Seasonal and Trend decomposition using LOESS (STL) - Overview

Time series decomposition

Contents

STL TO EXTRACT TREND

STL TO EXTRACT SEASONALITY

DISCUSS PRACTICALITIES

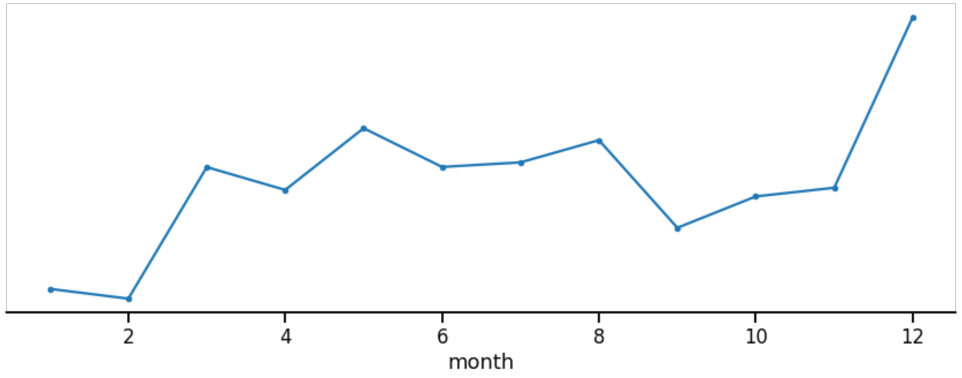
Recap: Classical decomposition - Seasonality

1. Identify order of seasonality T

2. Compute trend using T-MA (if odd) or 2 x T-MA (if even)

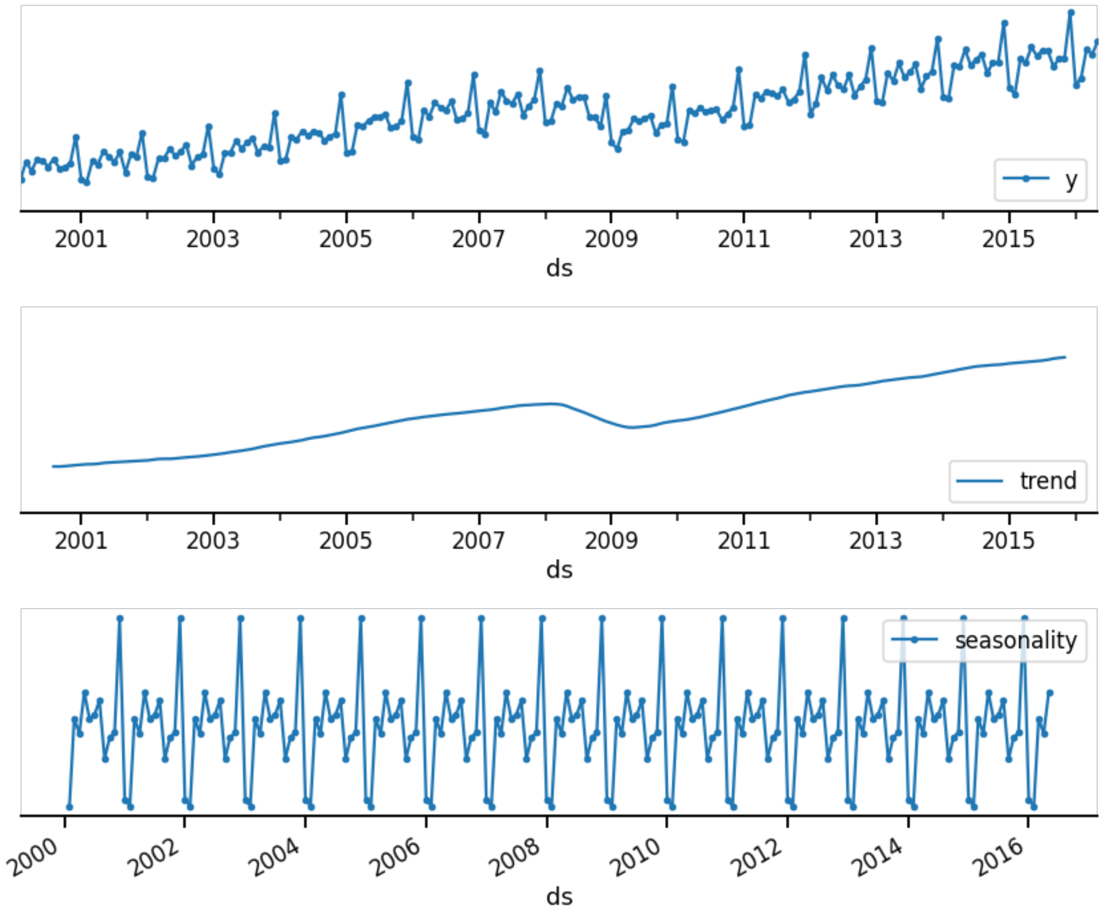
3. De-trend the data:
   1. If additive: $y_t - trend_t$
   2. If multiplicative: $y_t \;/\; trend_t$

4. Average the de-trended data over each seasonal index to remove noise (e.g., for monthly data average all the May months)



Recap: Classical decomposition - Seasonality

1. Identify order of seasonality T
2. Compute trend using T-MA (if odd) or 2 x T-MA (if even)
3. De-trend the data:
   1. If additive: $y_t - trend_t$
   2. If multiplicative: $y_t \ / \ trend_t$
4. Average the de-trended data over each seasonal index to remove noise (e.g., for monthly data average all the May months)

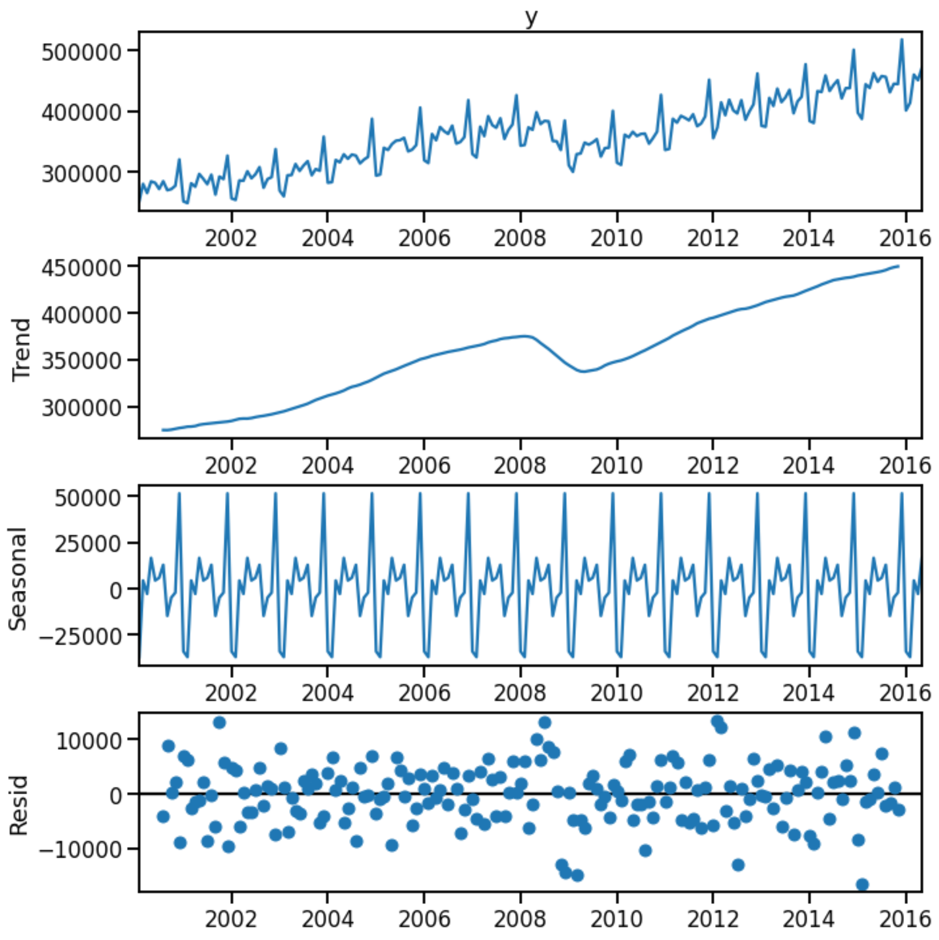


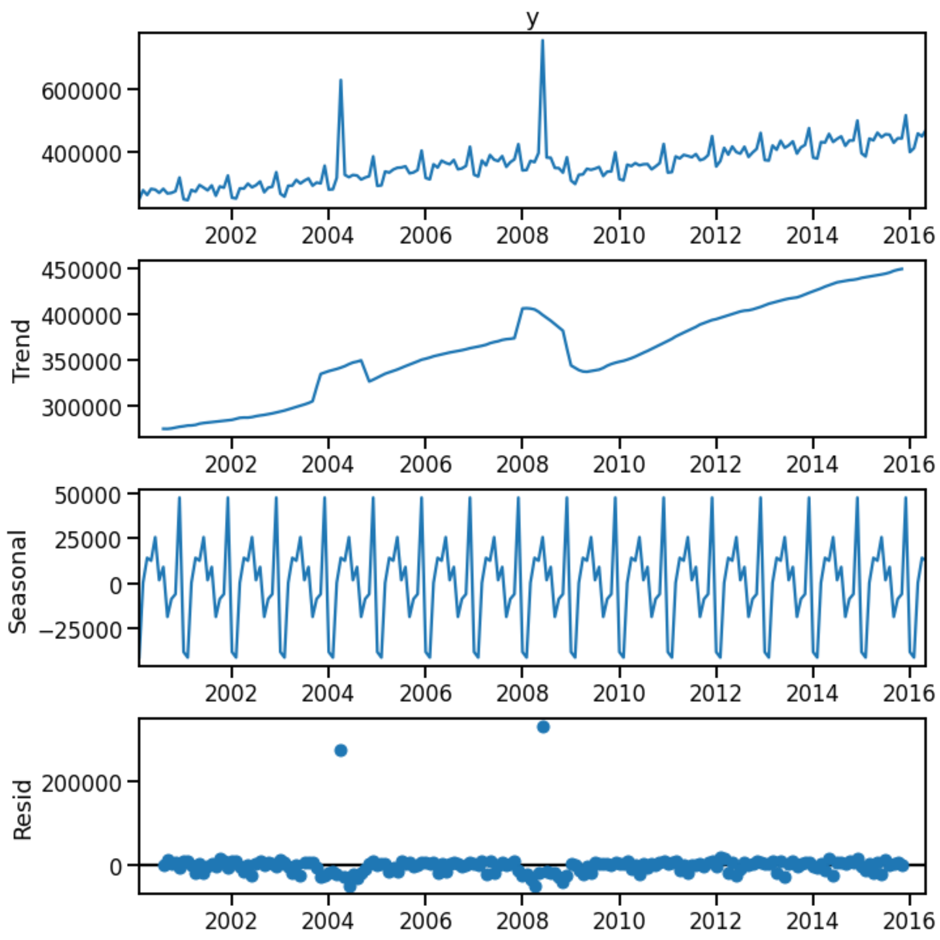
Limitations of moving averages

- Not robust to outliers
- Missing data at the edges
- Over-smooths rapid changes in trend
- The seasonal component does not change in time (e.g., the seasonal component for every January is the same)

Date	y	mean
2020-02-12	23	
2020-02-13	30	41.0
2020-02-14	70	43.3
2020-02-15	**30**	41.7
2020-02-16	**25**	**25.7**
2020-02-17	**22**	

Outliers distort trend and seasonality extracted using moving averages





STL

STL: A Seasonal-Trend Decomposition Procedure Based on Loess

Robert B. Cleveland,[1] William S. Cleveland,[2] Jean E. McRae,[2] and Irma Terpenning[2]

Abstract: STL is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components. STL has a simple design that consists of a sequence of applications of the loess smoother; the simplicity allows analysis of the properties of the procedure and allows fast computation, even for very long time series and large amounts of trend and seasonal smoothing. Other features of STL are specification of amounts of seasonal and trend smoothing that range, in a nearly continuous way, from a very small amount of smoothing to a very large amount; robust estimates of the trend and seasonal components that are not distorted by aberrant behavior in the data; specification of the period of the seasonal component to any integer multiple of the time sampling interval greater than one; and the ability to decompose time series with missing values.

Key words: Seasonal adjustment; time series; loess.

Cleveland, Robert B., et al. "STL: A seasonal-trend decomposition." *Journal of official statistics* 6.1 (1990): 3-73.

The main idea

- Assume: $y(t) = trend(t) + seasonal(t) + residual(t)$
- Want a way to calculate each individual component
- STL does this using LOESS

The main idea

Outer loop

Compute robustness weights from residuals used by LOESS in inner loop to protect against outliers

Inner loop

Compute estimate of trend(t) and seasonal(t) using LOESS

Compute residuals
residual(t) = y - trend(t) – seasonal(t)

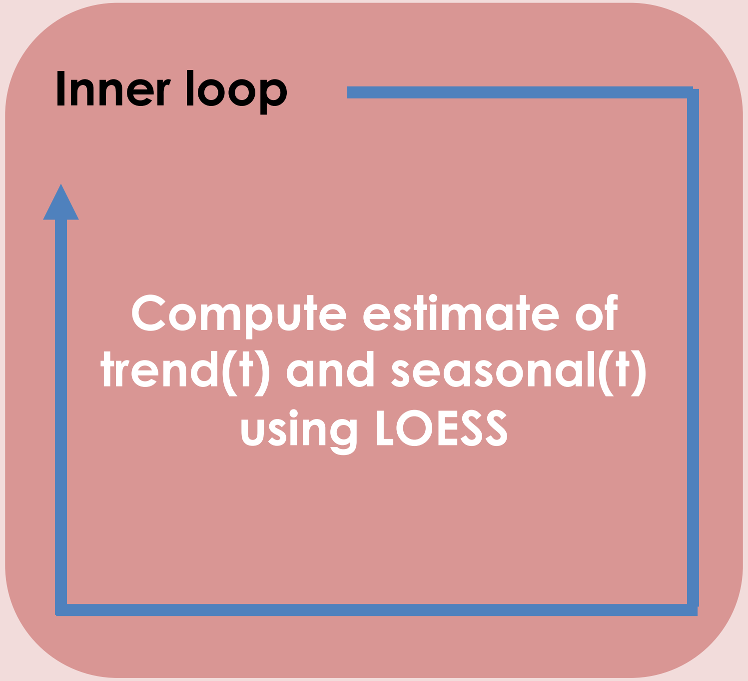
Inner loop summary

ITERATE

1. **De-trend the data**: y(t) – trend(t). For the first iteration assume trend(t) = 0

2. **Extract seasonal(t)** using LOESS on a set of time series derived from y_detrend(t) called cycle-subseries.

3. **Extract trend(t)** by subtracting seasonal(t) from y(t) and smoothing using LOESS



Outer loop summary

Outer loop

ITERATE

1. **Extract seasonal(t) and trend(t)** from inner loop

2. **Compute residuals:**
   y(t) – trend(t) – seasonal(t)

3. **Compute weights from residuals** ρ_t to pass to LOWESS in the inner loop . This is to down-weight outliers.

Inner loop

Compute estimate of trend(t) and seasonal(t) using LOESS

Implementation

statsmodels.tsa.seasonal.STL ¶

class `statsmodels.tsa.seasonal.STL`(*endog, period=None, seasonal=7, trend=None, low_pass=None, seasonal_deg=0, trend_deg=0, low_pass_deg=0, robust=False, seasonal_jump=1, trend_jump=1, low_pass_jump=1*)

Season-Trend decomposition using LOESS.

Parameters

endog : array_like

Data to be decomposed. Must be squeezable to 1-d.

period : {int, None}, **optional**

Periodicity of the sequence. If None and endog is a pandas Series or DataFrame, attempts to determine from endog. If endog is a ndarray, period must be provided.

seasonal : int, **optional**

Length of the seasonal smoother. Must be an odd integer, and should normally be >= 7 (default).

trend : {int, None}, **optional**

Length of the trend smoother. Must be an odd integer. If not provided uses the smallest odd integer greater than 1.5 * period / (1 - 1.5 / seasonal), following the suggestion in the original implementation.

low_pass : {int, None}, **optional**

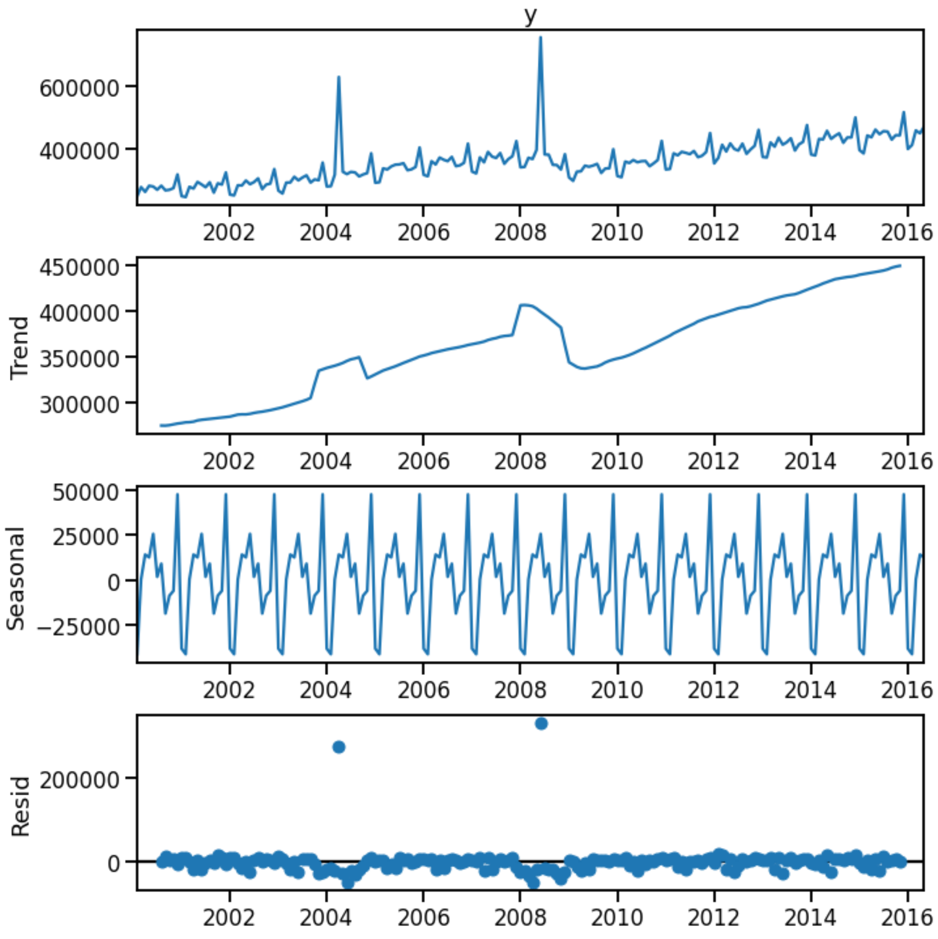
Length of the low-pass filter. Must be an odd integer >=3. If not provided, uses the smallest odd integer > period.

```python
# Perform STL decomposition
res = STL(
    endog=df["y"],  # Y values
    period=12,  # The periodicity of the seasonal component
    seasonal=7,  # Determines the window size for LOESS used
    # when smoothing the seasonal component
    # (i.e, the cycle-subseries)
    robust=True  # Flag to use robust regression when
    # fitting the LOESS curves so the fit
    # is robust to outliers
).fit()
```

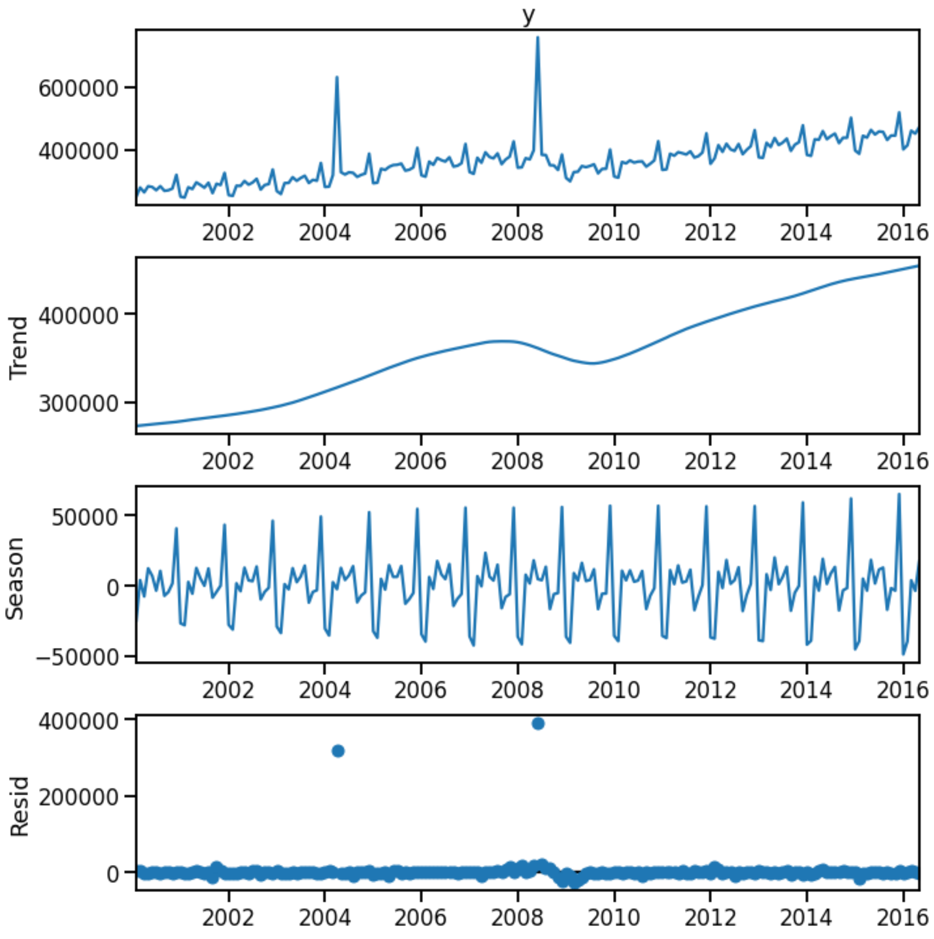
```python
# Extract the trend and seasonality individually
df['trend'] = res.trend
df['seasonal'] = res.seasonal
df.head()
```

ds	y	trend	seasonal
1992-01-01	146376	163507.268049	-18105.934264
1992-02-01	147079	164270.425274	-19220.032341
1992-03-01	159336	165042.317260	-5396.563294
1992-04-01	163669	165822.245976	-885.576431
1992-05-01	170068	166609.427740	2929.056385

Example with outliers



Classical decomposition

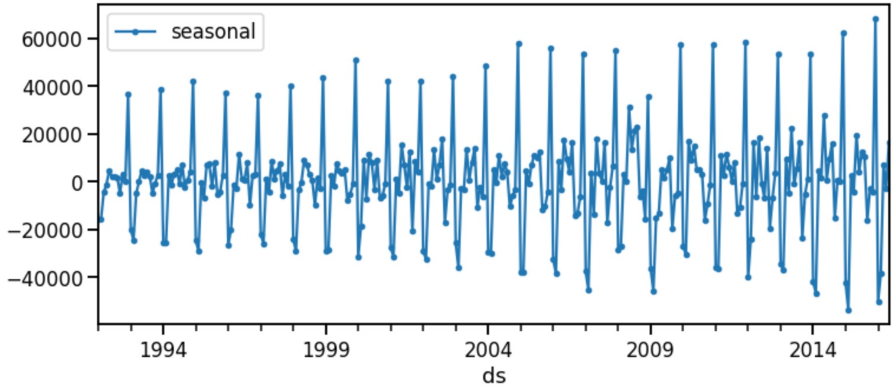


STL decomposition

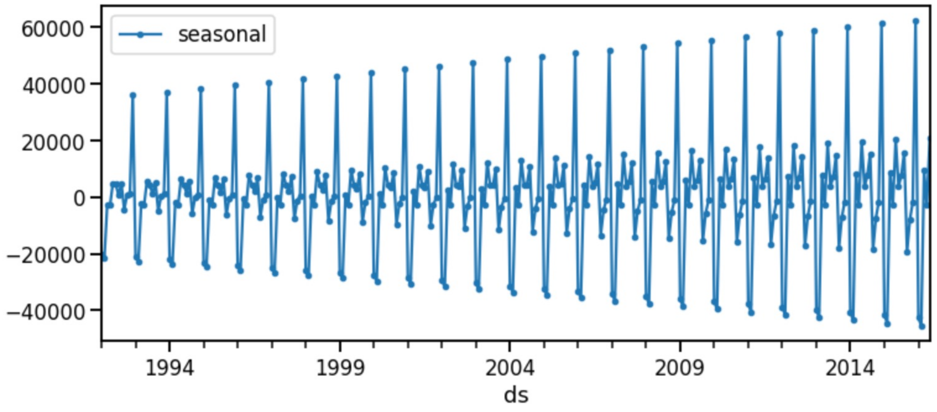
Most important parameters

- n_s (seasonal): Determines smoothness and uniformity of seasonal component. Default value of 7 typically good enough.

$n_s = 3$



$n_s = 101$

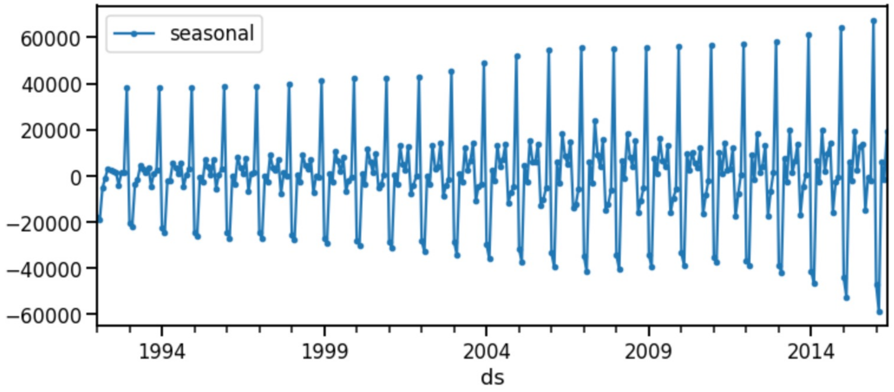


Most important parameters

- n_s (seasonal): Determines smoothness and uniformity of seasonal component. Default value of 7 typically good enough.

- n_p (period): The seasonal period (e.g., for daily data and weekly seasonality this would be 7). Normally determined by the use case

n_s = 7



How to handle multiplicative time series

- STL assumes the time series is additive
- If time series is multiplicative use log transform to make it additive

$$\log y(t) = \log\ (\text{trend}(t) \times \text{seasonal}(t) \times \text{residual}(t))$$
$$\log y(t) = \log \text{trend}(t) + \log \text{seasonal}(t) + \log \text{residual}(t)$$

- Transform back to original scale by taking inverse log transform

Summary

STL extracts the seasonality and trend iteratively using LOESS

STL is robust to outliers

There are two main parameters to set in practice, the remaining default parameters are normally sufficient

The seasonal component can vary in time and is not necessarily periodic