In seasonal decomposition, a time series is broken down into three components:

1. **Trend**: Represents the long-term movement or progression in the data.
2. **Seasonal**: Captures the repeating pattern or seasonality in the data within each period (e.g., monthly or quarterly).
3. **Residual**: Represents the irregular, random variations or the "leftover" part of the data that is not explained by trend or seasonal components.

These three components are combined in either an **additive** or **multiplicative** model. The choice between them depends on the nature of the data:

- **Additive model**: Used when the seasonal variations and residuals are roughly constant over time. The model assumes:

$$\text{Observed} = \text{Trend} + \text{Seasonal} + \text{Residual}$$

- **Multiplicative model**: Used when seasonal variations and residuals change proportionally with the level of the trend. It assumes:

$$\text{Observed} = \text{Trend} \times \text{Seasonal} \times \text{Residual}$$

Let's go through the formulas and explanations for each component:

1. Trend Calculation

The **trend** component is a smoothed version of the original series and represents the overall direction or movement over a longer period, ignoring short-term fluctuations.

In Python, libraries like `statsmodels` calculate the trend by applying a moving average or a more sophisticated smoothing algorithm.

For instance, a **centered moving average** might be used to smooth out seasonal fluctuations and highlight the longer-term trend:

$$\text{Trend} = \text{Moving Average of Observed Data (e.g., 12-month average)}$$

2. Seasonal Calculation

The **seasonal** component captures recurring fluctuations at regular intervals (e.g., monthly, quarterly). In an additive model, the seasonal component measures how much above or below the trend the data is at specific times (such as in January vs. July), while in a multiplicative model, it represents the proportion of the trend.

- **Additive Model Seasonal Calculation**:

$$\text{Seasonal} = \text{Observed} - \text{Trend} - \text{Residual}$$

- **Multiplicative Model Seasonal Calculation**:

$$\text{Seasonal} = \frac{\text{Observed}}{\text{Trend} \times \text{Residual}}$$

For monthly data, for example, we could average the values for each month (January, February, etc.) over multiple years to estimate the seasonal component.

3. Residual Calculation

The **residual** component represents the remaining part of the series after removing the trend and seasonal patterns, capturing any irregularities or anomalies.

- **Additive Model Residual Calculation**:

$$\text{Residual} = \text{Observed} - (\text{Trend} + \text{Seasonal})$$

- **Multiplicative Model Residual Calculation**:

$$\text{Residual} = \frac{\text{Observed}}{\text{Trend} \times \text{Seasonal}}$$

The residuals should ideally center around zero in an additive model, indicating that no systematic patterns remain after removing the trend and seasonal effects.

Applying Seasonal Decomposition in Python with `seasonal_decompose`

Using the `seasonal_decompose` function from `statsmodels`, the trend, seasonal, and residual components are calculated automatically based on the type of model specified:

python Copy code

```python
from statsmodels.tsa.seasonal import seasonal_decompose # Assuming monthly_sales is the time series data decomposition = seasonal_decompose(monthly_sales, model='additive', period=12) # Extract each component trend = decomposition.trend seasonal = decomposition.seasonal residual = decomposition.resid
```

This function will calculate the components based on the given model and period.

1. `trend` is calculated using moving averages or other smoothing techniques.
2. `seasonal` is derived from the repeating patterns after subtracting the trend.
3. `residual` is what remains after accounting for both trend and seasonal variations.

By understanding each component, you can better analyze, forecast, and identify anomalies in the data.