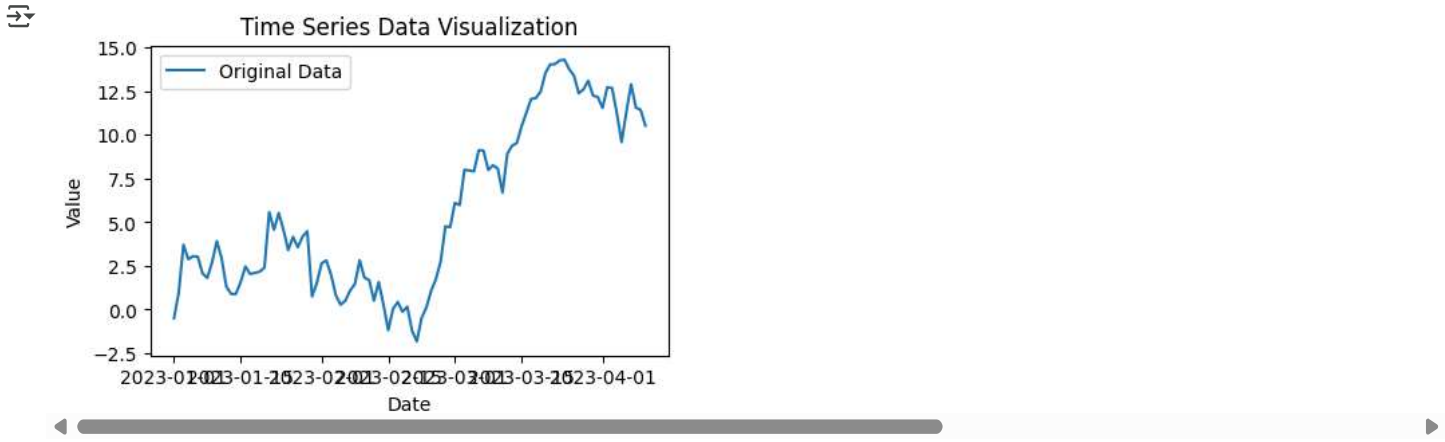
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
# Step 1: Generate Sample Time Series Data
dates = pd.date_range(start='2023-01-01', periods=100, freq='D')
values = np.cumsum(np.random.randn(100))  # Random walk
df = pd.DataFrame({'Date': dates, 'Value': values})
df.set_index('Date', inplace=True)
```

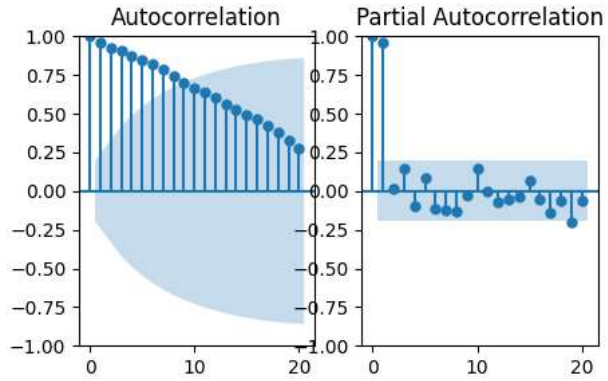
```
# Step 2: Visualizing the Time Series Data
plt.figure(figsize=(5, 3))
plt.plot(df.index, df['Value'], label='Original Data')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Time Series Data Visualization")
plt.legend()
plt.show()
```




```
# Step 3: Checking Stationarity using Augmented Dickey-Fuller (ADF) Test
def check_stationarity(timeseries):
    result = adfuller(timeseries)
    print("ADF Statistic:", result[0])
    print("p-value:", result[1])
    if result[1] <= 0.05:
        print("Data is stationary.")
    else:
        print("Data is not stationary.")
check_stationarity(df['Value'])
# Differencing if data is not stationary
df['Differenced'] = df['Value'].diff().dropna()
```

```
ADF Statistic: -1.2218985248835477
p-value: 0.6641052998293074
Data is not stationary.
```

```
# Step 4: Plot ACF and PACF
graph, axes = plt.subplots(1, 2, figsize=(5, 3))
plot_acf(df['Value'].dropna(), ax=axes[0])
plot_pacf(df['Value'].dropna(), ax=axes[1])
plt.show()
```




```
# Step 5: Building ARIMA Model (p=1, d=1, q=1 as example)
model = ARIMA(df['Value'], order=(1, 1, 1))
model_fit = model.fit()
print(model_fit.summary())


# Step 6: Forecasting Future Values
forecast_steps = 10
forecast = model_fit.forecast(steps=forecast_steps)
future_dates = pd.date_range(df.index[-1], periods=forecast_steps+1, freq='D')[1:]


plt.figure(figsize=(5, 3))
plt.plot(df.index, df['Value'], label='Original Data')
plt.plot(future_dates, forecast, label='Forecast', linestyle='dashed', color='red')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("ARIMA Forecasting")
plt.legend()
plt.show()
```