

Time Series

Fundamentals

What is a Time Series?

- A **time series** is a sequence of numerical **data** points in successive order, usually occurring in uniform intervals.
- **Time series *analysis*** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data.
- **Time series *forecasting*** is the use of a model to predict future values based on previously observed values.

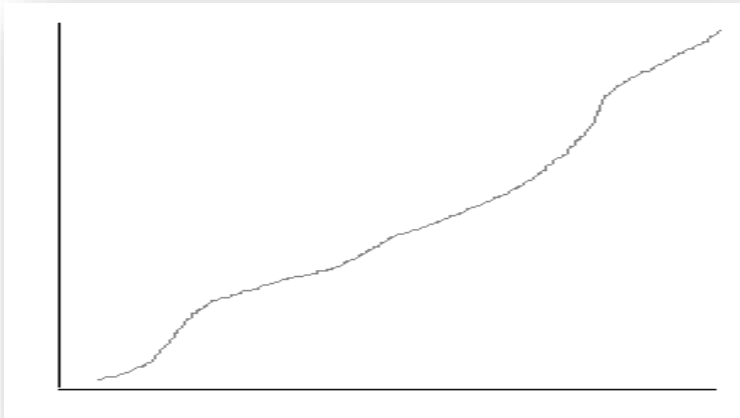
Courtesy: Wikipedia

Assumptions in Time Series Algorithms

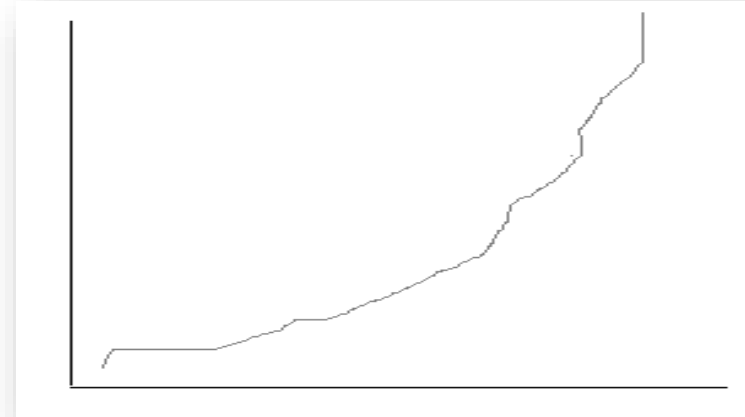
- Consecutive Observations in the series are equally spaced
- Series is indexed on specific period of time. e.g. Weekly, Daily, Yearly etc.
- There aren't any missing values

Types of Trends

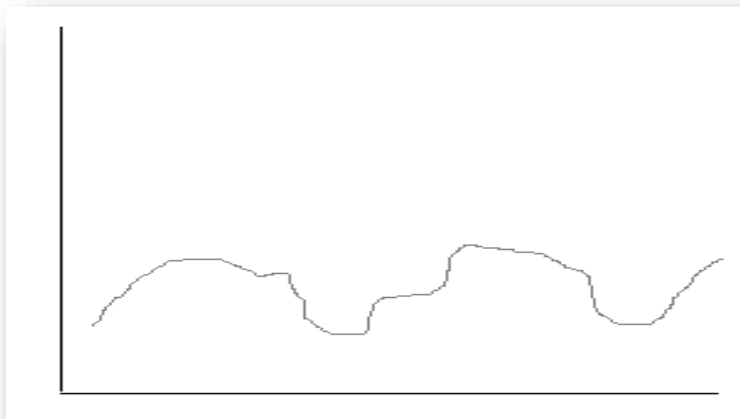
Linear



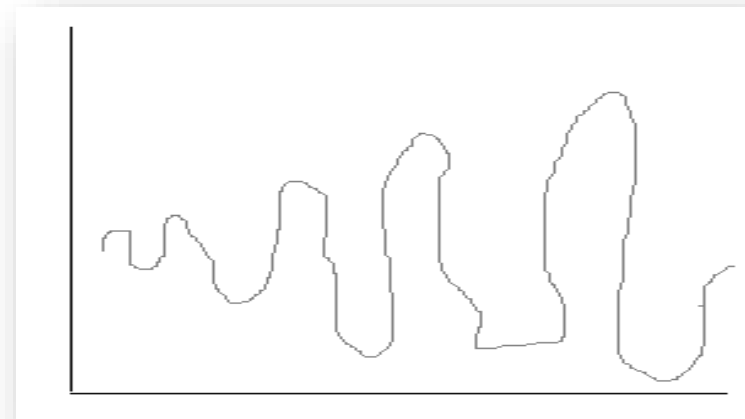
Rapid Growth



Periodic



Varying Variance



Some Transformations

- log: The `log()` function can linearize the rapid growth trend. It can also stabilize the varying variance series. It is only for positive values.
- diff: The `diff()` function can remove the linear trends. It can also remove periodic trends.

$$35 = 25 + 5 + 5 \quad \text{or} \quad 35 = 5 \times 7$$

Components of Time Series

- **Trend:** Indicates a long term increase or decrease in the data. It may be linear or non-linear.
- **Seasonal:** Seasonality is a pattern observed with regular intervals of time. e.g. Sale of woolen clothes increases in winter and is relatively low in other seasons.
- **Cyclic:** Data exhibits rise and fall not in regular time intervals. e.g. Recession and Boom
- **Random:** This is an error component. Also called irregular component.

Classical Decomposition

- There are two types of classical decompositions:
 - Additive
 - Multiplicative
- We assume here that the seasonal component is constant from year to year.
- Suppose that we have m seasonal periods. Then there are m seasonal values which are called *seasonal indices*.

Notations

- y_t : Value in time series at time t
- \hat{T}_t : Trend-cycle component (Moving Average) calculated for time t
- \hat{S}_t : Seasonal Index for time t

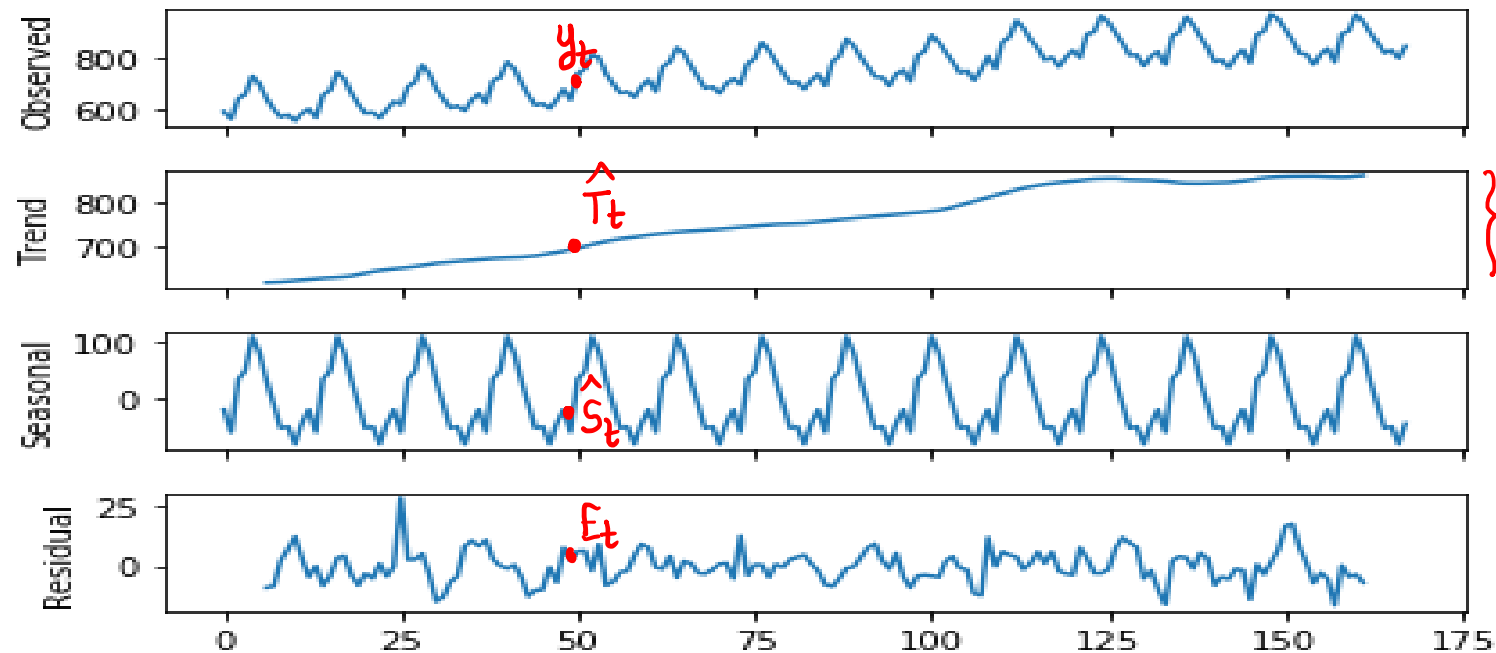
Additive Decomposition

1. If m is even number, then centered MA is calculated otherwise non-centered MA is calculated.
2. Calculate the de-trended series, $y_t - \hat{T}_t$
3. For estimating the seasonal component for each month, a simple average is calculated for detrended values for that particular month. It is denoted by \hat{S}_t .
4. The random component is calculated by subtracting seasonal and trend-cycle components. $\hat{E}_t = \hat{y}_t - \hat{T}_t - \hat{S}_t$

Example

```
In [55]: from statsmodels.tsa.seasonal import seasonal_decompose
....: from matplotlib import pyplot
....: series = df['Milk']
....: result = seasonal_decompose(series, model='additive', freq=12)

In [56]: result.plot()
....: pyplot.show()
```



$$y_t = \hat{T}_t + \hat{S}_t + E_t$$

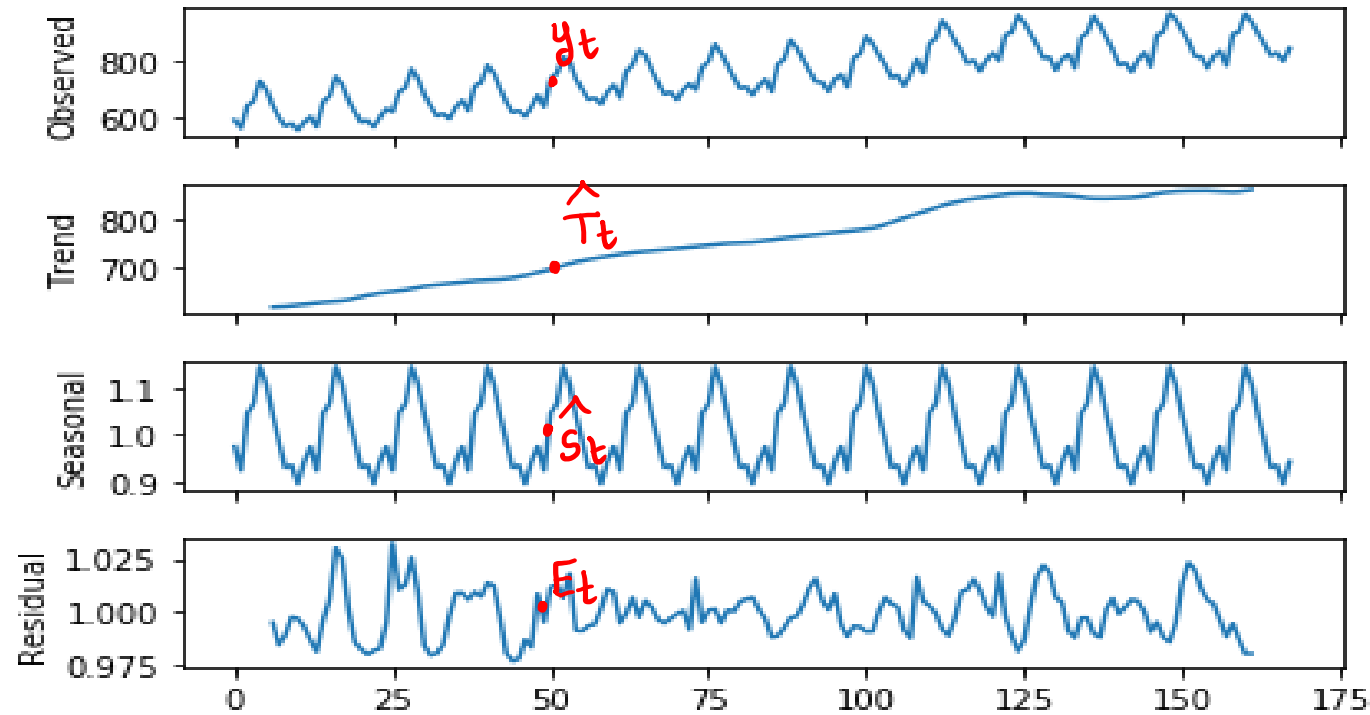
Multiplicative Decomposition

1. If m is even number, then centered MA is calculated otherwise non-centered MA is calculated.
2. Calculate the de-trended series, y_t/\hat{T}_t
3. For estimating the seasonal component for each month, a simple average is calculated for de-trended values for that particular month. It is denoted by \hat{S}_t .
4. The random component is calculated by subtracting seasonal and trend-cycle components. $\hat{E}_t = \hat{y}_t/(\hat{T}_t\hat{S}_t)$

Example

```
In [57]: result = seasonal_decompose(series, model='multiplicative', freq=12)
```

```
In [58]: result.plot()  
...: pyplot.show()
```



$$y_t = \hat{T}_t \hat{S}_t E_t$$

e.g.

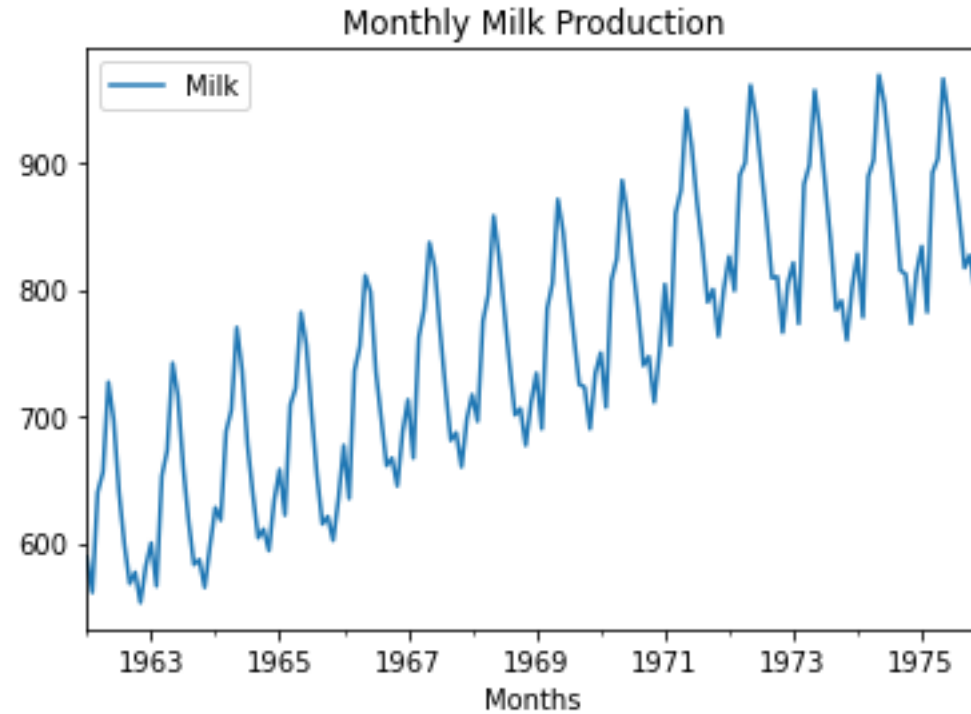
$$105 = 7 \times 5 \times 5$$

Resampling

Resampling

- In order to make the data consistent with time, resampling can be done.
- e.g. We may be given monthly data and want to change it into quarterly data

Monthly Data



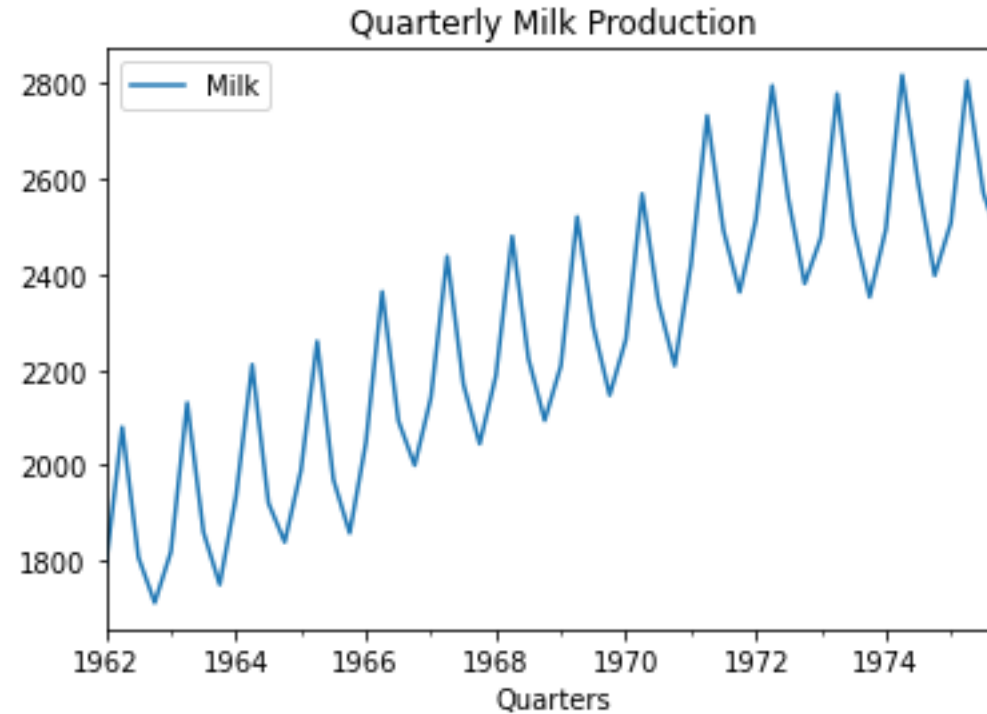
Month	1962-01	1962-02	1962-03	1962-04	...	1975-09	1975-10	1975-11	1975-12
Milk	589	561	640	656	...	817	827	797	843

Resampling Example

```
downsampled = df.resample('Q').sum()
```

```
downsampled.index.rename('Quarter', inplace=True)  
downsampled.plot()  
plt.title("Quarterly Milk Production")  
plt.xlabel("Quarters")  
plt.show()
```


Quarterly Data



Quarter	1962Q1	1962Q2	1962Q3	1962Q4	...	1975Q1	1975Q2	1975Q3	1975Q4
Milk	1790	2080	1807	1712	...	2508	2806	2571	2467

Questions?