

# Smoothing Time Series in Univariate Analysis

**Smoothing** is a technique used in time series analysis to remove noise or fluctuations, making patterns and trends in the data easier to analyze and interpret. It is commonly applied to highlight long-term trends, cyclic behaviors, or seasonal patterns in time series data.

## Key Concepts

### 1. Purpose of Smoothing

- To remove short-term fluctuations (noise) and reveal important patterns.
- To improve the interpretability of time series data for forecasting or analysis.

### 2. Common Smoothing Techniques

- **Moving Average:**
  - Calculates the average of a sliding window of values.
  - Smooths fluctuations by averaging neighboring data points.
- **Exponential Smoothing:**
  - Assigns exponentially decreasing weights to older observations.
  - Provides a balance between recent and past data.
- **Rolling Mean:**
  - A simple moving average where the window size determines the degree of smoothing.

## Syntax for Smoothing Techniques

### *Moving Average*

```
data['Smoothed'] = data['Value'].rolling(window=3).mean()
```

### *Exponential Smoothing*

```
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
model = SimpleExpSmoothing(data['Value']).fit(smoothing_level=0.2)
data['Smoothed'] = model.fittedvalues
```

### *Custom Smoothing (e.g., Weighted Moving Average)*

```
import numpy as np
data['Smoothed'] = np.convolve(data['Value'], np.ones(window)/window, mode='valid')
```

## Example:

### **Moving Average Smoothing**

```
import pandas as pd
import numpy as np
```

```

import matplotlib.pyplot as plt

# Create a sample time series dataset
data = {
    'Month': pd.date_range(start='2023-01', periods=12, freq='M'),
    'Sales': [120, 150, 130, 170, 200, 220, 210, 230, 250, 240, 260, 280]
}

# Create DataFrame
df = pd.DataFrame(data)

# Add Moving Average Smoothing (3-month window)
df['Smoothed_Sales'] = df['Sales'].rolling(window=3).mean()

# Plot Original and Smoothed Data
plt.figure(figsize=(10, 6))
plt.plot(df['Month'], df['Sales'], label='Original Sales', marker='o')
plt.plot(df['Month'], df['Smoothed_Sales'], label='Smoothed Sales (3-Month MA)',
color='orange', linewidth=2)
plt.title("Sales Time Series with Moving Average Smoothing")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.legend()
plt.grid()
plt.show()

```

### Output

- **Plot:** The smoothed line is less jagged than the original data, making trends more visible.
- **Data:**

	Month	Sales	Smoothed_Sales
0	2023-01-31	120	NaN
1	2023-02-28	150	NaN
2	2023-03-31	130	133.333333
3	2023-04-30	170	150.000000
4	2023-05-31	200	166.666667
5	2023-06-30	220	196.666667

1. **Dataset:**
  - Simulates monthly sales data over a year (12 data points).
2. **Moving Average Smoothing:**
  - A 3-month rolling window (window=3) is applied.

- Each smoothed value is the average of the current and the preceding two months:  
$$\text{Smoothed}_t = \frac{X_t + X_{t-1} + X_{t-2}}{3}$$
- For example, the smoothed value for March is:  $\frac{120+150+130}{3} = 133.33$

3. **Plotting:**

- The original sales data is plotted with markers.
- The smoothed line highlights the overall trend by reducing noise.

