Data Science methods V.Ger Travel

This Notebook will help you generate synthetic datasets for Time Series.
Instructions for setting parameters can be found at the end of the Notebook.

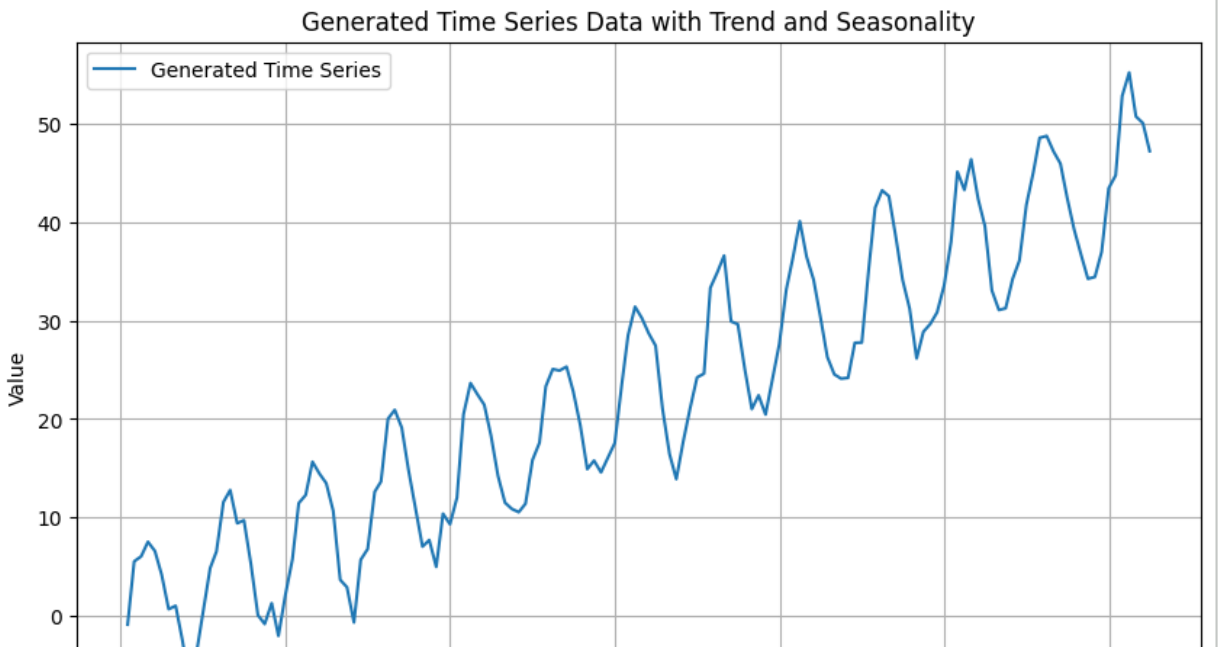
The generated Time Series are plotted and saved in a .csv file. After generation the Notebook provides a decomposition of the Time series. It is up to you to find the correct set of parameters based on the instructions provided and through inspecting the code below.

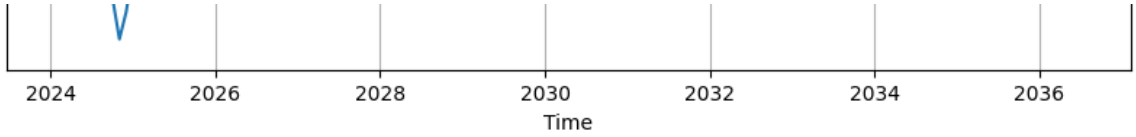
Show code

```
Drive already mounted at /content/drive; to attempt to forcibly remo
```

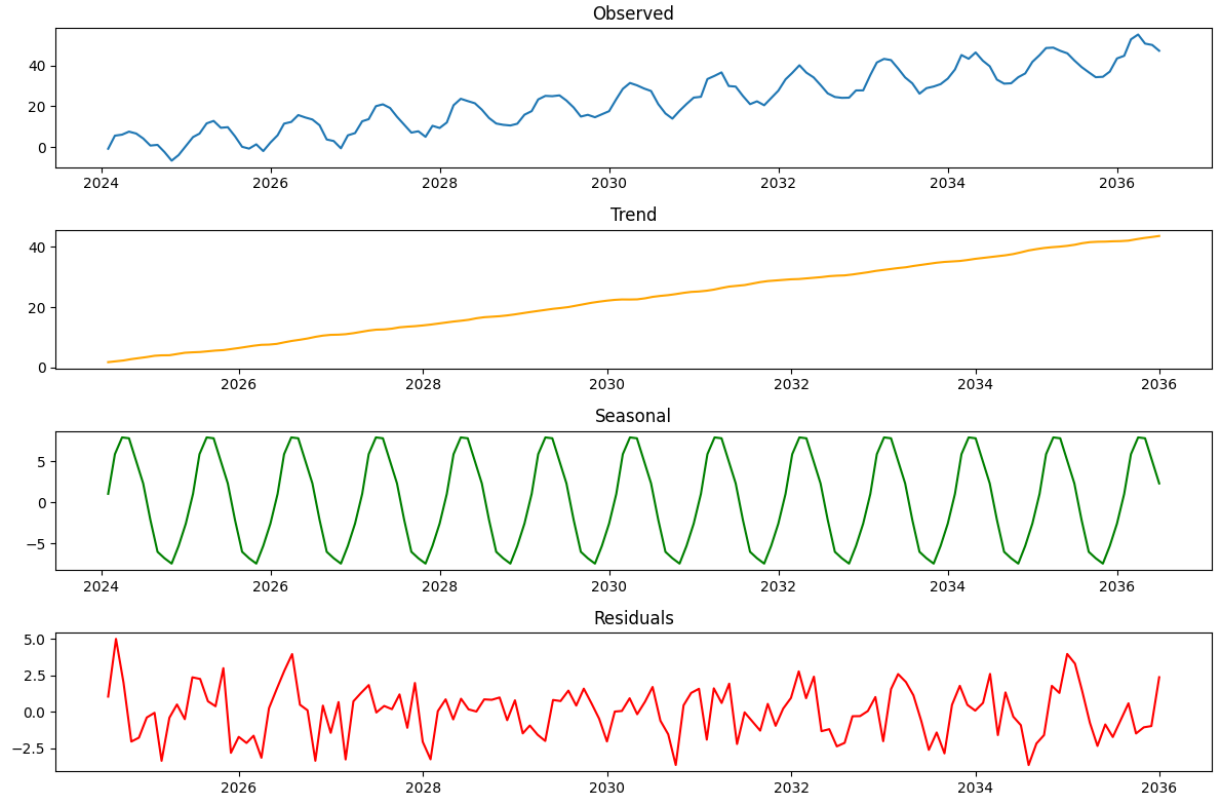
```
# Decompose the Time Series
t_s = generate_time_series(n_periods=150, frequency='months', trend_s

decompose_time_series(t_s)
```

```
/tmp/ipython-input-2138353997.py:16: FutureWarning: 'M' is deprecate
  time_index = pd.date_range(start='2024-01-01', periods=n_periods,
```






```
Time series data saved to 't_s.csv'.
```



1. n_periods (Number of Time Steps)

   Description: Specifies how many data points you want to generate. Usage: Affects the length of the time series. Example Values: n_periods=100 for a short time series. n_periods=500 for a longer time series. Default: 100.

2. frequency (Time Frequency)

   Description: Defines the time intervals of the series. Options: 'days': Daily time intervals. 'weeks': Weekly time intervals. 'months': Monthly time intervals. Usage: Determines the time range of the time series. Example Values: frequency='days' for daily observations. frequency='weeks' for weekly data. frequency='months' for monthly data. Default: 'days'.

3. trend_slope (Linear Trend Slope)

   Description: Controls how quickly the trend increases over time. Usage: Affects the magnitude of the overall increase in the series. Example Values: trend_slope=0.1: A slow-growing trend. trend_slope=0.5: A faster-growing trend. trend_slope=0: No trend (flat). Default: 0.3.

4. seasonal_period (Seasonal Pattern)

   Description: Specifies the type of seasonality to include in the series. Options: 'monthly': Seasonal variations occur every 12 periods (e.g., months in a year). 'weekly': Seasonal variations occur every 52 periods (e.g., weeks in a year). 'annual': Seasonal variations occur every 365 periods (e.g., days in a year). Usage: Adjust this to reflect real-world seasonality based on the data's frequency. Example Values: seasonal_period='monthly' for sales data. seasonal_period='weekly' for weekly activity patterns. seasonal_period='annual' for long-term trends. Default: 'monthly'.

5. seasonal_amplitude (Strength of Seasonality)

   Description: Controls the magnitude of the seasonal fluctuations. Usage: Sets the "height" of seasonal oscillations relative to the trend. Example Values: seasonal_amplitude=2: Subtle seasonal variation. seasonal_amplitude=10: Strong seasonal variation. Default: 5.

6. noise_level (Random Variation)

   Description: Adds random noise to simulate natural variability in the data. Usage: Higher values create noisier data, while lower values make it smoother. Example Values: noise_level=0: No noise. noise_level=0.5: Moderate noise. noise_level=2: High noise. Default: 0.5.

Tips for Setting Parameters:

```
Match Seasonality to Frequency: Ensure that the seasonal_period makes
    Example: For frequency='months', use seasonal_period='monthly' for
Control Noise for Clarity: Use noise_level=0 for clean series or incre
Trend Dominance: Ensure the trend_slope is much larger than seasonal_a
```