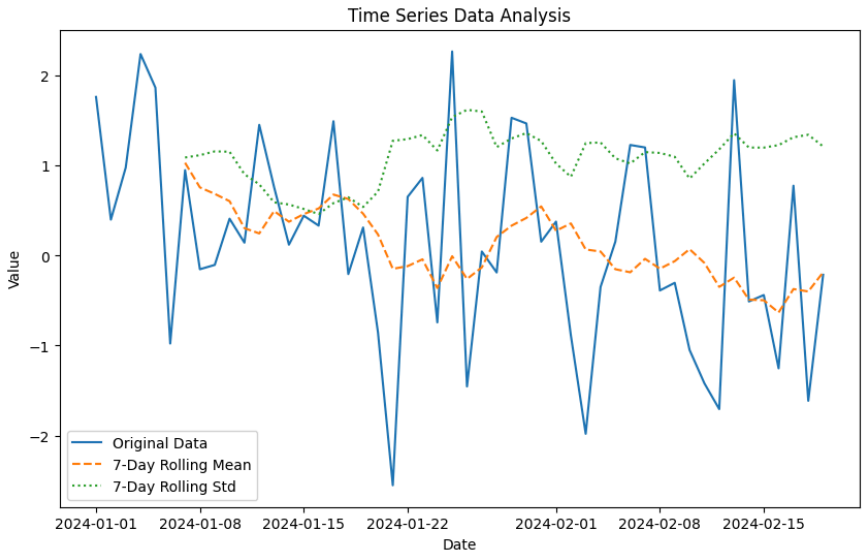
Program7: Time series using pandas Generate random time series data for the period from January 1, 2024, to February 19, 2024. Resample the data to obtain monthly averages. Calculate rolling statistics (7-day rolling mean and standard deviation). Plots the original data along with the rolling mean and rolling standard deviation.

```
#Import pandas, Numpy and Matplotlib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Generate some random time series data
np.random.seed(0)
date_range = pd.date_range(start='2024-01-01', end='2024-02-19', freq='D')
data = np.random.randn(len(date_range))
ts = pd.Series(data, index=date_range)
# Resampling (convert daily data to monthly data)
monthly_ts = ts.resample('M').mean()
# Calculating rolling statistics (rolling mean and rolling standard deviation)
rolling_mean = ts.rolling(window=7).mean()  # 7-day rolling mean
rolling_std = ts.rolling(window=7).std()
# 7-day rolling standard deviation
# Plotting time-based data
plt.figure(figsize=(10, 6))
plt.plot(ts, label='Original Data')
plt.plot(rolling_mean, label='7-Day Rolling Mean', linestyle='--')
plt.plot(rolling_std, label='7-Day Rolling Std', linestyle=':')
plt.title('Time Series Data Analysis')
plt.xlabel('Date')
plt.ylabel('Value')
plt.legend()
plt.show()
```



Conclusion: The provided program demonstrates time series analysis using Pandas, including data generation, resampling, calculation of rolling statistics, and visualization. This analysis technique enables understanding and forecasting sequential data patterns effectively. Through random data generation, resampling, and rolling statistics calculation, insights into trends and variability in the data over time are obtained. Visualization aids in interpreting time series data, facilitating the identification of patterns, anomalies, and trends. This program serves as a basic example, laying the groundwork for more advanced analyses and forecasting tasks.