalized\_Price'] = scaler.fit\_transform(loaded\_df[['Price']])

# Step 6: Visualize Data

plt.figure(figsize=(12, 6))

plt.plot(loaded\_df['Date'], loaded\_df['Normalized\_Price'], marker='o', linestyle='-', color='b')

plt.xlabel('Date')

plt.ylabel('Normalized Petrol Price')

plt.title('Time Series Visualization of Cleaned Petrol Prices')

plt.grid(True)

plt.show()

#Line Plot (Trend Visualization)

plt.figure(figsize=(12, 6))

plt.plot(df['Date'], df['Price'], marker='o', linestyle='-', color='b')

plt.title('Petrol Price Trend Over Time')

plt.xlabel('Date')

plt.ylabel('Price')

plt.grid(True)

plt.show()

#rolling plot

df['Rolling\_Avg'] = df['Price'].rolling(window=7).mean()

plt.figure(figsize=(12, 6))

plt.plot(df['Date'], df['Price'], label='Original Price', alpha=0.5)

plt.plot(df['Date'], df['Rolling\_Avg'], label='7-Day Rolling Average', color='red', linewidth=2)

plt.title('Petrol Price with 7-Day Rolling Average')

plt.xlabel('Date')

plt.ylabel('Price')

plt.legend()

plt.grid(True)

plt.show()

#Bar Plot

plt.figure(figsize=(12, 6))

plt.bar(df['Date'], df['Price'], color='skyblue')

plt.title('Petrol Price Distribution by Date')

plt.xlabel('Date')

plt.ylabel('Price')

plt.xticks(rotation=45)

plt.grid(axis='y')

plt.show()

#Box Plot

plt.figure(figsize=(8, 6))

plt.boxplot(df['Price'].dropna(), vert=True, patch\_artist=True, boxprops=dict(facecolor='lightblue'))

plt.title('Box Plot of Petrol Prices')

plt.ylabel('Price')

plt.grid(axis='y')

plt.show()

#Histogram

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

plt.hist(df['Price'].dropna(), bins=15, color='orange', edgecolor='black')

plt.title('Histogram of Petrol Prices')

plt.xlabel('Price')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()