Time Series Stationarity Check

Aim

To analyze and determine the stationarity of a given time series using the Augmented Dickey-Fuller (ADF) test and visualize its properties through rolling statistics and autocorrelation plots.

Algorithm

1. Import necessary libraries.
2. Read the time series data from a CSV file.
3. Apply the Augmented Dickey-Fuller (ADF) test.
4. Calculate and plot rolling mean and rolling standard deviation.
5. Plot the Autocorrelation Function (ACF).
6. Determine whether the time series is stationary based on the p-value.
7. Display results and visualization.

Prerequisites

Ensure the following Python libraries are installed:

Unset

```
pip install pandas numpy matplotlib statsmodels
```

Code Explanation

Importing Required Libraries

Python

```python
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

```
from statsmodels.tsa.stattools import adfuller
```

- pandas: For handling time series data.
- numpy: For numerical operations.
- matplotlib.pyplot: For visualization.
- adfuller from statsmodels.tsa.stattools: Performs the Augmented Dickey-Fuller test to check stationarity.

Function: check_stationarity

Python

```python
def check_stationarity(timeseries):
    result = adfuller(timeseries)
    p_value = result[1]
    critical_values = result[4]

    if p_value < 0.05:
        stationary = True
        status = "Time Series is Stationary"
    else:
        stationary = False
        status = "Time Series is Non-Stationary"
```

- This function applies the ADF test to determine if the time series is stationary.
- If the p-value is below 0.05, the null hypothesis (non-stationarity) is rejected, and the series is considered stationary.

Visualization

Python

```python
    fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(12, 8))


    timeseries.plot(ax=ax1, title="Original Time Series", color='blue')
    ax1.set_ylabel("Value")


    roll_mean = timeseries.rolling(window=12).mean()
    roll_std = timeseries.rolling(window=12).std()
    timeseries.plot(ax=ax2, label='Original', color='blue')
    roll_mean.plot(ax=ax2, label='Rolling Mean', color='red')
    roll_std.plot(ax=ax2, label='Rolling Std', color='black')
    ax2.set_ylabel("Value")
    ax2.legend(loc='best')
    ax2.set_title("Rolling Mean & Standard Deviation")
```

- The function generates three subplots:
    - The first plot shows the original time series.
    - The second plot displays the rolling mean and rolling standard deviation to visually assess stationarity.

Autocorrelation Function (ACF)

Python

```python
    try:
        from statsmodels.graphics.tsaplots import plot_acf
        plot_acf(timeseries, ax=ax3, lags=20)
        ax3.set_title("Autocorrelation Function (ACF)")
        plt.tight_layout()
        plt.show()
    except ImportError:
        print("Warning: statsmodels not installed. ACF plot not generated.")
```

- The third plot represents the autocorrelation function, which helps analyze dependencies in the time series.

Reading and Processing Data

Python

```
data = pd.read_csv('/content/Microsoft_Stock.csv', parse_dates=['Date'], index_col='Date')

timeseries = data['Close']

is_stationary, p_value, status, fig = check_stationarity(timeseries)
```

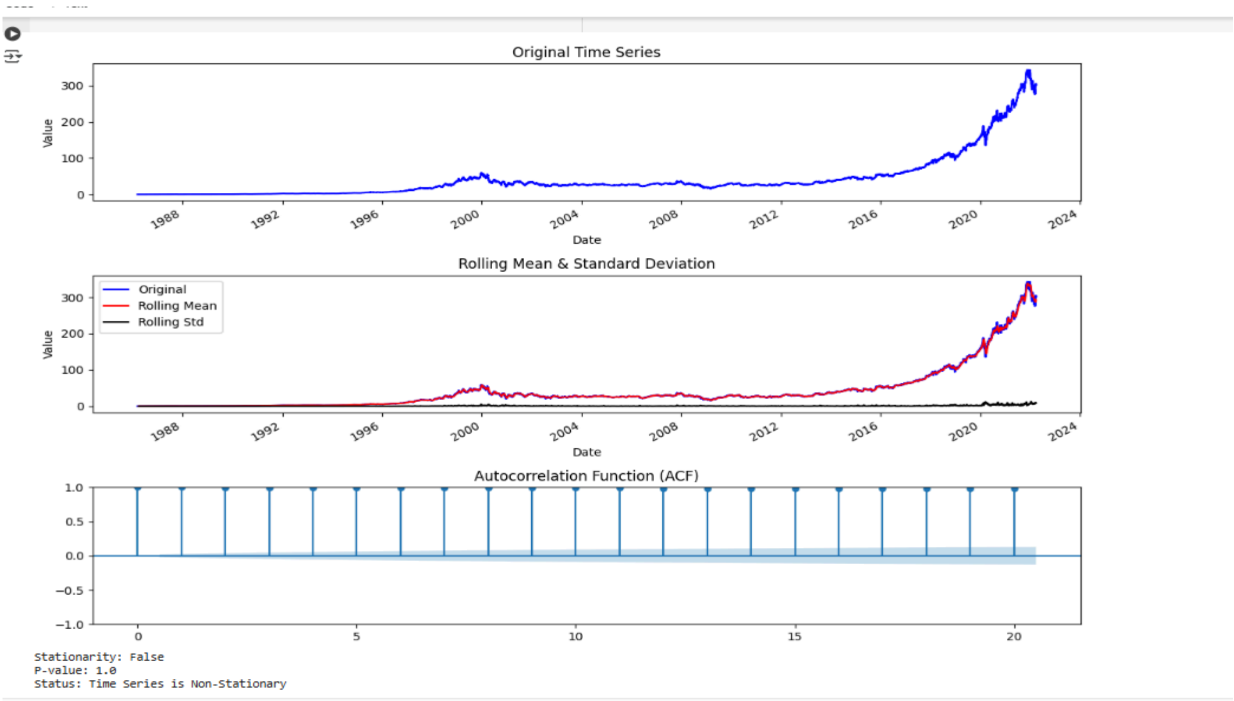
- The CSV file containing stock prices is read using `pandas.read_csv`.
- The `Close` price column is extracted as a time series.
- The `check_stationarity` function is called to analyze the time series.

Output Results

```python
print(f"Stationarity: {is_stationary}")

print(f"P-value: {p_value}")

print(f"Status: {status}")

plt.show()
```

- The function returns:
  - `is_stationary`: Boolean indicating stationarity.
  - `p_value`: The test's significance level.
  - `status`: A textual summary of the stationarity check.
- The results are printed to the console.

OUTPUT:-



```
Stationarity: False
P-value: 1.0
Status: Time Series is Non-Stationary
```

Result

This script provides a comprehensive method to check the stationarity of time series data using statistical and visual techniques. If the time series is non-stationary, techniques such as differencing or transformation may be required before further analysis.