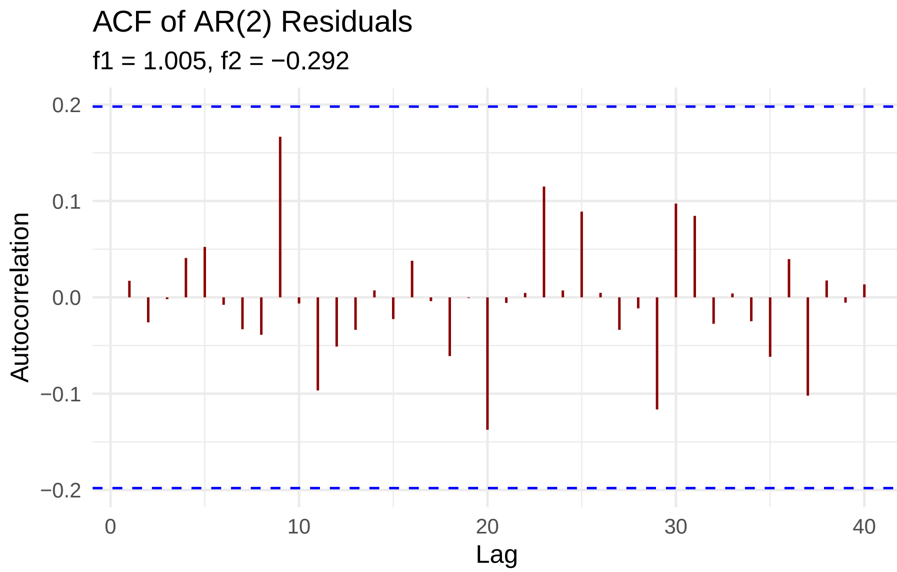

We found residual variance ($\sigma^2 = 0.4571$) to be lesser than before and also from the ACF, we can see no autocorrelation. The fitted model:

$$Y_t = 1.005Y_{t-1} - 0.292Y_{t-1} + Z_t, \quad Z_t \sim IID(0, 0.457)$$

$\phi_1 = 1.005$ indicates strong persistence from the immediate past and $\phi_2 = -0.292$ indicates negative correction for over-persistence, introducing mean reversion. All lags within confidence bounds confirmed noise independence.

The Lake Huron residuals were best modeled by an AR(2) process, which reduced residual variance by 11.8% compared to AR(1) and also eliminated significant autocorrelation in residuals.

Estimation and Elimination of Trend and Seasonal Components

The first step in time series analysis is **visual inspection** of the data to identify structural breaks, outliers, or patterns. If trends or seasonality are present, the **classical decomposition model** is often applied:

$$X_t = m_t + s_t + Y_t$$

where m_t is a slowly varying **trend component**, s_t is a periodic **seasonal component** with known period d, and Y_t represents stationary **noise**. This decomposition isolates systematic patterns from random fluctuations, enabling focused analysis of each component (e.g., modeling trends separately or validating stationarity in residuals).

Estimation and Elimination of Trend in the Absence of Seasonality

A non-seasonal model with trend will be of form:

$$X_t = m_t + Y_t, \quad t = 1, 2, \cdots$$

m_t is the trend component and Y_t is zero-mean noise.

Trend Estimation

Smoothing with a finite moving average filter:

A **moving-average filter** is the simplest sort of "low-pass" smoother: it replaces each point in your series by the average of its neighbors over a fixed window, thereby stripping away the rapidly-fluctuating (high-frequency) component and leaving us with a slowly-varying estimate of the trend. Here, we will see **two-sided** moving-average filter.

The two-sided moving average for a time series X_t is defined as:

$$\hat{m}_t = \frac{1}{2q+1} \sum_{j=-q}^{q} X_{t-j}, \quad q+1 \leq t \leq n-q$$

where q determines the window size (e.g., $q = 2$ uses 5 terms: $X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}$).

For $t \leq q$, we use $X_t := X_q$ and for $t \geq n - q$, we use $X_t := X_n$. The trend m_t is approximately linear over the window $[t - q, t + q]$. The noise Y_t averages to zero within the window.

By averaging values within a window of size $2q + 1$, rapid fluctuations (high frequencies) cancel out due to their randomness. Slow-moving trends (low frequencies) remain largely unaffected if they are approximately linear or polynomial over the window.

A moving average (MA) can be expressed as a **linear filter** or **linear operator** by defining it as a weighted sum of the time series observations. Mathematically, this is represented using a **convolution** of the time series with a set of filter weights. We can think of $\hat{m}_t$ defined above as a linear filter that applies a set of weights $\{a_j\}$ o lagged values of the time series $\{X_t\}$ to produce a smoothed series $\{\hat{m}_t\}$:

$$\hat{m}_t = \sum_{j=-\infty}^{\infty} a_j X_{t-j}$$

Here, the weights a_j are non-zero only over a finite window $[-q, q]$, making it a **finite impulse response (FIR) filter**.

For symmetric moving average, we consider weights to be equal.

$$a_j = (2q+1)^{-1}, \quad |j| \leq q$$

For large q, provided $(2q+1)^{-1} \sum_{j=-q}^{q} Y_{t-j} \approx 0$ it not only will attenuate noise but at the same time will allow linear trend functions to pass without distortion. However, we must beware of choosing q to be too large, since if trend function is not linear, the filtered process, although smooth, will not be a good estimate of trend component. By clever choice of the weights $\{a_j\}$ it is possible to design a filter that will not only be effective in attenuating noise in the data, but that will also allow a larger class of trend functions (for example all polynomials of degree less than or equal to 3) to pass through without distortion. The **Spencer 15-point** moving average is a filter that passes **polynomials of degree 3** without distortion.

Exponential Smoothing:

We now introduced one-sided moving average where only past datapoints influence the values. **Exponential smoothing** is a technique used to smooth time series data and generate forecasts by assigning **exponentially decreasing weights** to past observations. We first consider a **smoothing parameter** $\alpha \in (0, 1)$ value between **0 and 1** that controls the weight given to recent observations. We initialize for $t = 1$, $\hat{m}_1 = X_1$ and then define recursive formula for $t \geq 2$:

$$\hat{m}_t = \alpha X_t + (1-\alpha)\hat{m}_{t-1}, \quad t = 2, 3, \cdots$$

We can see the exponential weighted structure as the smoothed value $\hat{m}_t$ is a weighted average of all past observations:

$$\hat{m}_t = \sum_{j=0}^{t-2} \alpha(1-\alpha)^j X_{t-j} + (1-\alpha)^{t-1} X_1, \quad t \geq 2$$

Here, weights are decaying exponentially and the oldest term $(1-\alpha)^{j-1} X_1$ becomes negligible as t grows. The weights form a **geometric series**, decreasing exponentially with each lag.

Polynomial Fitting:

Trend estimation by polynomial fitting involves modeling the underlying trend m_t of a time series as a polynomial function of time,

$$m_t = a_0 + a_1 t + a_2 t^2 + \cdots + a_p t^p,$$

where p is the polynomial order and $(a_0, a_1, \ldots, a_p)$ are coefficients estimated via least squares minimization. The coefficients are chosen to minimize the sum of squared residuals:

$$\sum_{i=1}^{n} \left(x_i - \left(a_0 + a_1 t + a_2 t^2 + \cdots + a_p t^p \right) \right)^2 ,$$

ensuring the polynomial closely follows the observed data $(\{x_1, \ldots, x_n\})$. For example, a quadratic trend $(p = 2)$ is modeled as

$$m_t = a_0 + a_1 t + a_2 t^2,$$

capturing curvature in the data. Higher-order polynomials $(p \leq 10)$ allow flexibility for complex trends, while generalized least squares (available in softwares) adjusts for autocorrelated residuals. Though interpretable and useful for extrapolation, high p risks overfitting noise, and polynomials may diverge unrealistically outside the observed range. This method balances simplicity and mathematical rigor, ideal for data with clear polynomial trends when validated carefully.

Trend Elimination by Differencing

Trend Elimination by Differencing is a method used to remove polynomial trends from a time series by applying the **differencing operator** ∇. We define lag-1 differencing operator as:

$$\nabla X_t = X_t - X_{t-1} = (1 - B) X_t$$

Here, B_t is the backward shift operator $BX_t = X_{t-1}$ and so $B^j X_t = X_{t-j}$. Moreover, $\nabla^j (X_t) = \nabla(\nabla^{j-1}(X_t)), \quad j \geq 1$ with $\nabla^0(X_t) = X_t$.

If our $\{X_t\}$ has some linear trend $X_t = c_0 + c_1 t + Y_t$, then the first order difference:

$$\nabla X_t = X_t - X_{t-1} = c_1 + Y_t - Y_{t-1} = c_1 + \nabla Y_t$$

leaves us with a stationary process with mean c_1.

More generally, for a polynomial trend of degree k, applying k-th times differencing:

$$\nabla^k X_t = k!c_k + \nabla^k Y_t$$

will leave us with a stationary process $\nabla^k Y_t$ with mean $k!c_k$.

We can just visualize this with a time-series X_t having cubic trend:

$$X_t = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + Y_t$$

Applying first difference:

$$\nabla X_t = X_t - X_{t-1} = (c_0 + c_1 t + c_2 t^2 + c_3 t^3 + Y_t) - (c_0 + c_1(t-1) + c_2(t-1)^2 + c_3(t-1)^3 + Y_{t-1})$$

$$\nabla X_t = (c_1 - c_2 + c_3) + (2c_2 - 3c_3)t + 3c_3 t^2 + (Y_t - Y_{t-1})$$

Applying second difference:

$$\nabla^2(X_t) = \nabla(\nabla X_t) = ((c_1 - c_2 + c_3) + (2c_2 - 3c_3)t + 3c_3 t^2 + (Y_t - Y_{t-1}))$$

$$-((c_1 - c_2 + c_3) + (2c_2 - 3c_3)(t-1) + 3c_3(t-1)^2 + (Y_{t-1} - Y_{t-2}))$$

$$\nabla^2 X_t = (2c_2 - 6c_3) + 6c_3 t + \nabla Y_t - \nabla Y_{t-1} = (2c_2 - 6c_3) + 6c_3 t + \nabla^2 Y_t$$

Applying third difference:

$$\nabla^3 X_t = \nabla^2 X_t - \nabla^2 X_{t-1} = 6c_3 + \nabla^2 Y_t - \nabla^2 Y_{t-1}$$

We can see:

$$\nabla^3 X_t = 3!c_3 + \nabla^3 Y_t$$

Therefore, we say this process eliminates the cubic trend, leaving a stationary process $\nabla^3 Y_t$ with mean $3!c_3$.

To determine differencing order k, start with $k = 1$. If residuals still show trends, increase k. Differencing reduces the series length by k. For example, a series of length n becomes $n - k$ after differencing. Excessive differencing ($k > 2$) can introduce artificial correlations or invertible patterns. Differencing eliminates polynomial trends by transforming the series into a stationary process through repeated application of the ∇ operator.

Estimation and Elimination of Both Trend and Seasonality

When you have both a trend m_t and a seasonal component s_t in your series, we define our **classical decomposition model** as:

$$X_t = m_t + s_t + Y_t, \quad E[Y_t] = 0, \quad s_{t+d} = s_t, \quad \sum_{j=1}^{d} s_j = 0$$

the symbol d denotes the seasonal period, i.e. the number of observations in one full cycle of seasonality.

Estimation of Trend and Seasonal components

To estimate the trend and seasonal components in a time series using the classical decomposition method, one can follow these steps:

We estimate the trend $\hat{m}_t$ using a moving average filter, its purpose is to remove seasonality and noise to isolate the trend.

$$\hat{m}_t = \begin{cases} \frac{0.5x_{t-q} + \sum_{j=q-1}^{q+1} x_{t-j} + 0.5x_{t+q}}{d}, \quad q < t \leq n-q \text{ if d is even } d = 2q \\ \frac{1}{d}\sum_{j=-q}^{j=q} x_{t+j}, \quad q+1 < t \leq n-q \text{ if d is odd } d = 2q+1 \end{cases}$$

The first q and last q datapoints will have no trend estimates.

We then compute the deviations from the trend:

$$\delta_t = x_t - \hat{m}_t$$

The edge points will have no trend estimates here also. We can use interpolation for visualtion but we keep them **NA** in initial computation.

Computing average deviations for each seasonal position k, for each $k = 1, 2, \cdots, d$

$$w_k = \frac{1}{N_k}\sum_{j} \delta_{k+jd}$$

N_k is the number of observations at poisition k. We now adjust seasonal indices to sum to zero and estimate seasonal component as:

$$\hat{s}_k = \begin{cases} w_k - \frac{1}{d}\sum_{i=1}^{d} w_i, \quad k = 1, 2, \cdots, d \\ \hat{s}_{k-d}, \quad k > d \end{cases}$$

We now remove this seasonal component to restimate the trend and this will now have trend values assigned for edge points as well.

$$d_t = x_t - \hat{s}_k, \quad t = 1, 2, \cdots, n$$

We reestimate the trend parametrically by fitting polynomial to the deseasonalized data $\{d_t\}$:

$$\hat{m}_t = \sum_{j=0}^{p} a_p t^p$$

In the end, we compute final residuals by $Y_t = x_t - \hat{m}_t - \hat{s}_t$ and our final model is now defined as:

$$X_t = \underbrace{a_0 + a_1 t + \cdots + a_p t^p}_{\textbf{Reestimated Trend}} + \underbrace{\hat{s}_t}_{\textbf{Seasonality}} + \underbrace{\hat{Y}_t}_{\textbf{Noise}}$$

The initial moving average trend is non-parametric and cannot be extrapolated. A parametric polynomial allows forecasting. Polynomial trends may overfit or diverge outside observed range. This method isolates interpretable components for analysis, forecasting, and anomaly detection.

Elimination of Trend and Seasonal Components by Differencing

To eliminate both trend and seasonal components from a time series using differencing, we will first remove the seasonality and then we go simply like we use differencing described earlier.

We first apply seasonal differencing by using the lag-d differencing operator to remove the seasonal component s_t. This removes seasonality by differencing observations separated by the seasonal period d.

$$\nabla_d X_t = X_t - X_{t-d} = (1 - B^d) X_t$$

After seasonal differencing, the series will become:

$$\nabla_d X_t = (m_t - m_{t-d}) + (Y_t - Y_{t-d})$$

It will have trend component as $m_t - m_{t-d}$ and have noise $Y_t - Y_{t-d}$.

If a trend remains in $\nabla_d X_t$, we can now eliminate trend using method described earlier. We apply the first-difference operator as:

$$\nabla(\nabla_d X_t) = \nabla_d X_t - \nabla_d X_{t-1}$$

$$= (m_t - m_{t-d} - m_{t-1} + m_{t-d-1}) + (Y_t - Y_{t-d} - Y_{t-1} + Y_{t-d-1})$$

$$= \nabla m_t - \nabla m_{t-d} + \nabla Y_t - \nabla Y_{t-d}$$

So, we can also write:

$$\nabla(\nabla_d X_t) = \nabla_d(\nabla m_t) + \nabla_d(\nabla Y_t) = \nabla_d(\nabla m_t + \nabla Y_t)$$

Therefore, we have:

$$\nabla(\nabla_d X_t) = \nabla_d(\nabla X_t)$$

This first-order differencing eliminates residual trend (e.g., linear trends become stationary after one differencing). Even for complex trends (e.g., quadratic), we can apply ∇^k until it leads us to stationary process with constant mean. This method transforms the original series into a stationary process by systematically removing trend and seasonality.

In time series analysis, the elimination of trend and seasonal components can be approached through two primary methods: classical decomposition and differencing. Classical decomposition involves estimating and subtracting explicit trend (m_t) and seasonal (s_t) components from the data, leaving a residual noise sequence (Y_t). This method is particularly useful when interpretable estimates of these components are required, such as identifying long-term trends or periodic patterns. Alternatively, differencing employs operators like ∇_d (seasonal differencing) and ∇ (regular differencing) to systematically remove trends and seasonality by transforming the series into a stationary process. For example, $\nabla_d X_t = X_t - X_{t-d}$ eliminates fixed-period seasonality, while subsequent differencing (∇) addresses residual trends. The choice between methods depends on factors such as the need for component estimates, the constancy of seasonal patterns, and the goal of forecasting. After applying either technique, it is critical to validate the resulting noise sequence for independence (e.g., via autocorrelation checks or hypothesis tests like Ljung-Box). If significant serial dependence persists, the noise can be modeled using stationary processes (e.g., ARIMA) to leverage its structure for improved forecasting.