Wrap up

Time series decomposition

What is Time Series Decomposition?

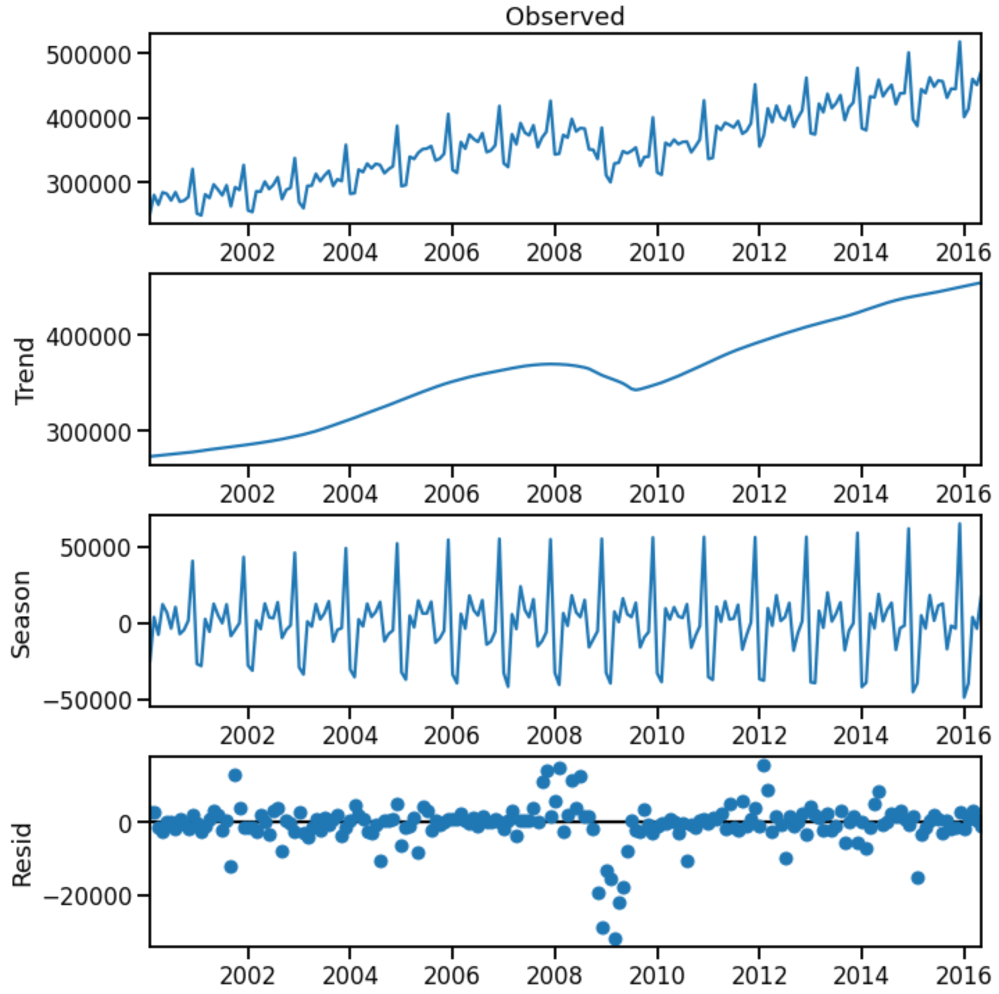
- Breaking a time series down into components: trend, seasonality, residual

- Additive:

y(t) = trend(t) + seasonal (t) + residual(t)

- Multiplicative:

y(t) = trend(t) x seasonal (t) x residual(t)



Why is decomposition useful?

Exploratory data analysis: To answer questions such as "what was the impact of an ad campaign once we account for seasonality?"

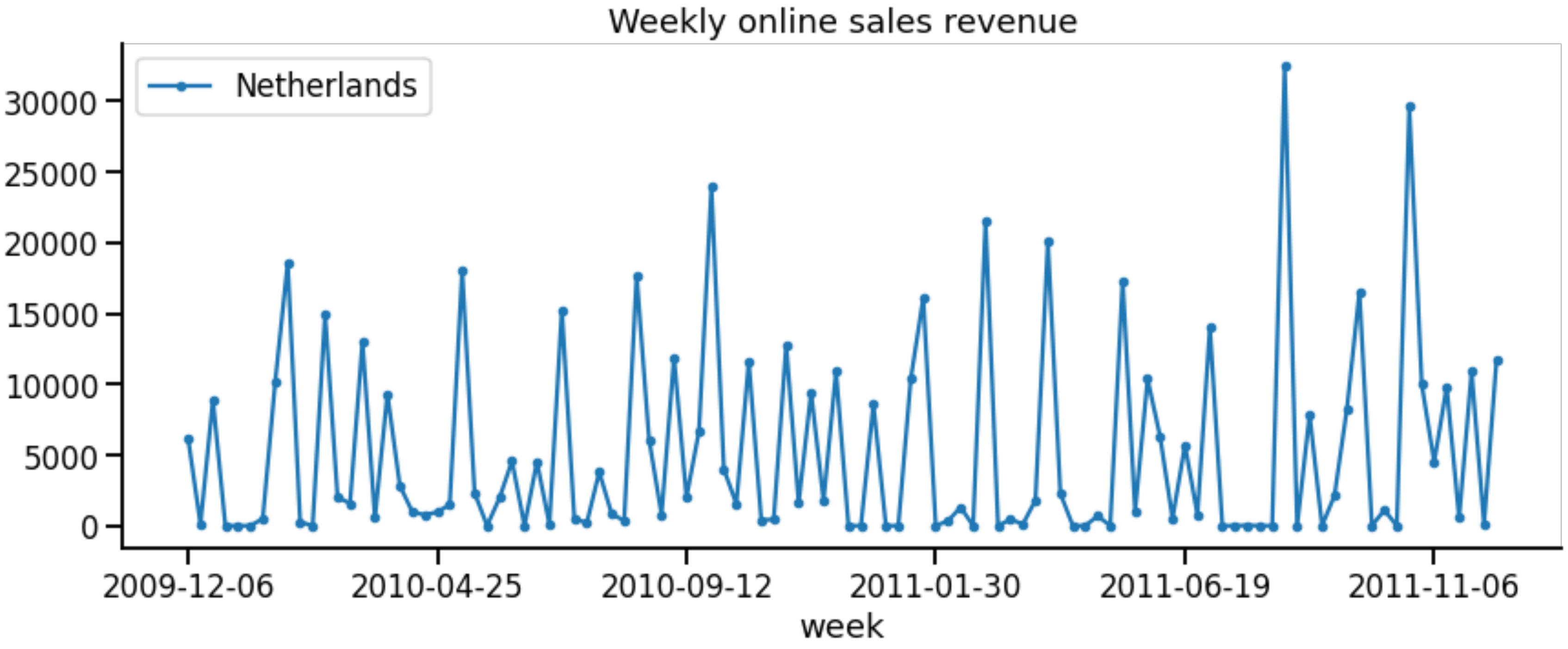
Pre-processing: Useful for identifying outliers and can be used to impute outliers and missing data

Feature engineering: Can derive features from the components to use as inputs in machine learning models

Forecasting: Forecast the components and aggregate to produce the final forecast

Decomposition is not always possible

- Not all time series can be easily broken down into components



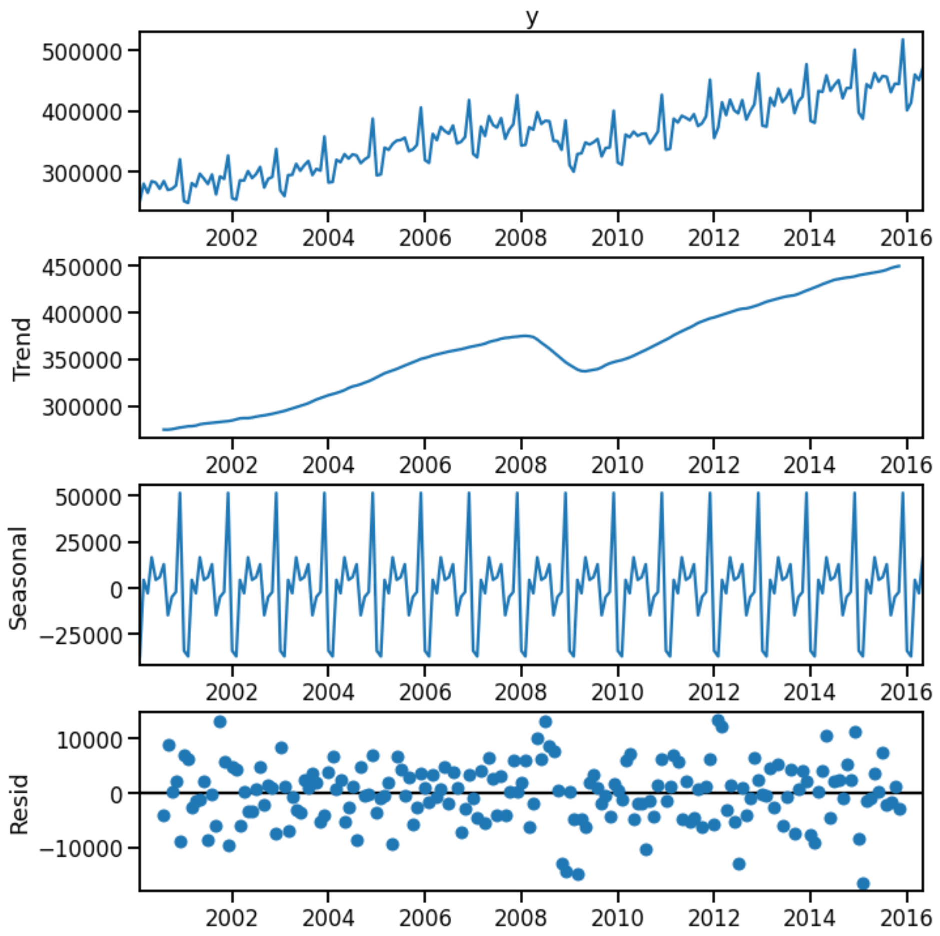
Classical Decomposition

1. Identify order of seasonality T

2. Compute trend using moving averages. 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)



Cycle-subseries

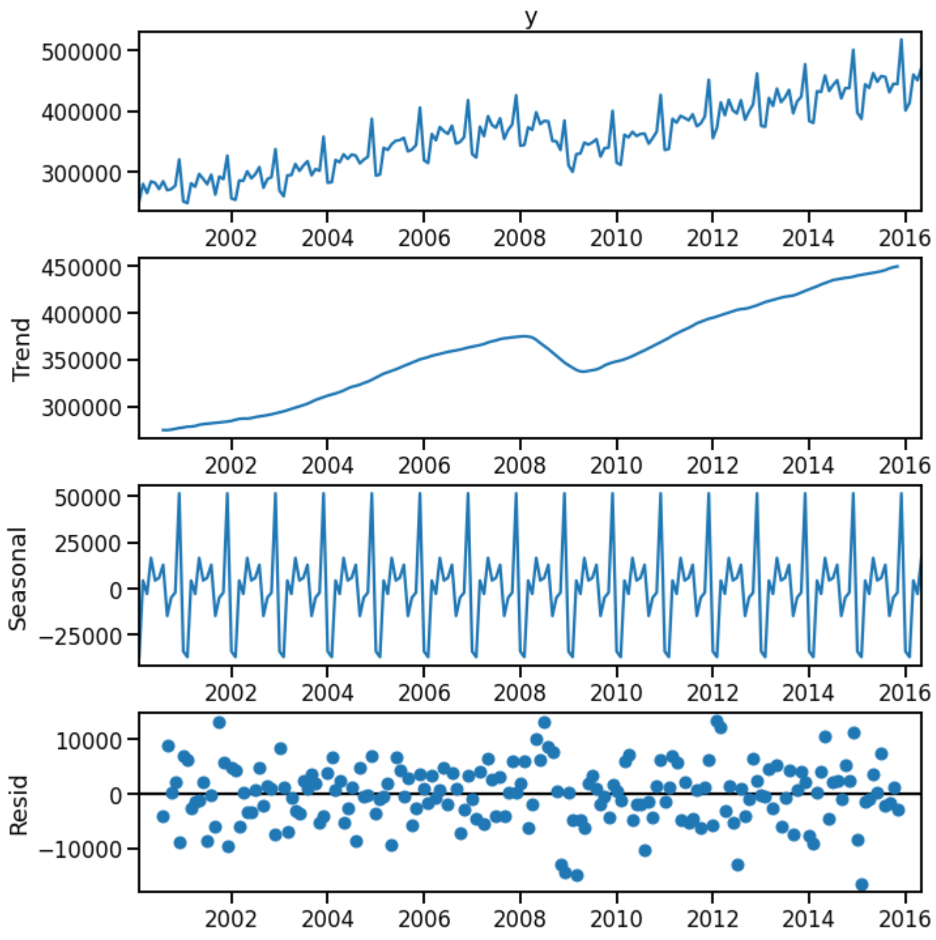
Year-Month	Y_detrended
2011-Jan	112
2011-Feb	146
2011-Mar	80
2011-Apr	90
…	…

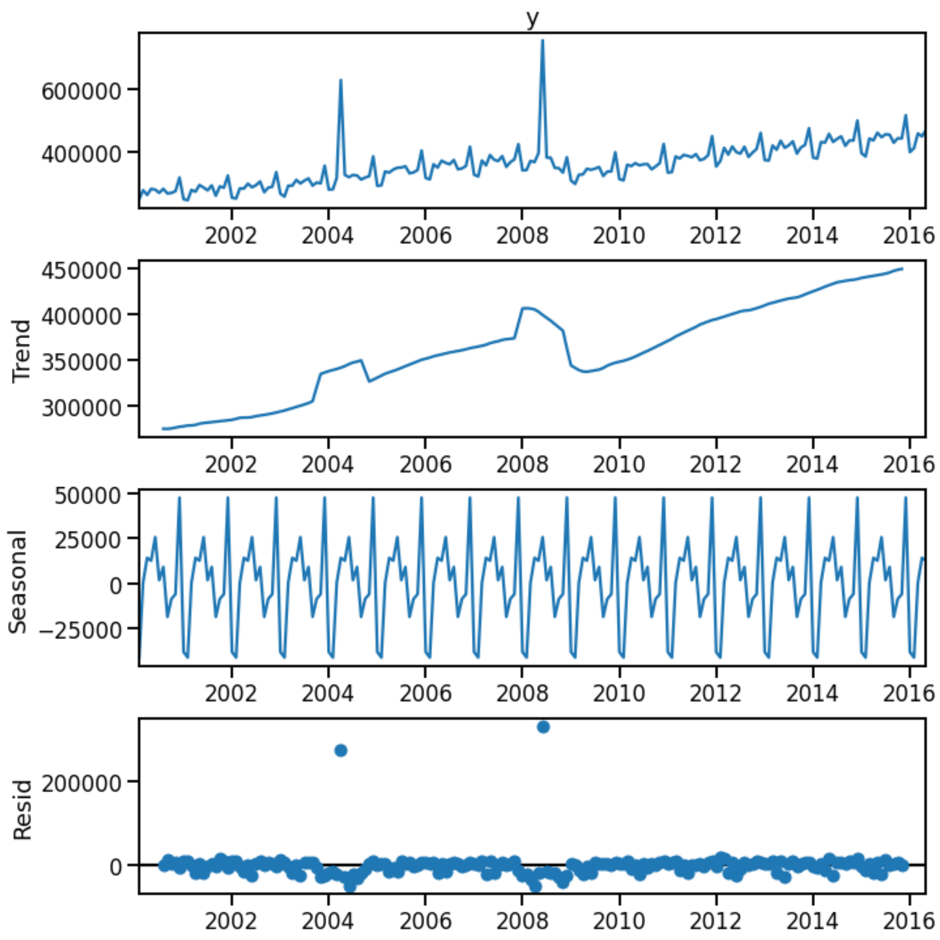
Note relation to classical decomposition for seasonality:

1) De-trend original time series
2) Average over seasonal index

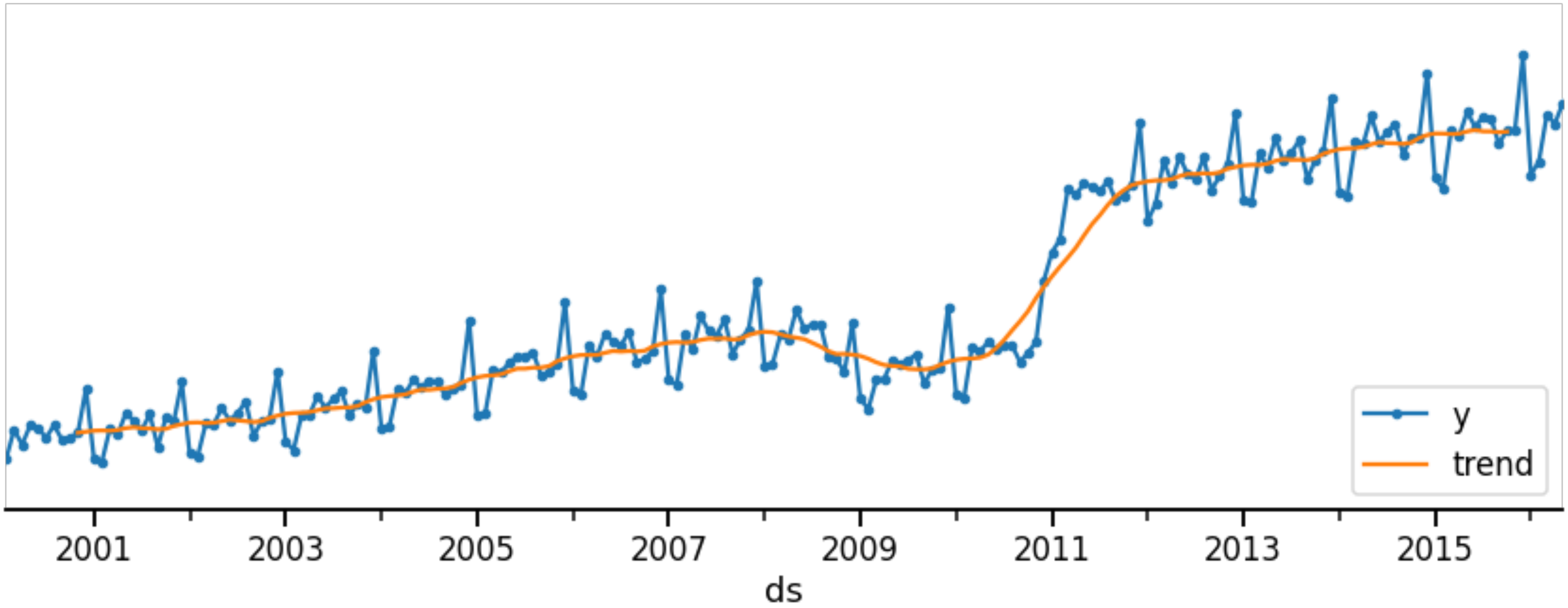
Month/Year	2011	2012	2013	…	Mean
Jan	112	134	156	…	130
Feb	146	145	151	…	148
Mar	80	85	86	…	82
Apr	90	93	98	…	94
…	…	…	…	…	…

Limitations: Outliers, edges, rapid changes



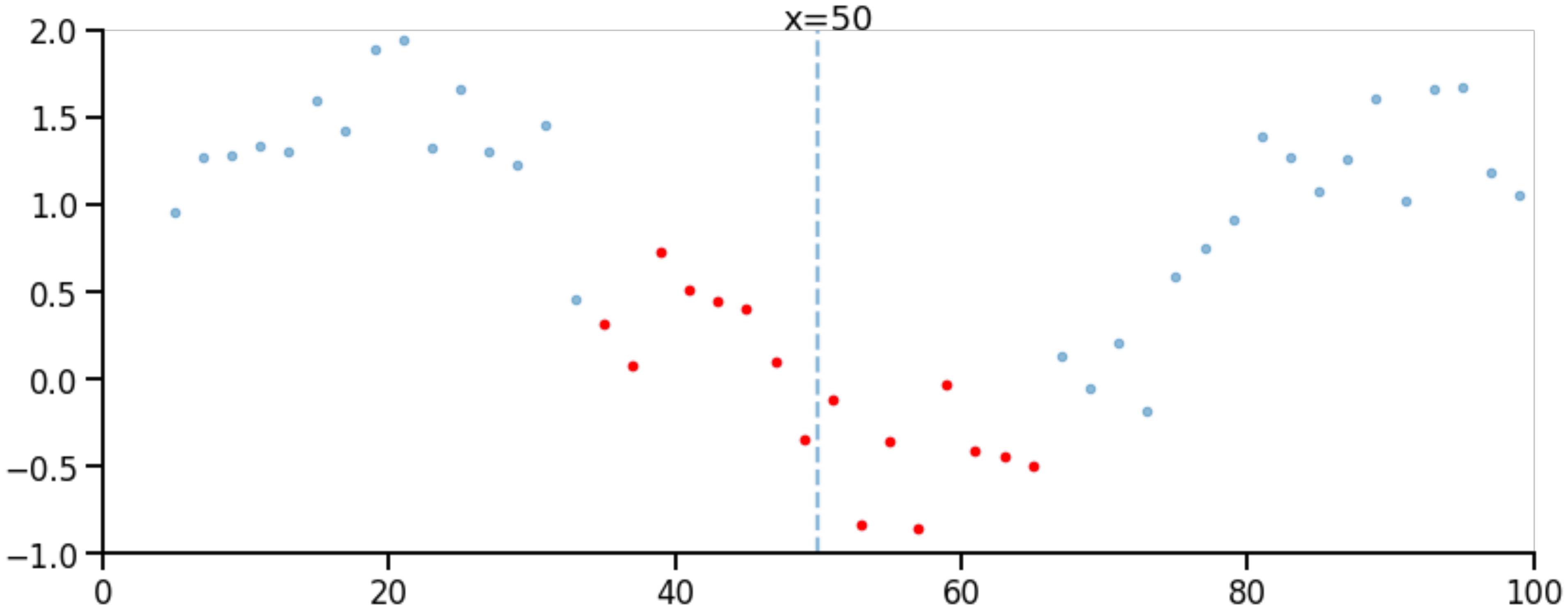


Limitations: Outliers, edges, rapid changes



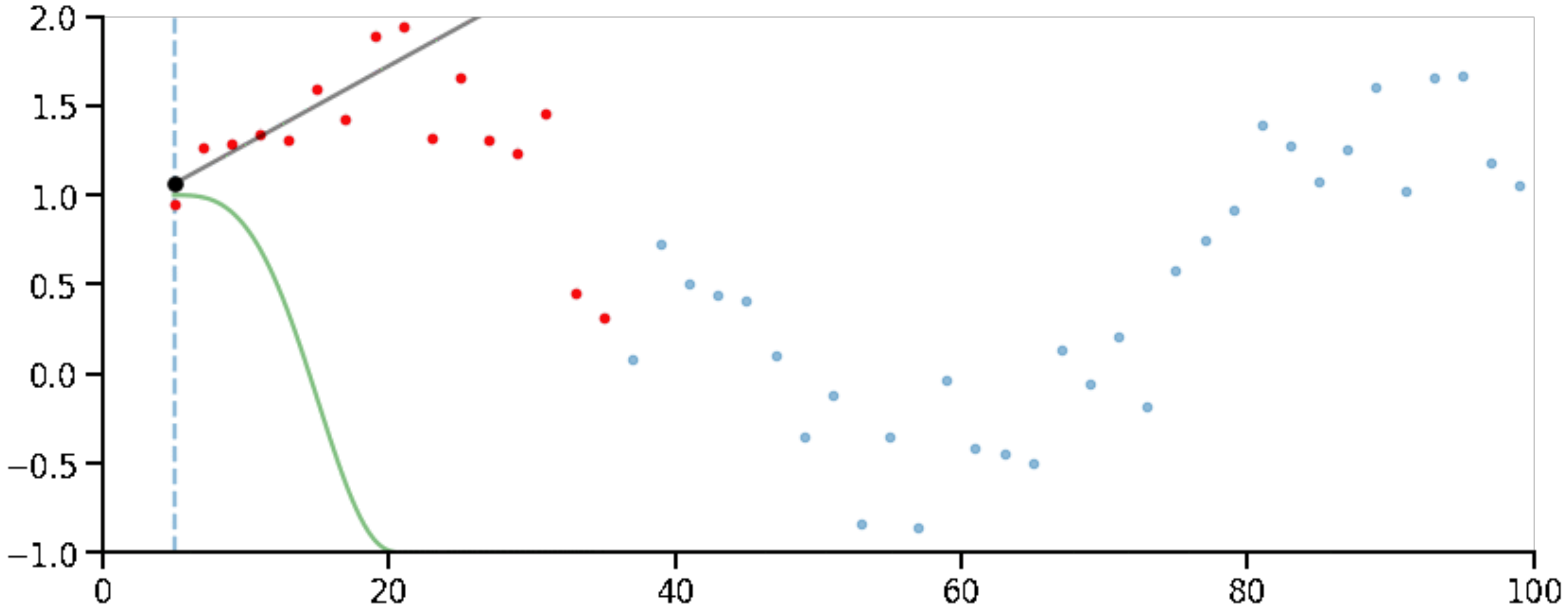
LOWESS

- Fit a weighted robust linear regression to this subset of the data

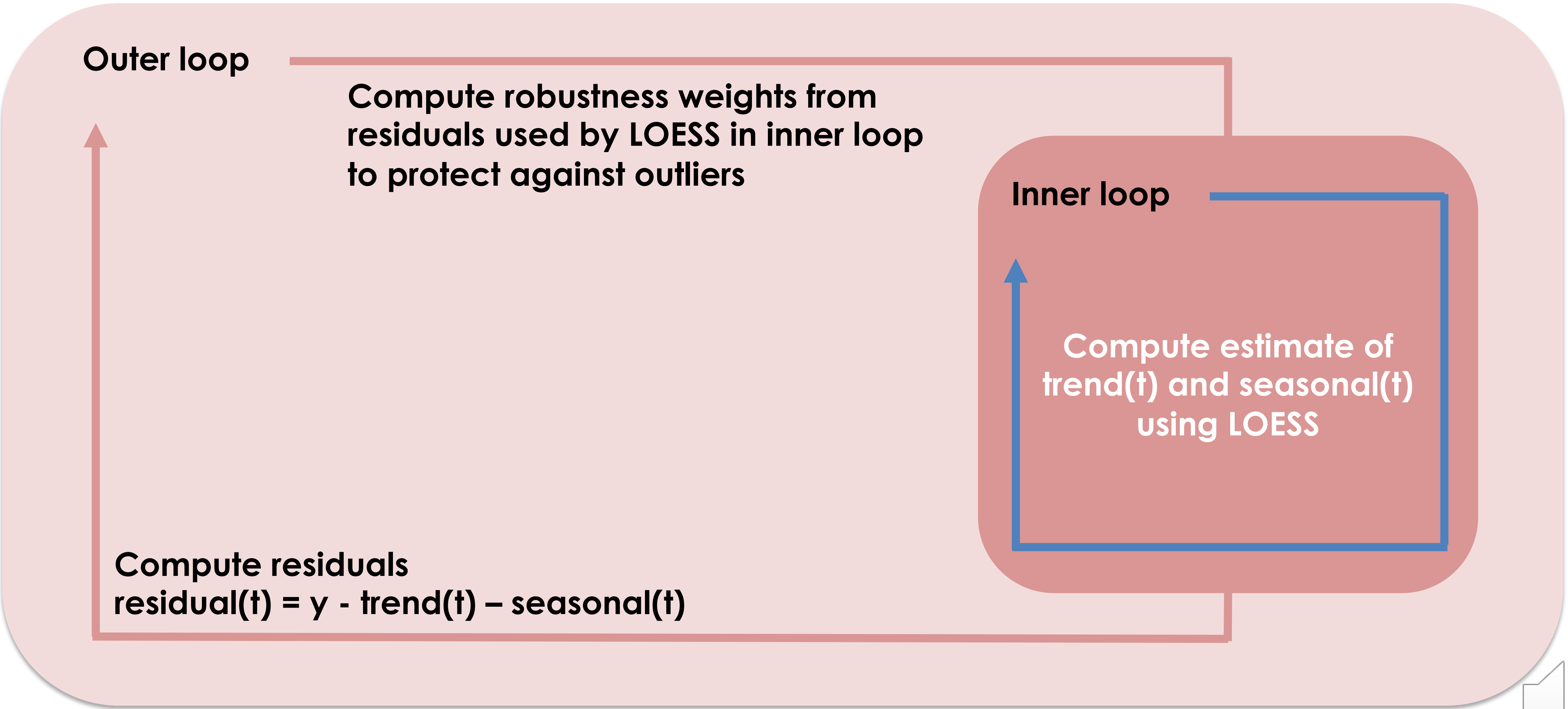


LOWESS

- Evaluate the same process across many x values to obtain a smooth fit



Seasonal and Trend Decomposition using LOESS (STL)



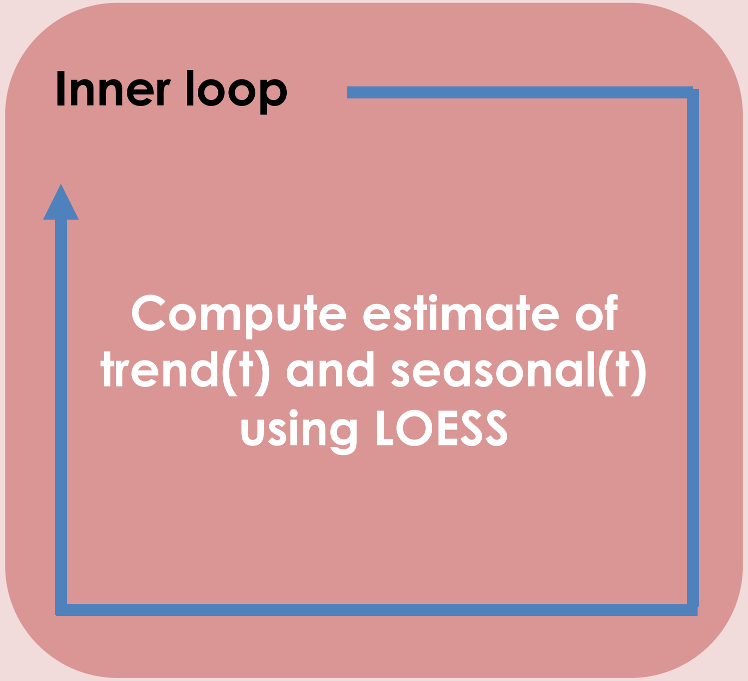
Inner loop summary

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.



Cycle-subseries

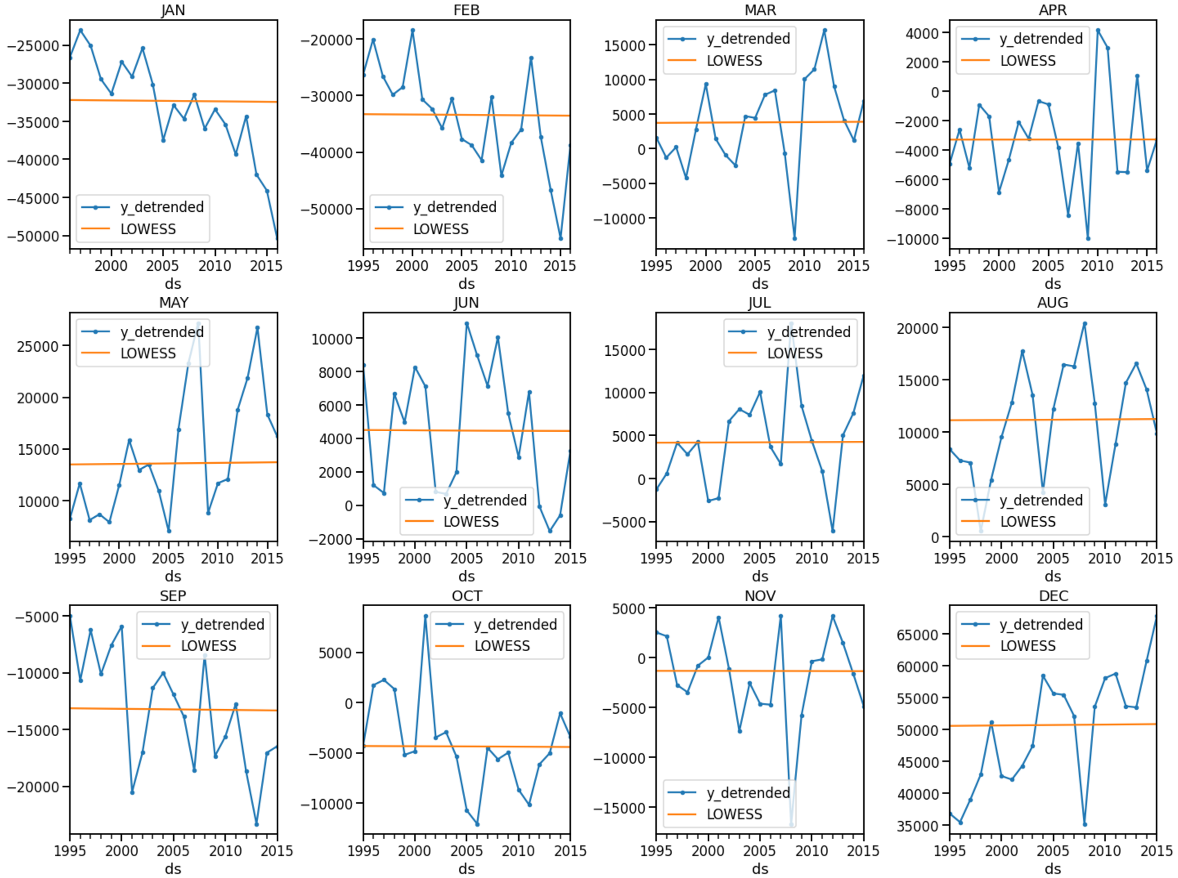
Year-Month	y
2011-Jan	112
2011-Feb	146
2011-Mar	80
2011-Apr	90
…	…

Note relation to classical decomposition for seasonality:

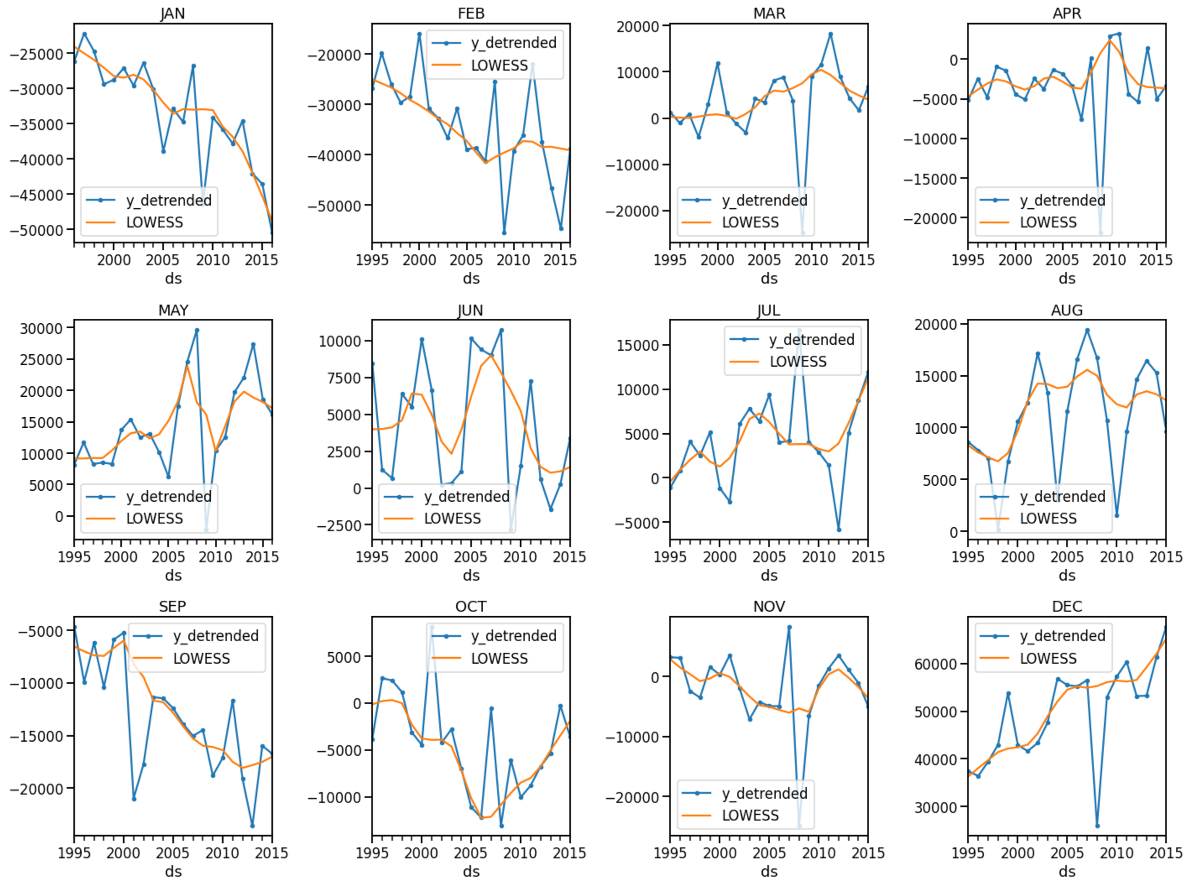
1) De-trend original time series
2) Average over seasonal index

Month/Year	2011	2012	2013	…	Mean
Jan	112	134	156	…	130
Feb	146	145	151	…	148
Mar	80	85	86	…	82
Apr	90	93	98	…	94
…	…	…	…	…	…

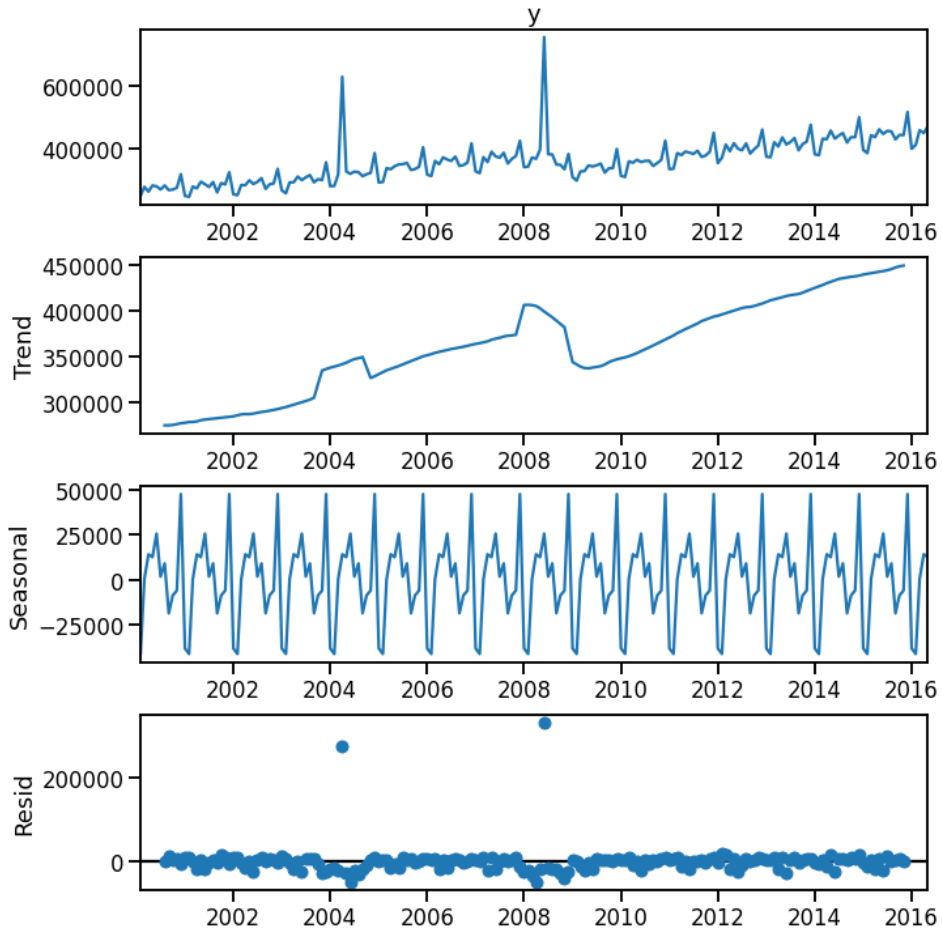
Cycle-subseries: Classical Decomposition



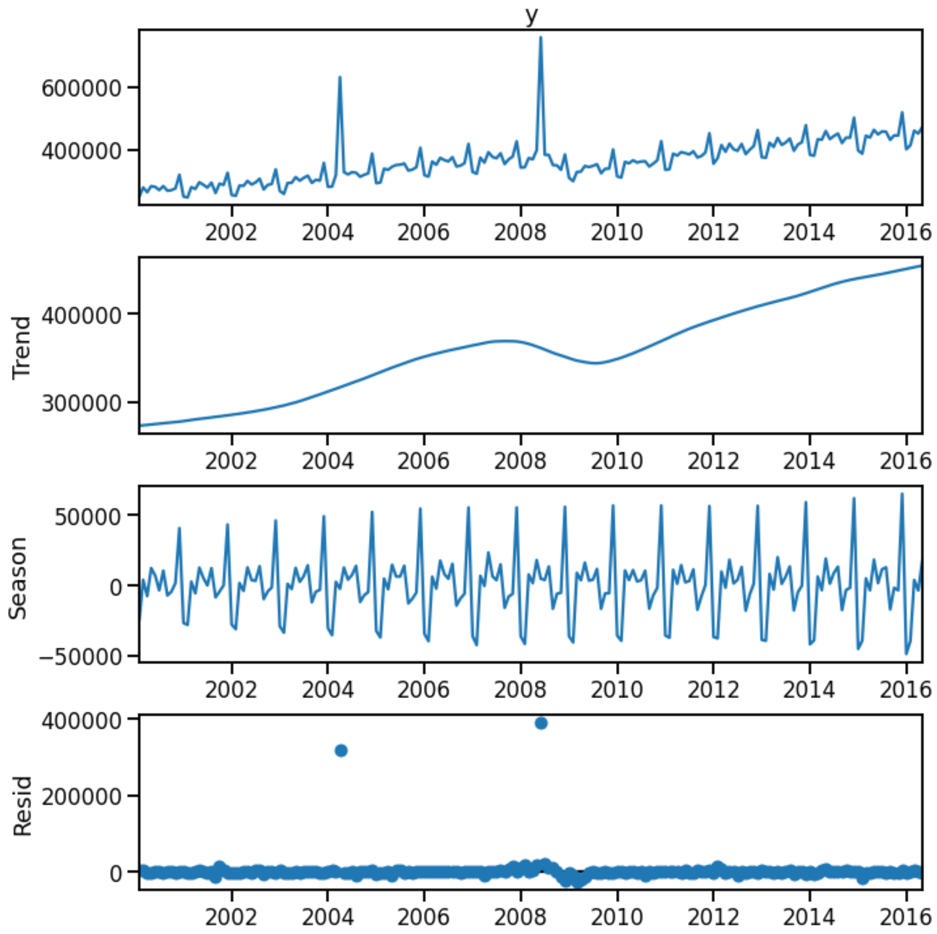
Cycle-subseries: STL



Example with outliers



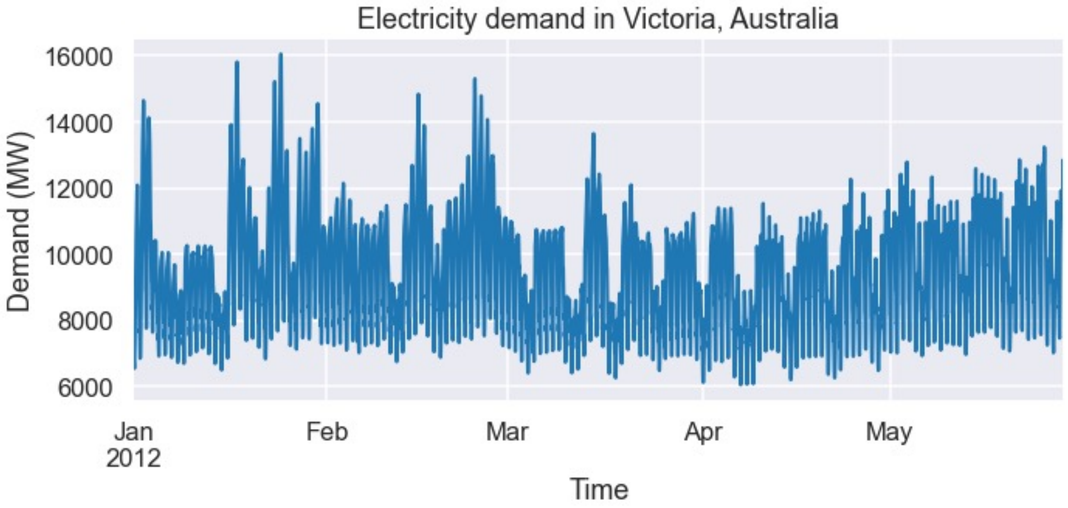
Classical decomposition



STL decomposition

Time series can have multiple seasonalities

- This time series of electricity demand has daily, weekly, and yearly seasonality.
- In general, as data becomes more granular we start to see more seasonal patterns.
- We want to be able to decompose it into a trend and multiple seasonal components.



$$y_t = \underbrace{\hat{T}_t}_{\text{Trend component}} + \underbrace{\hat{S}_t^{(1)} + \hat{S}_t^{(2)} + \cdots + \hat{S}_t^{(N)}}_{\text{Seasonal components}} + \underbrace{\hat{R}_t}_{\text{Residual component}}$$

MSTL: What is it?

- **M**utliple **S**easonal-**T**rend decomposition using **L**oess (MSTL).

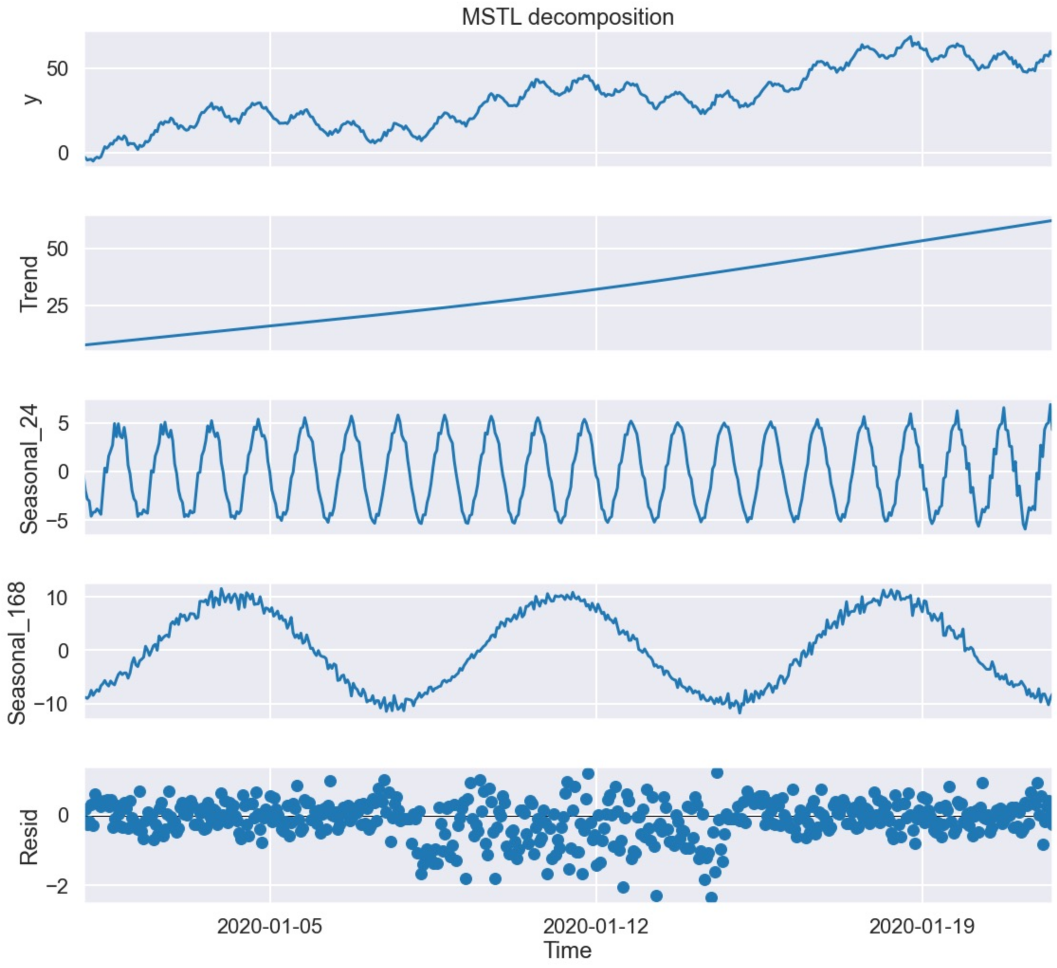
- A method to decompose a time series into a trend component, multiple seasonal components, and a residual component by repeatedly applying STL.

$$y_t = \underbrace{\hat{T}_t}_{\text{Trend component}} + \underbrace{\hat{S}_t^{(1)} + \hat{S}_t^{(2)} + \cdots + \hat{S}_t^{(N)}}_{\text{Seasonal components}} + \underbrace{\hat{R}_t}_{\text{Residual component}}$$

- MSTL, like STL, assumes the time series can be broken into an additive decomposition.

- Transform the time series (e.g., using Box Cox) if it is not additive.

MSTL: What is it?



What if you have long or many time series?

- LOWESS and STL are slow to compute as they require fitting a regression model and making predictions.

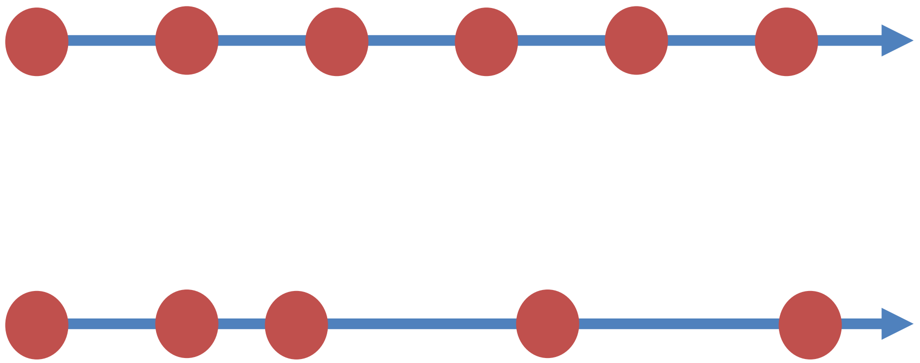
- For long time series it may be faster to use classical decomposition and accept trade off with impact from outliers.

- Alternatively use moving median or other filtering methods for trend extraction.

- Depends on available computational resource and use case.

Regular vs Irregularly spaced time series

- Many methods assume evenly spaced (i.e., regular) time series (e.g., STL).
- Some methods can handle irregularly spaced time series (e.g., LOWESS).
- Many methods assume regular time series.
- Potential solution is to convert the irregularly timeseries into a regular one (e.g., through aggregating).



Summary

- Time series can be decomposed into components: Trend, Seasonal, and Residual.
- Classical decomposition using moving averages, LOWESS, and STL are useful methods to decompose a time series.
- Decomposition is used in multiple ways including EDA, feature engineering, preprocessing, and forecasting.