Seasonal decomposition and interpolation
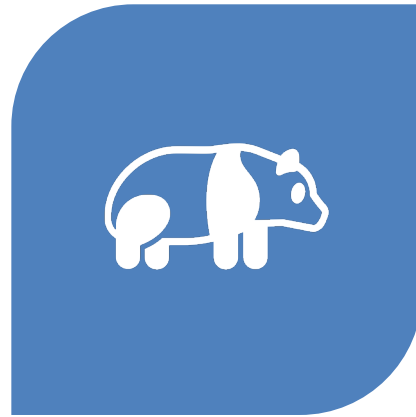
Missing data

Contents

SEASONAL DECOMPOSITION AND INTERPOLATION
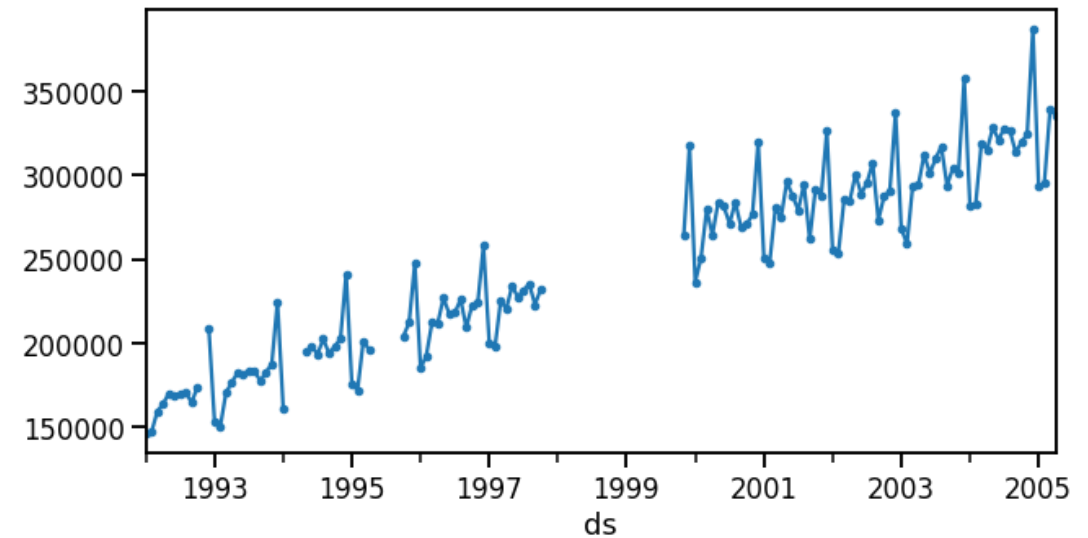
IMPLEMENTATION IN PANDAS

PRACTICAL CONSIDERATIONS
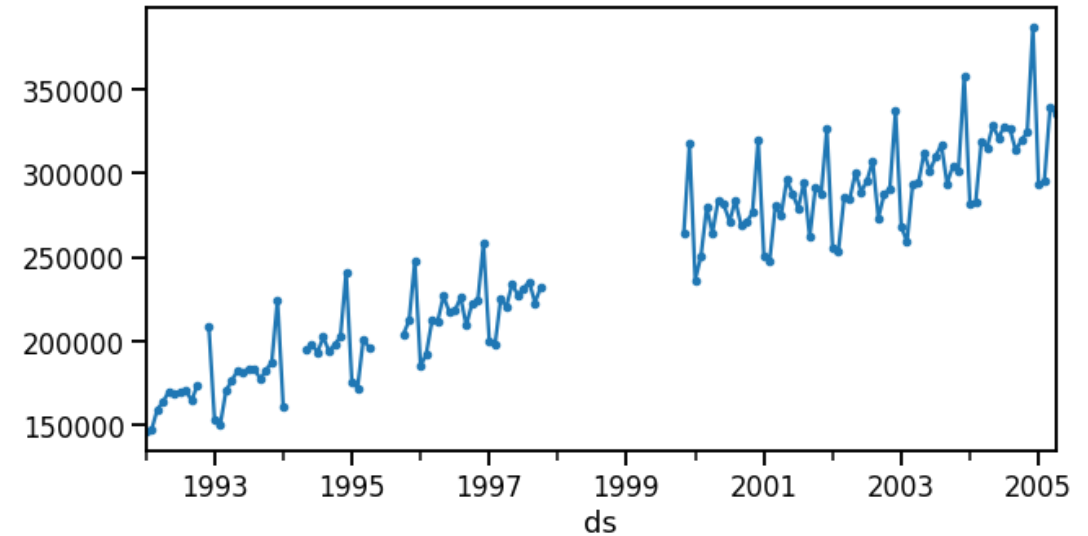
Seasonal decomposition and interpolation

- Interpolate with a simple method
- Y = trend + seasonality + noise
- Subtract seasonality
- Interpolate missing data on de-seasoned data
- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- **Interpolate with a simple method**
- Y = trend + seasonality + noise
- Subtract seasonality
- Interpolate missing data on de-seasoned data
- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- **Interpolate with a simple method**
- Y = trend + seasonality + noise
- Subtract seasonality
- Interpolate missing data on de-seasoned data
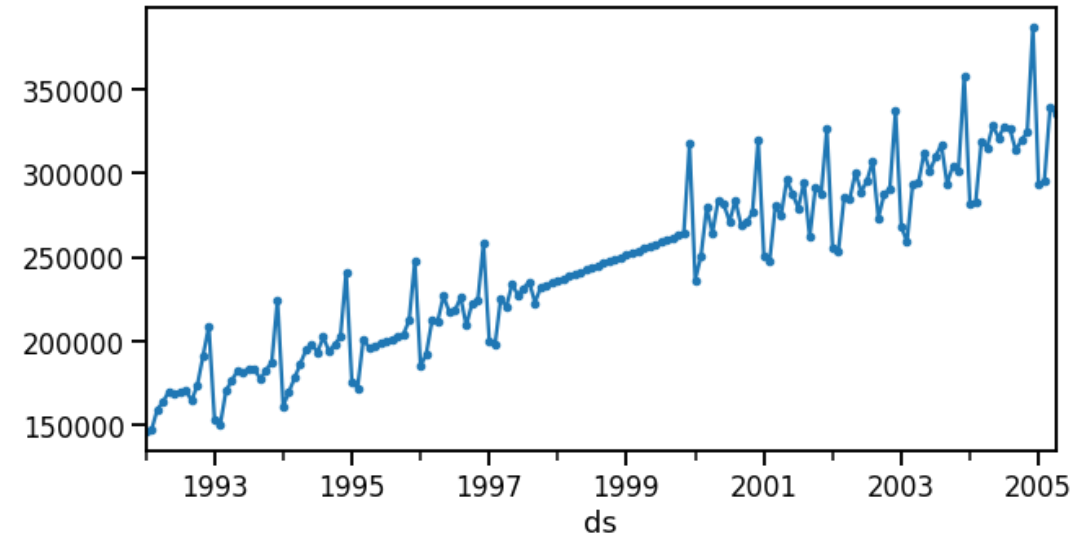- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- Interpolate with a simple method
- **Y = trend + seasonality + noise**
- Subtract seasonality
- Interpolate missing data on de-seasoned data
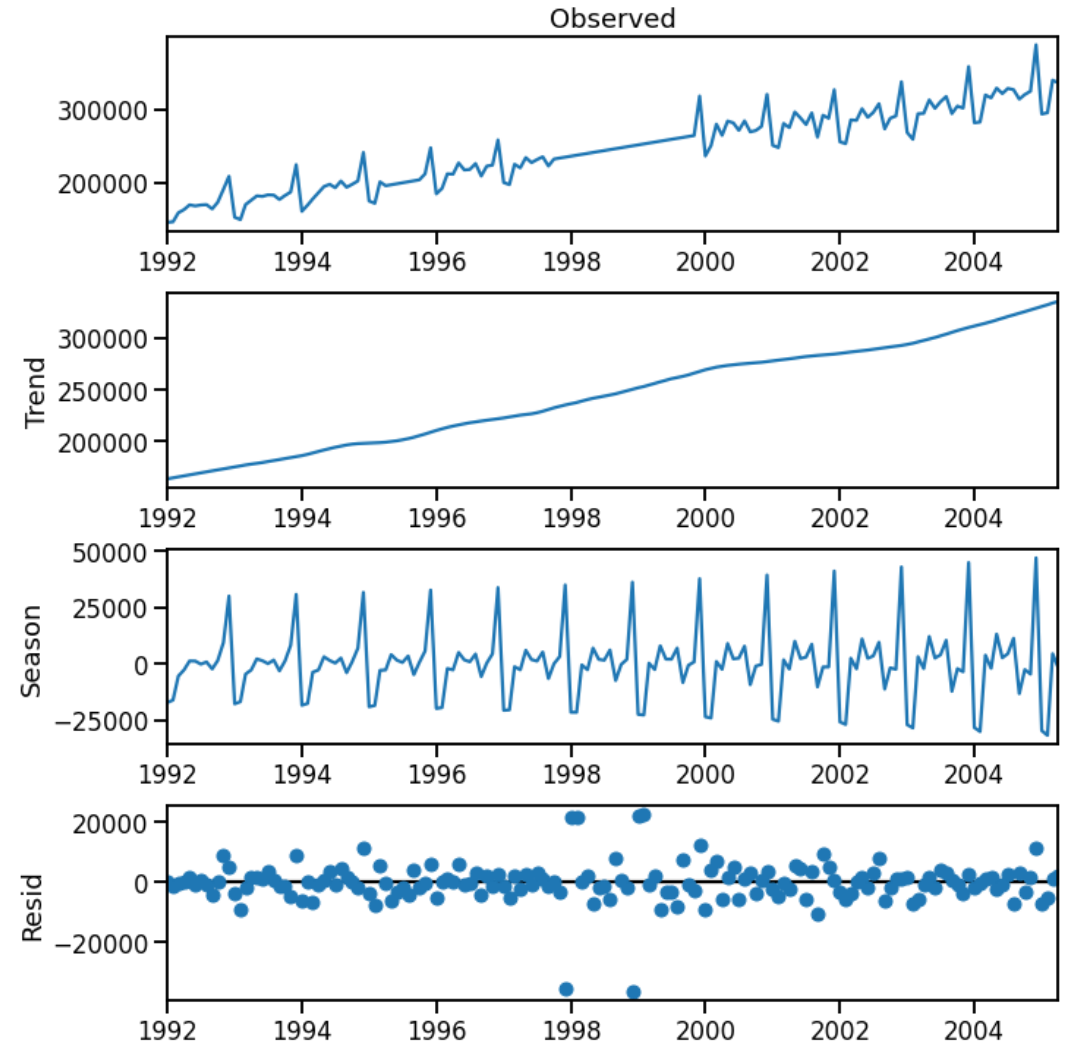- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- Interpolate with a simple method
- Y = trend + seasonality + noise
- **Subtract seasonality**
- Interpolate missing data on de-seasoned data
- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- Interpolate with a simple method
- Y = trend + seasonality + noise
- Subtract seasonality
- **Interpolate missing data on de-seasoned data**
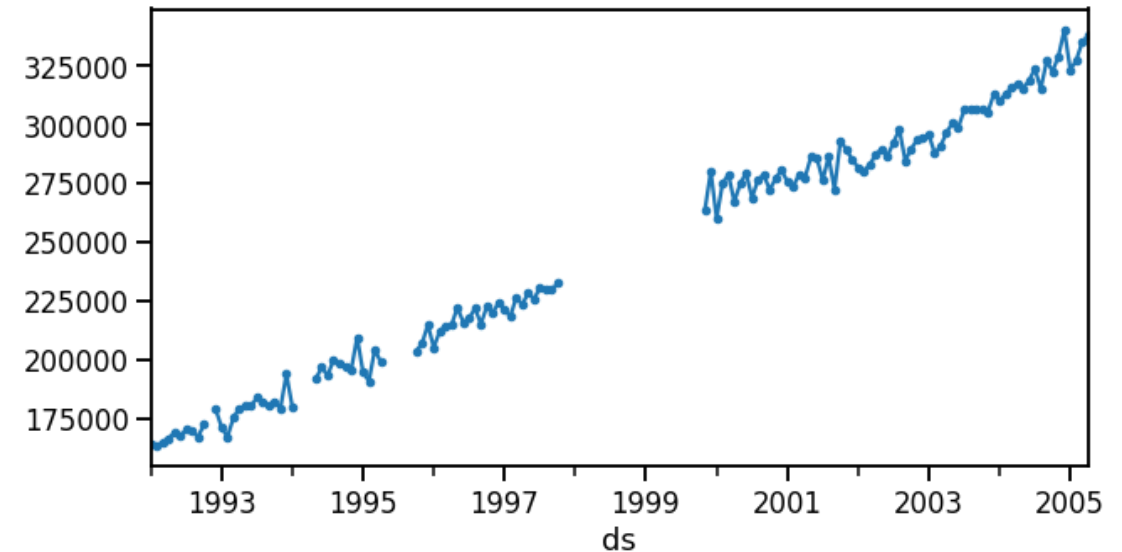- Add seasonality back to de-seasoned data

Seasonal decomposition and interpolation

- Interpolate with a simple method
- Y = trend + seasonality + noise
- Subtract seasonality
- Interpolate missing data on de-seasoned data
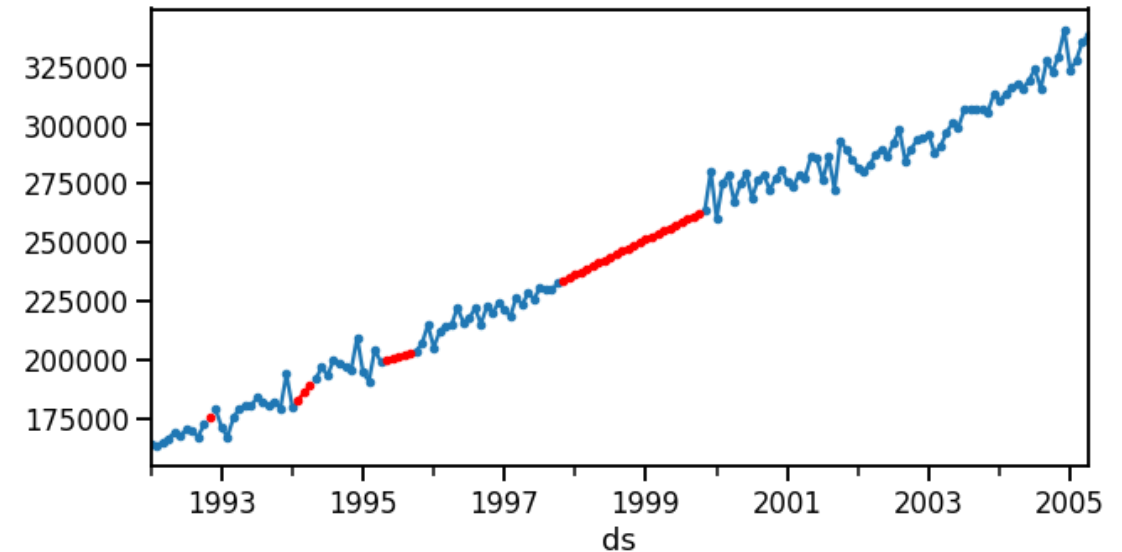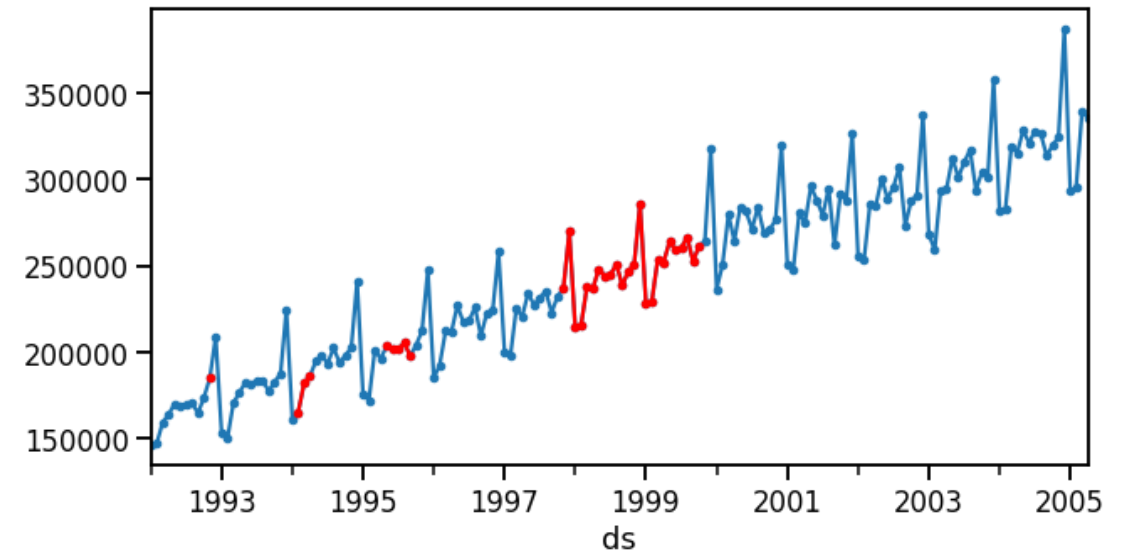- **Add seasonality back to de-seasoned data**

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.

```python
# STL decomposition on interpolated data
res = STL(df.interpolate(method='linear')).fit()

# Extract seasonal component
seasonal_component = res.seasonal

# De-seasonlise original data
df_deseasonalised = df['y'] - seasonal_component

# Perform linear interpolation on de-seasonalised data
df_deseasonalised_imputed = df_deseasonalised.interpolate(method='linear')

# Add seasonal component back to get the final imputed time series
df_imputed = df_deseasonalised_imputed + seasonal_component
```

Paper: Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. *Journal of official statistics*, 6(1), 3-73.

Practical tips

- Consider how the method will distort the time series

  - Does the method distort seasonality or long term trends?

  - Does the method create artificial jumps in the data?

- Small gaps: forward fill or linear interpolation

- Larger gaps: consider structure of time series

  - No trend or seasonality: Forward fill, linear interpolation, mean

  - Strong trend and no seasonality: linear interpolation or splines

  - Strong seasonality: Seasonal decomposition and interpolation

- Sense check time series plots after interpolation