HSLU Hochschule Luzern

# Time Series Analysis

## Forecasting


**Dr. Ludovic Amruthalingam**
ludovic.amruthalingam@hslu.ch

**Informatik**


# Outline

- Forecasting
- Forecasting with baselines, ARIMA
- Underfitting and overfitting
- Evaluation workflow
- Splitting time series data
- Residual analysis
- Confidence intervals
- Information criteria
- Performance metrics
- Back-transforms

# Forecasting

**Extrapolating** past observations to predict future data.

- Works well provided future data follows past patterns.
- Strong signals (low noise) can lead to accurate forecasts.
- Noise increases **uncertainty**, making predictions reliable only for the **short term**.

Sources of uncertainty

- Data: unexpected **disruption** from past patterns.
- Model: chosen model may not represent the **true data-generating process**.
- Parameters: even with the correct model, estimated **parameters may be inaccurate**.
- Forecasts: model typically yield an estimate of the **conditional mean** of future instances, which may be strongly influenced by future **unpredictable innovations**.

Forecasts must be complemented with a measure of the model uncertainty, typically **prediction intervals**.

# Forecasting

Given a time series realization $\{x_1, x_2, \dots x_T, \dots x_n\}$,

The $h$-step forecast of $x_{T+h}$ based on the data $\{x_1, x_2, \dots x_T\}$ is represented as $\hat{x}_{T+h|T}$.

- $T$ is the **forecast time**.
- $h$ is the **forecast horizon** i.e., how far into the future the forecast is made.
- $T+h$ is the **target time** i.e., the time point of the forecast.

Considering a monthly time series and a 1-year forecast horizon,

- **Point** forecast is $\hat{x}_{T+12|T}$.
- **Multi-step** forecast is $\{\hat{x}_{T+1|T}, \dots \hat{x}_{T+12|T}\}$.

# Forecasting baselines

**Mean**: forecasts are equal to the average of the observed data, $\hat{x}_{T+h|T} = \frac{1}{T}\sum_{i=1}^{T} x_i$

**Naïve**: forecasts are equal to the last observed value of the series, $\hat{x}_{T+h|T} = x_T$

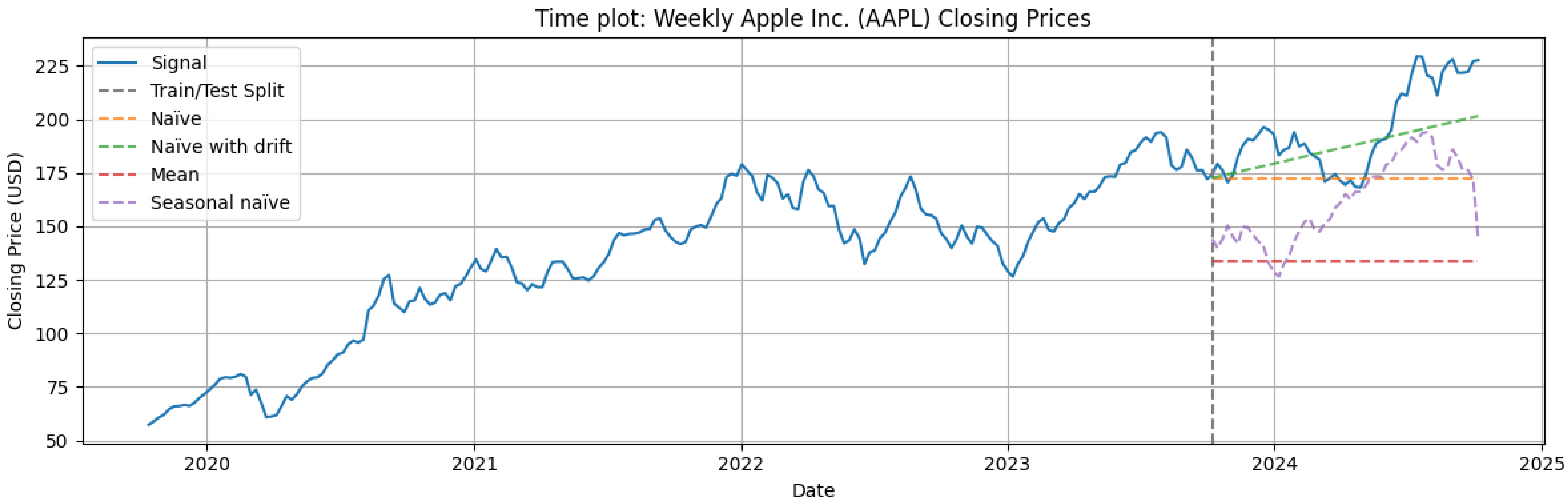
- Naïve is optimal for random walk process.

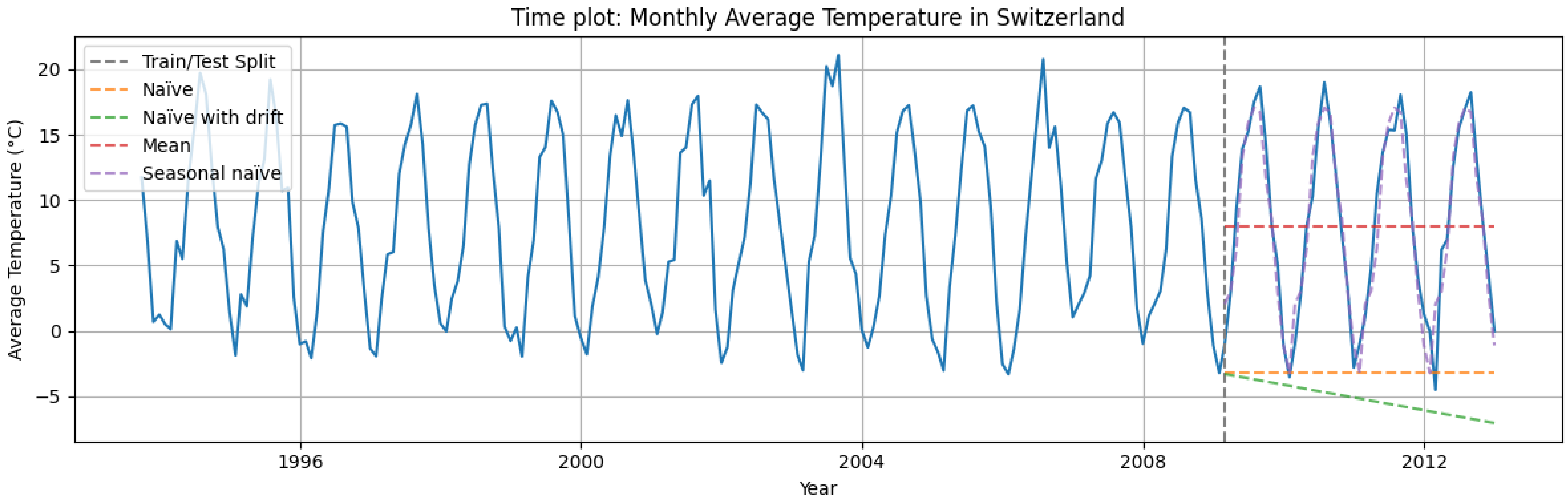
**Seasonal naïve**: forecasts are equal to the last observed value from the same season, $\hat{x}_{T+h|T} = x_{T+h-\lceil h/P \rceil P}$

- Setting $P = 1$ (non-seasonal data) results in the naïve forecast

**Naïve with drift**: naïve forecast with linear drift, $\hat{x}_{T+h|T} = x_T + h\frac{x_T - x_1}{T-1}$

Baselines serve as **benchmark** to evaluate the added value of more complex methods.


Time plot: Weekly Apple Inc. (AAPL) Closing Prices
Signal
Train/Test Split
Naïve
Naïve with drift
Mean
Seasonal naïve
Closing Price (USD)
225
200
175
150
125
100
75
50
2020
2021
2022
2023
2024
2025
Date

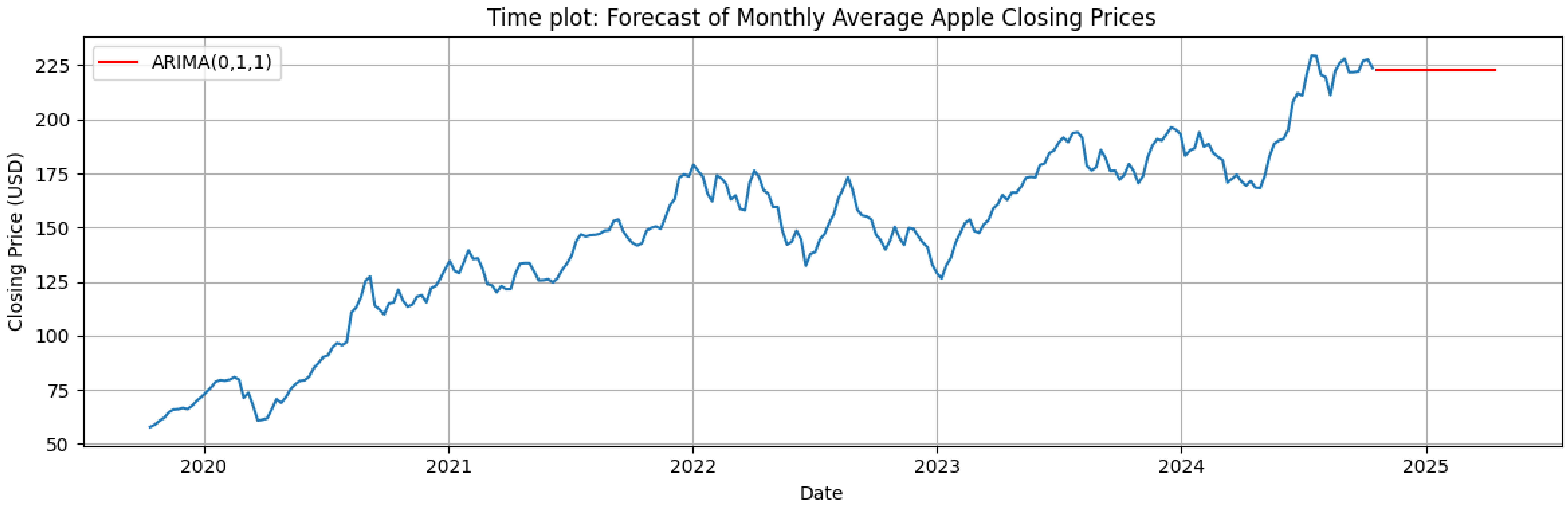

Time plot: Monthly Average Temperature in Switzerland
Train/Test Split
Naïve
Naïve with drift
Mean
Seasonal naïve
Average Temperature (°C)
20
15
10
5
0
−5
1996
2000
2004
2008
2012
Year

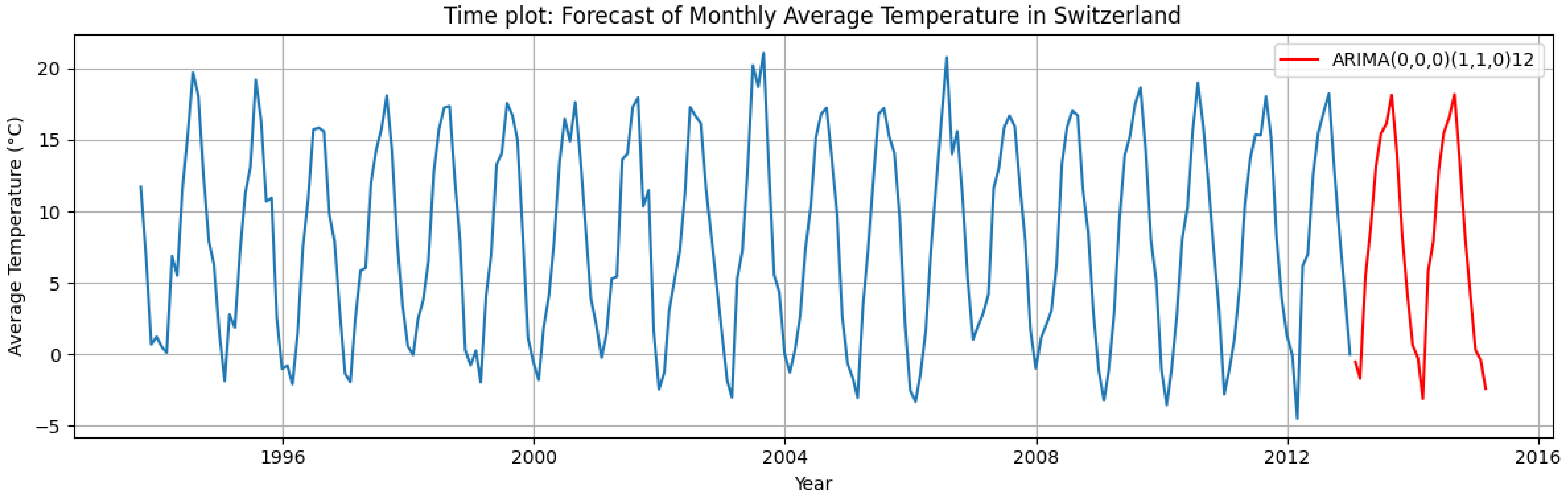
# Forecasting with ARIMA

The forecast $\hat{x}_{T+h|T}$ from an ARIMA model can be computed as follows:

1. Rewrite the ARIMA equation $\Phi(B)\nabla^d X_t = \Phi(B)(1-B)^d X_t = c + \Theta(B)W_t$ with $x_t$ on the left-hand side.
   - For an ARIMA(1,1,1): $\left(1-\hat{\phi}B\right)(x_T - x_{T-1}) = \hat{c} + \left(1+\hat{\theta}B\right)w_T \Leftrightarrow x_T = \hat{c} + w_T + \hat{\theta}w_{T-1} + \left(\hat{\phi}+1\right)x_{T-1} - \hat{\phi}x_{T-2}$

2. Replace future observations with their forecast, future errors with zero, and past errors with the ARIMA residuals.
   - $T+h \longrightarrow x_{T+h} \quad = \hat{c} + w_{T+h} + \hat{\theta}w_{T+h-1} + \left(\hat{\phi}+1\right)x_{T+h-1} - \hat{\phi}x_{T+h-2}$
   - $h=1 \longrightarrow \hat{x}_{T+1|T} = \hat{c} + 0 \quad + \hat{\theta}\hat{w}_T \quad + \left(\hat{\phi}+1\right)x_T \quad - \hat{\phi}x_{T-1}$
   - $h=2 \longrightarrow \hat{x}_{T+2|T} = \hat{c} + 0 \quad + 0 \quad + \left(\hat{\phi}+1\right)\hat{x}_{T+1|T} \quad - \hat{\phi}x_T$
   - $h=3 \longrightarrow \hat{x}_{T+3|T} = \hat{c} + 0 \quad + 0 \quad + \left(\hat{\phi}+1\right)\hat{x}_{T+2|T} \quad - \hat{\phi}\hat{x}_{T+1|T}$
   - …


Time plot: Forecast of Monthly Average Apple Closing Prices
ARIMA(0,1,1)
Closing Price (USD)
225
200
175
150
125
100
75
50
2020
2021
2022
2023
2024
2025
Date


Time plot: Forecast of Monthly Average Temperature in Switzerland
ARIMA(0,0,0)(1,1,0)12
Average Temperature (°C)
20
15
10
5
0
−5
1996
2000
2004
2008
2012
2016
Year

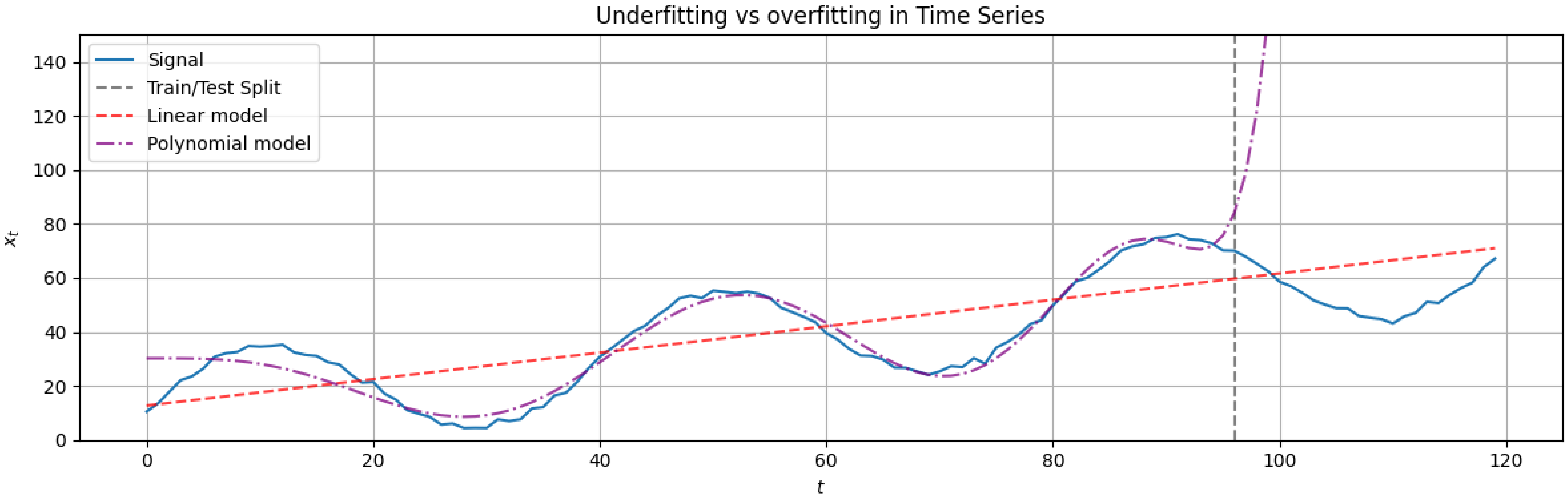
# Underfitting and overfitting

**Underfitting** occurs when a model fits training data poorly and fails to generalize to new data.

- Model is too simple to capture the underlying patterns in the data.

**Overfitting** occurs when a model fits training data very well but fails to generalize to new data.

- Model is too complex and learns to reproduce the training data exactly.
- Training data is too small.
- Training procedure does not involve regularization (discussed in following lectures).



# Evaluation workflow

Given a time series realization $\{x_1, x_2, \dots x_T, \dots x_V, \dots x_n\}$,

1. **Split dataset** into **training** $\{x_1, \dots x_T\}$, **validation** $\{x_{T+1}, \dots x_V\}$, and **test** $\{x_{V+1}, \dots x_n\}$ sets.
   - Validation set is required either when model fitting involves hyperparameters tuning or when model selection is based on performance metrics.
   - Multiple versions of training and validation sets can be considered with cross-validation.
2. **Train** candidate models on training set.
   - Tune hyperparameters using the validation set.
3. **Select model** based on model fit, complexity and performance on the validation set.
   - Information criteria, residual analysis, uncertainty, performance metrics
4. **Train selected model** on training + validation sets then **evaluate performance** on test set.
   - Metrics provide an indication of how well the model will forecast new data.

# Splitting time series data & rolling cross-validation

**Split dataset** into **training** $\{x_1, \dots x_T\}$, **validation** $\{x_{T+1}, \dots x_V\}$, and **test** $\{x_{V+1}, \dots x_n\}$ sets

- Time series must be split **chronologically** → **no random splits**.
- **Seasonality and trends**: ensure the splits account for any patterns in the data.
- Validation and test set should be at least as large as the **forecast horizon** i.e., $\mathrm{h} \leq V - T$ and $\mathrm{h} \leq n - V$.

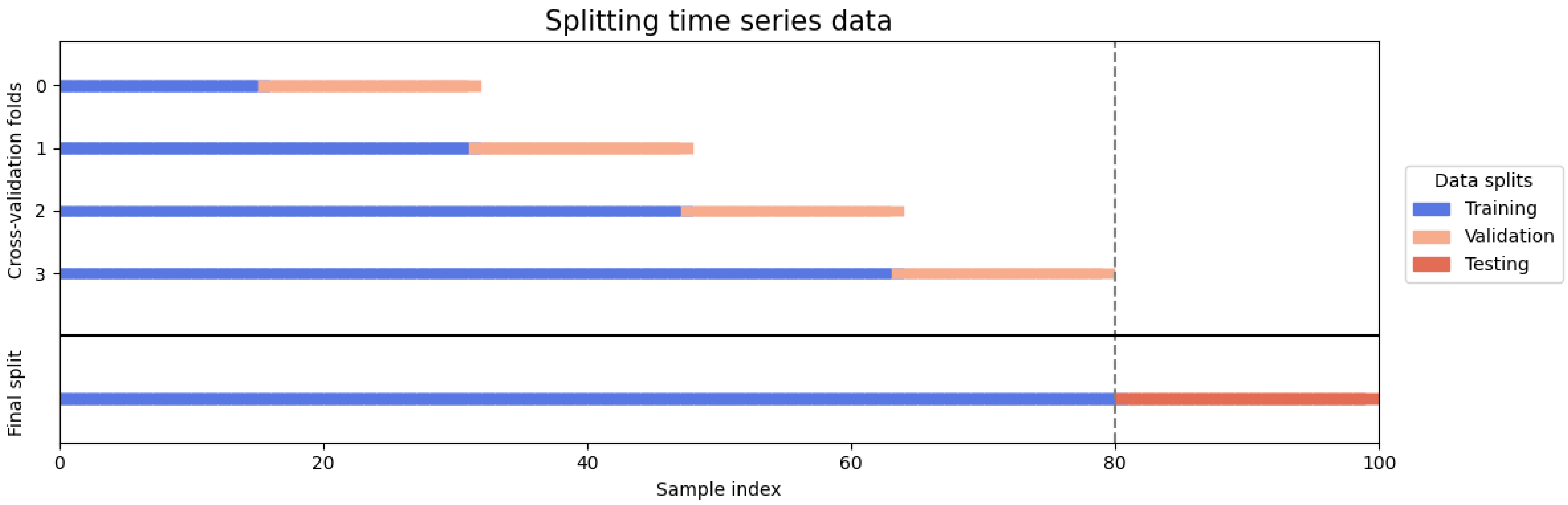
**Use test data once for final evaluation**, otherwise risk of **over-estimating** performance on new data

→ compare candidate models performance on the validation set.

**Rolling Cross-Validation**: sequentially increase the training set, while moving the validation set forward.

Expanding Window Cross Validation

HSLU



# Residual analysis

**Residuals** are the difference between observed values and predicted values: $e_i = x_i - \hat{x}_{i|T}$ for $\mathrm{i} = 1, \ldots T$.

- Also called **training set errors**, it is an estimate of the noise/innovation component of the data.
- Residuals are expected to be **normal, uncorrelated, zero-mean, and homoscedastic**.
- Analyse **standardized residuals** $\tilde{e}_i = e_i / \hat{\sigma}_e$

**Identify patterns or autocorrelations** that the model did not capture.

- Time plot, correlogram
- Ljung-Box test: null hypothesis ($H_0$) states that residuals are **uncorrelated** up to a certain lag.

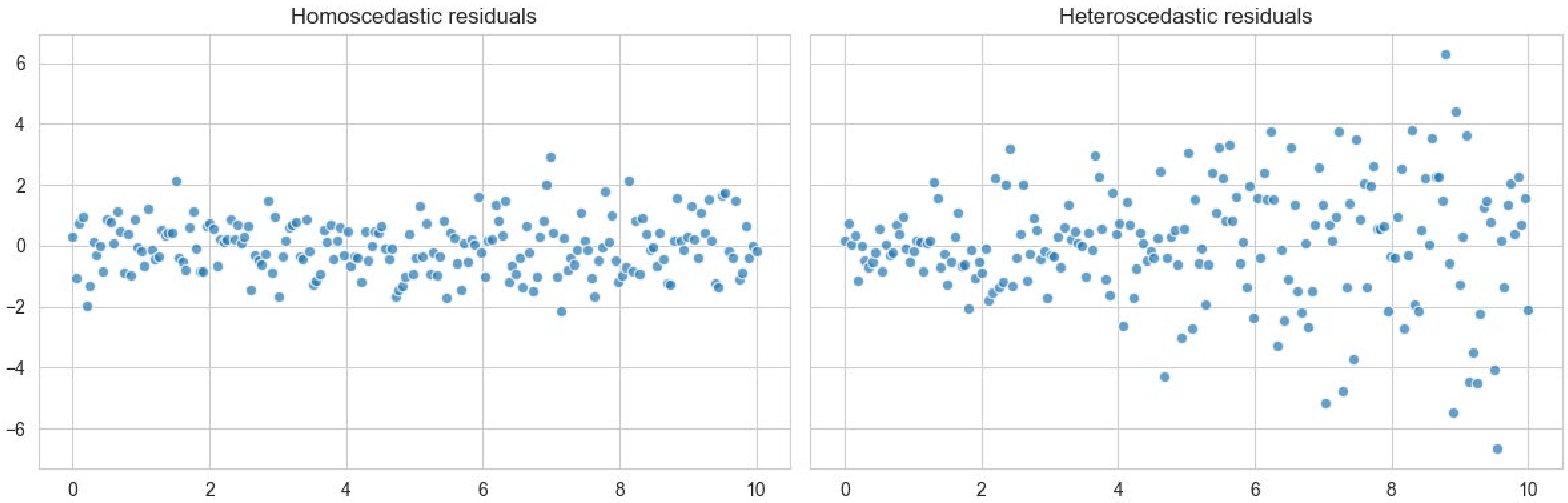
**Validate model assumptions**, typically $E_\mathrm{t} \sim \mathcal{N}(0, \sigma^2)$.

- Q-Q plot: compare the residuals quantiles with normal quantiles.
- Histogram: visual representation of the residuals distribution.

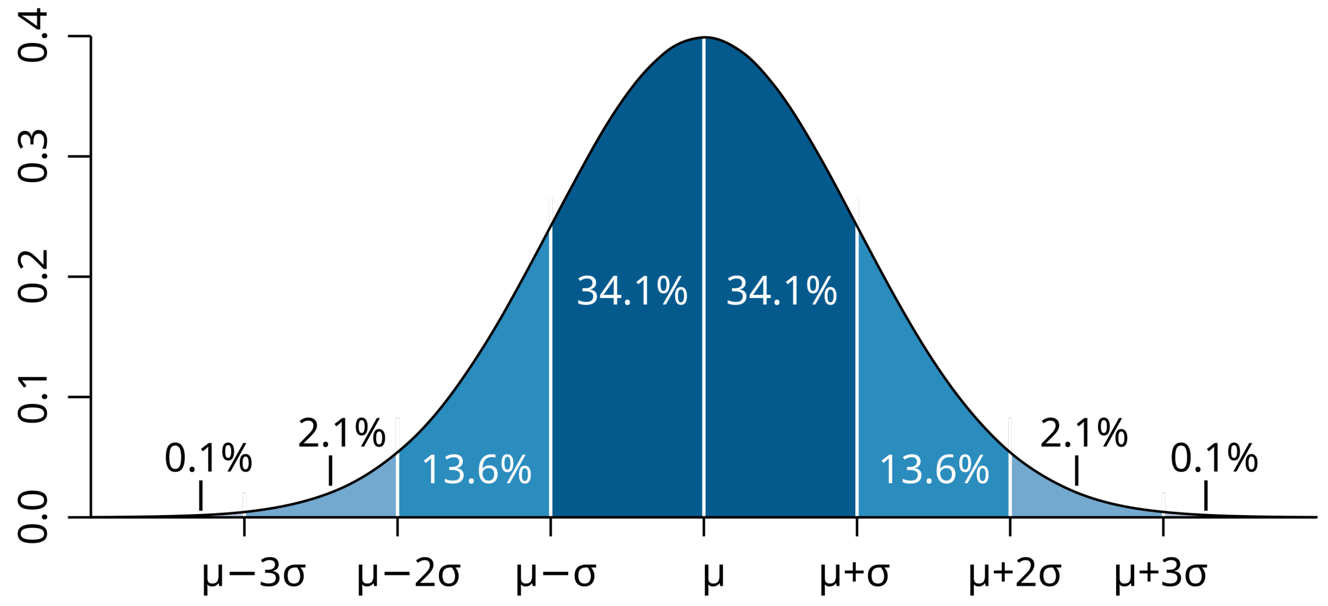
Evaluate how well a model **utilizes available signal** in the data but **does not help with model selection**.

# Recap – Homoscedasticity

The variance of the error is **constant** over the entire feature space.



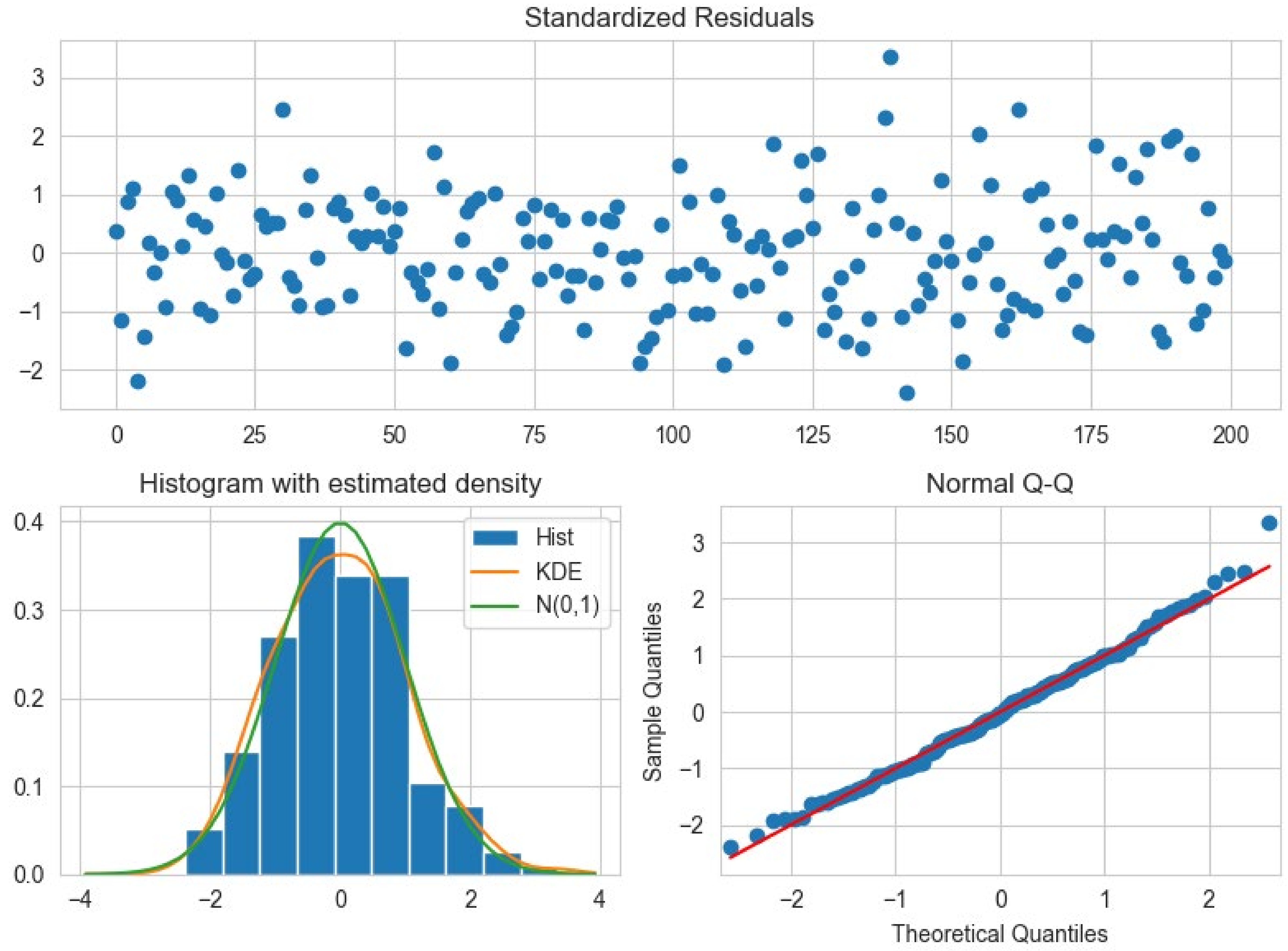
# Recap – Normality



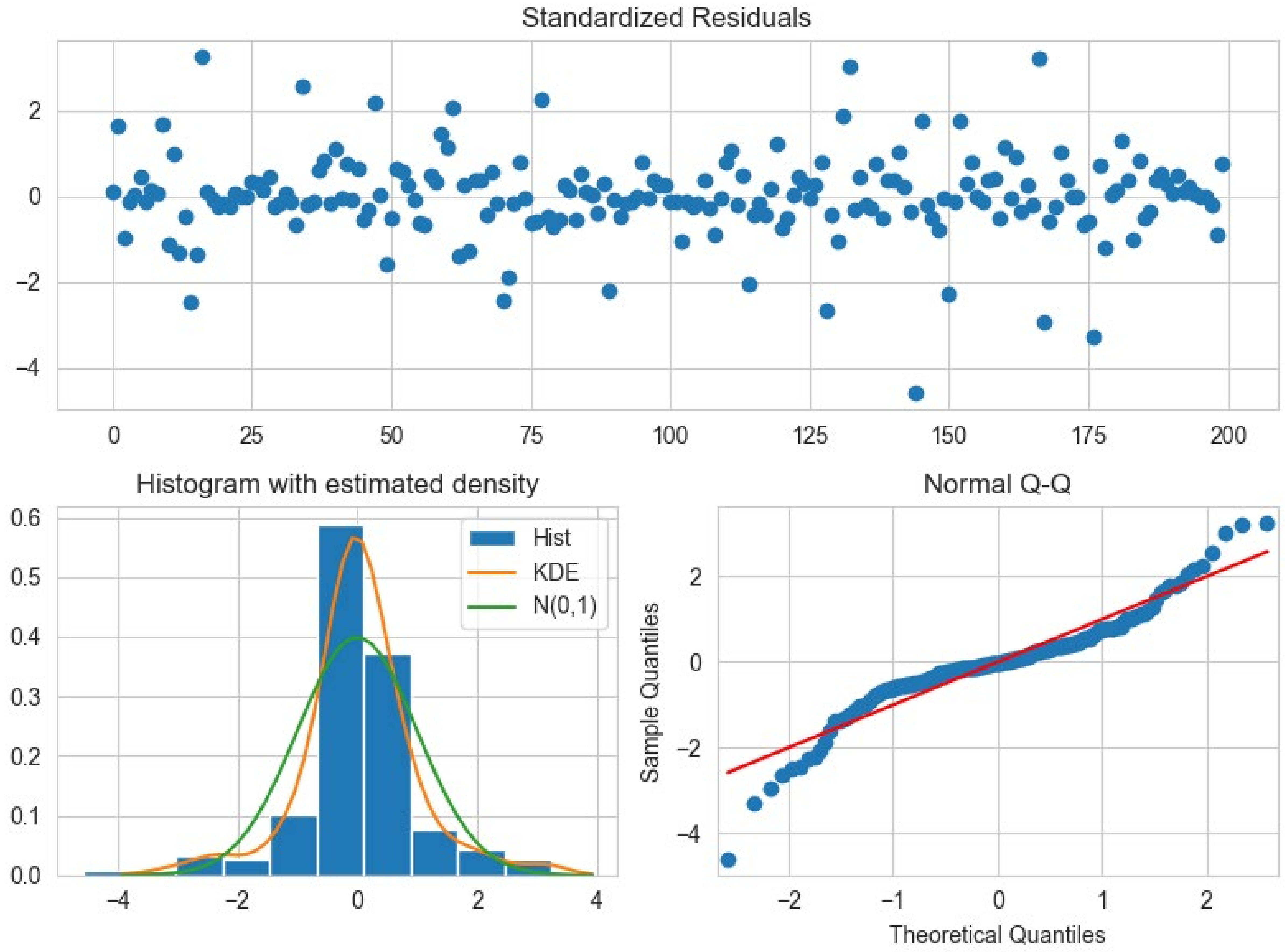

Ainali, Wikimedia

The model error given the features follows a **normal distribution** $\epsilon \mid X \sim \mathcal{N}(0, \sigma^2)$.

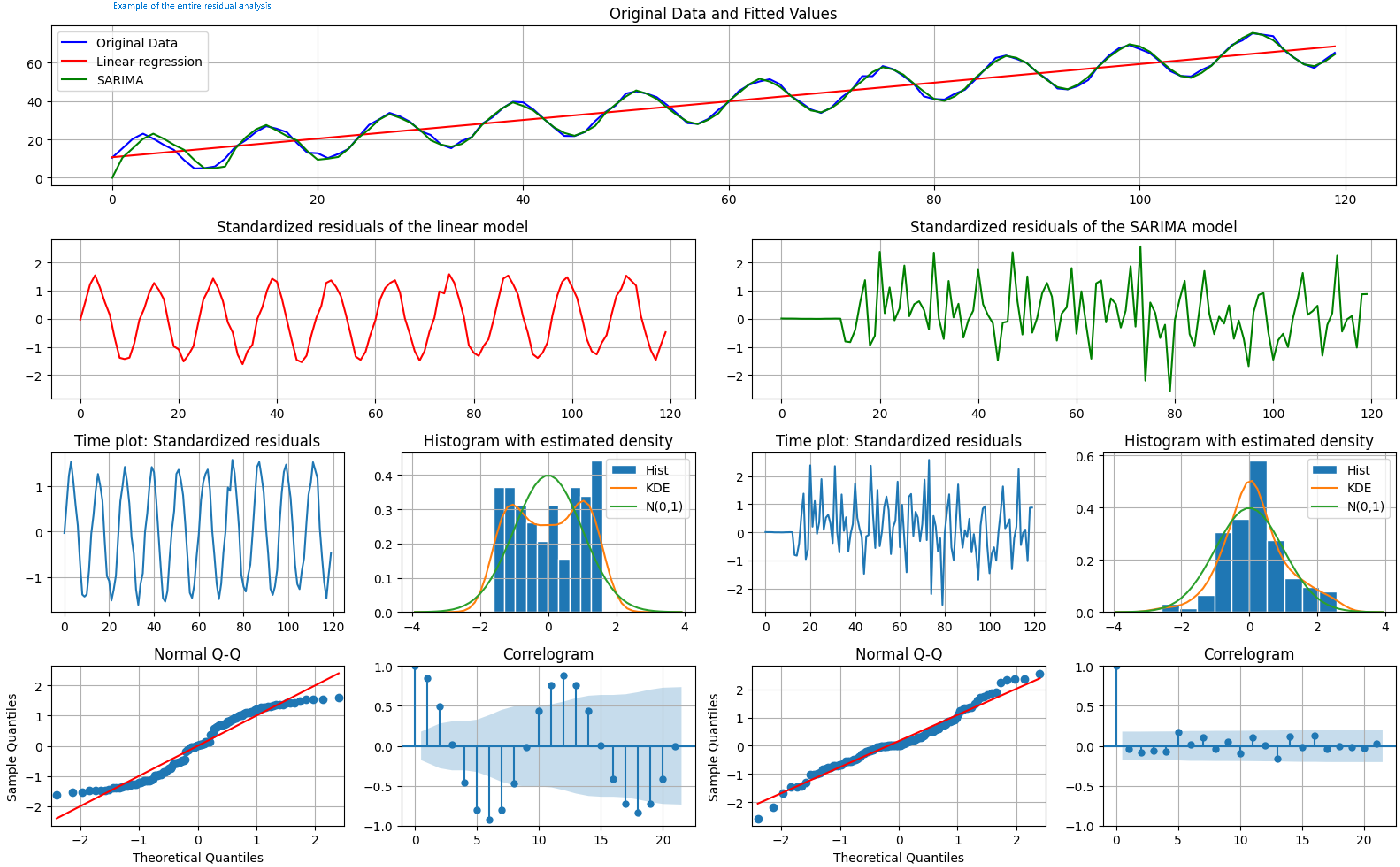
normal



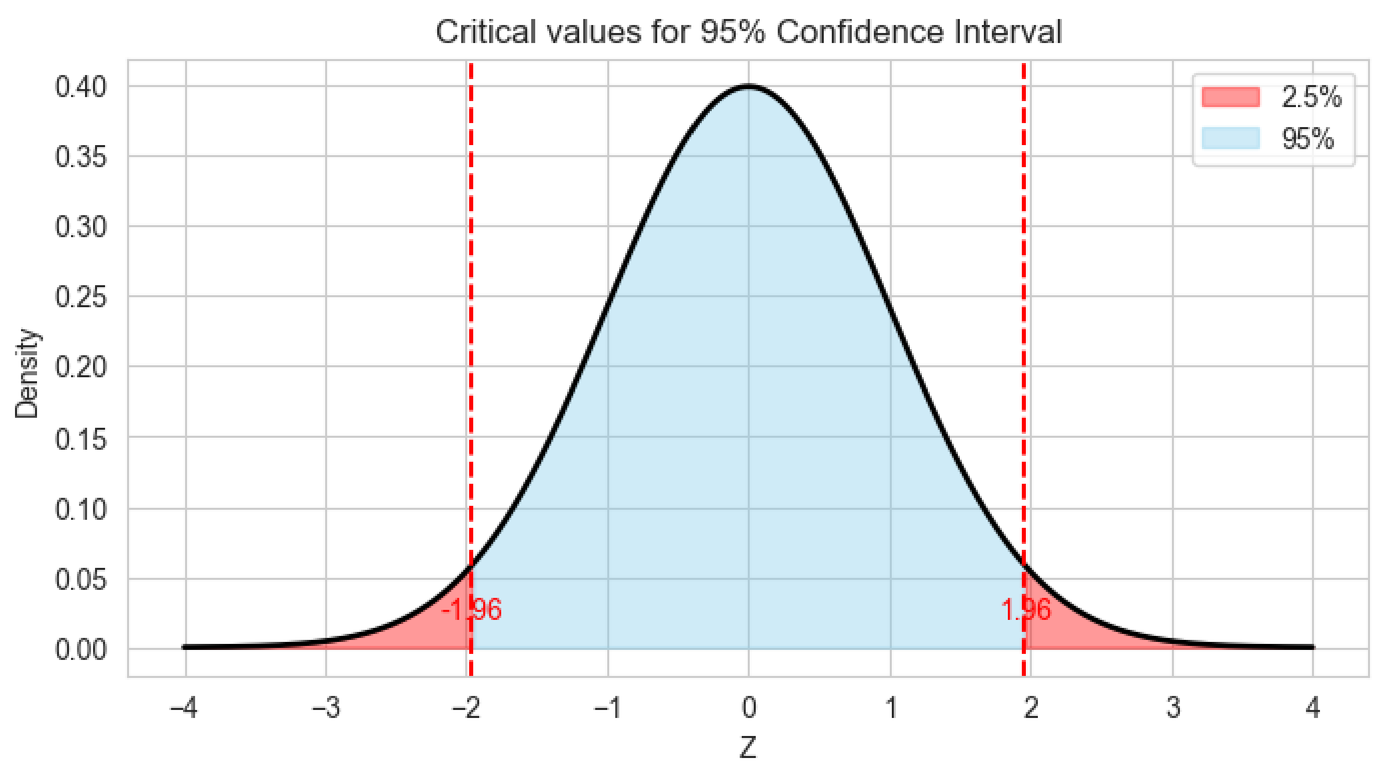

non-normal



Example of the entire residual analysis



# Confidence intervals (CI) for normal residuals




Confidence intervals provide a measure of **forecast uncertainty**.

When the residuals are **normally distributed**, the CI of $\hat{x}_{T+h|T}$ is $\hat{x}_{T+h|T} \pm z_{1-\alpha/2}\hat{\sigma}_{h|T}$

- $z_{1-\alpha/2}$ is the critical value of the normal distribution e.g., for 95% CI $z_{1-\alpha/2} = 1.96$
- $\hat{\sigma}_{h|T}$ is an **estimate of the standard deviation** of the $h$-step forecast with $\hat{\sigma}_1 = \hat{\sigma}_e$ (residuals std)

$$\hat{\sigma}_e = \sqrt{\frac{1}{T-k-m}\sum_{i=m+1}^{T} e_i^2}$$

don't really care about this bias if a lot of data is available

with $k$ the number of model parameters and $m$ the number of missing residuals due to initialization.

For $h > 1$, $\hat{\sigma}_h$ depends on the forecasting method and how the **innovations accumulate**:

| Mean | Naïve | Seasonal naïve | Naïve with drift |
|---|---|---|---|
| $\hat{\sigma}_{h|T} = \hat{\sigma}_e\sqrt{1+\frac{1}{T}}$ | $\hat{\sigma}_{h|T} = \hat{\sigma}_e\sqrt{h}$ | $\hat{\sigma}_{h|T} = \hat{\sigma}_e\sqrt{\lceil h/P \rceil}$ | $\hat{\sigma}_{h|T} = \hat{\sigma}_e\sqrt{h\left(1+\frac{h}{T-1}\right)}$ |

# Confidence intervals (CI) for non-normal residuals

non-parametric way of calculating the confidence interval

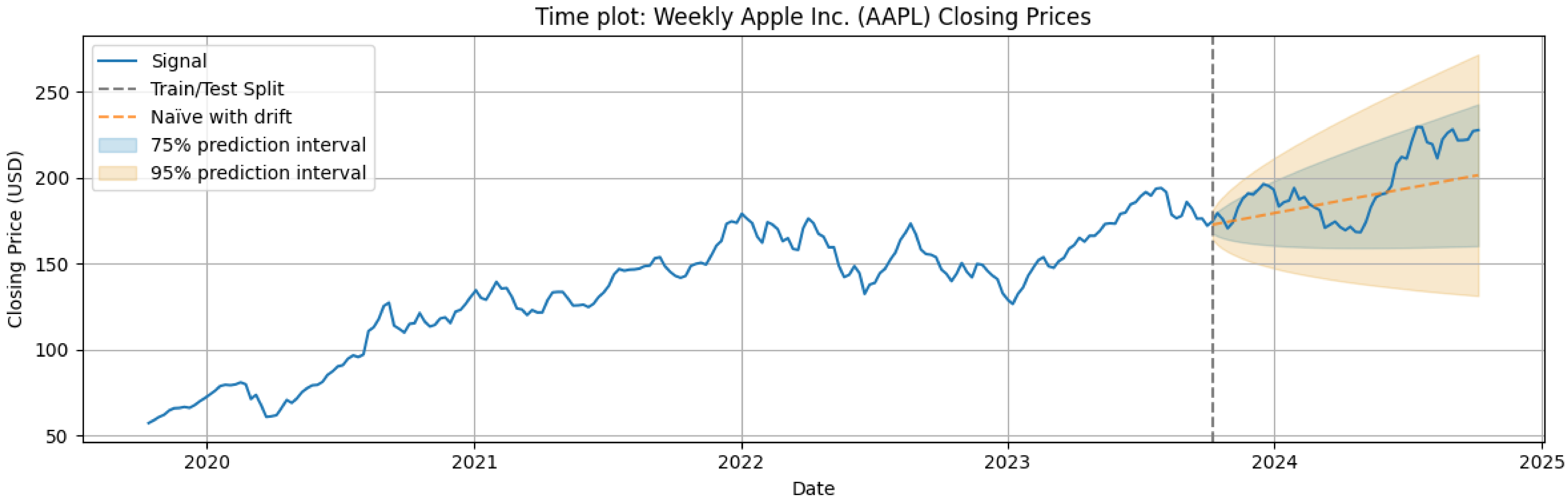
Use **bootstrapping** when the residuals are **uncorrelated** and have **constant variance**.

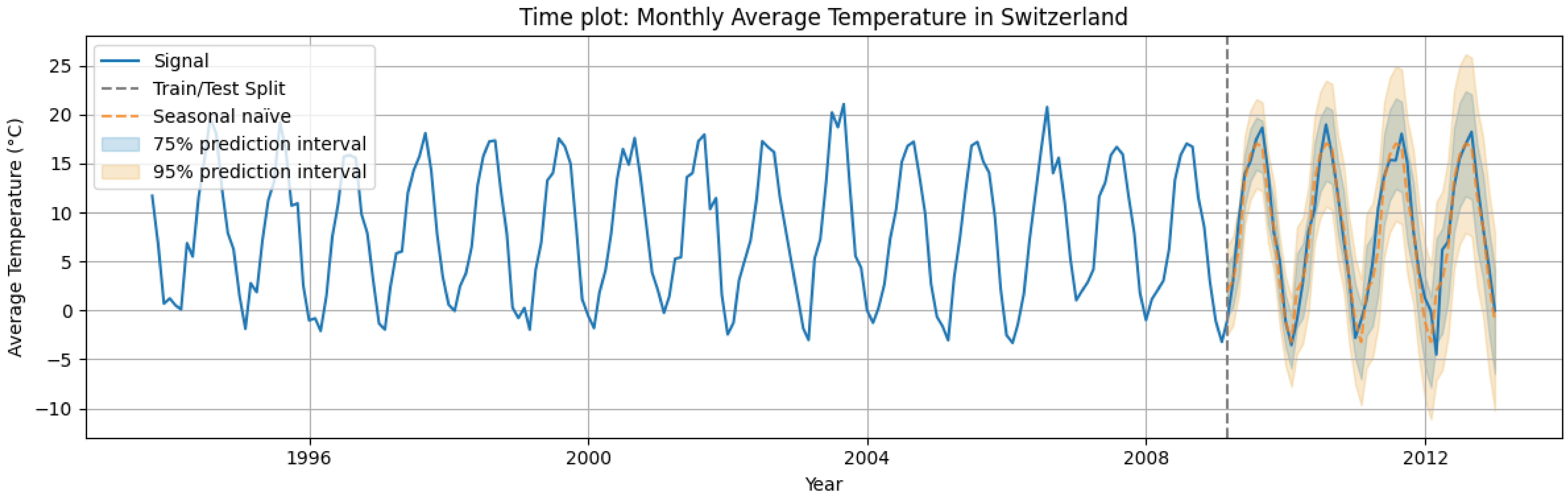
with (strongly) correlated residuals --> problem
correlated residuals mean remaining signal that the model hasn't picked up.

Assuming future and past errors will be similar, generate possible futures:

1. Fit forecasting model on $\{x_1, x_2, \dots x_T\}$ and compute residuals $\{e_1, e_2, \dots e_T\}$.
2. **Resample residuals with replacement** to simulate future forecast errors $\{e_{k_1}, e_{k_2}, \dots e_{k_h}\}$.
3. Generate futures by adding resampled residuals to model forecasts $\{\hat{x}_{T+1|T} + e_{k_1}, \hat{x}_{T+2|T} + e_{k_2}, \dots \hat{x}_{T+h|T} + e_{k_h}\}$.
4. Repeat step 2, 3 multiple times e.g., 1000 iterations.
5. Derive confidence intervals by computing the **percentiles**.
    - e.g., for 95% CI use the 2.5$^{th}$ and 97.5$^{th}$ percentiles.

Note that the bootstrapped confidence intervals are **not symmetric**.


Time plot: Weekly Apple Inc. (AAPL) Closing Prices
Signal
Train/Test Split
Naïve with drift
75% prediction interval
95% prediction interval
Closing Price (USD)
250
200
150
100
50
2020
2021
2022
2023
2024
2025
Date


Time plot: Monthly Average Temperature in Switzerland
Signal
Train/Test Split
Seasonal naïve
75% prediction interval
95% prediction interval
Average Temperature (°C)
25
20
15
10
5
0
−5
−10
1996
2000
2004
2008
2012
Year

# Information criteria

Measure of the **goodness of fit** of a model while penalizing for **model complexity**.

Goodness of fit is measured by the **likelihood** of the data under the model:

$$L = \prod_{i=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{{e_i}^2}{2\sigma^2}\right) \text{ with } e_i = x_i - \hat{x}_{i|T} \text{ and assuming } e_i \sim \mathcal{N}(0, \sigma^2)$$

**Akaike's Information Criterion**: $AIC = 2k - 2log(L)$ with $k$ the number of model parameters.

**Bayesian Information Criterion**: $\mathrm{BIC} = \log(T)\, k - 2log(L)$

For ARIMA $e_i = \widehat{w}_i$ and $k = \begin{cases} p+q & if\ c = 0 \\ p+q+1 & if\ c \neq 0 \end{cases}$

Select model **minimizing either AIC or BIC** (for models in the same class)

- BIC tends to favor simpler models than AIC due to a larger penalty term.
- AIC prioritizes **model fit** (potentially better performance), while BIC emphasizes **model simplicity** (faster inference, simpler model interpretation).

# Performance metrics

Given a validation set $\{x_{T+1}, \ldots x_V\}$, the forecast performance can be evaluated based on

- **Scale-dependent errors**: $e_i = x_i - \hat{x}_{i|T}$ (same unit as the data → not comparable for TS with different units)
- **Scaled errors**: $e_i = \left(x_i - \hat{x}_{i|T}\right) / \left(\frac{1}{T} \sum_{j=1}^{T} \left| x_j - \hat{x}'_{j|T} \right|\right)$ with $\hat{x}'_{j|T}$ a baseline **training** forecasts.

Considering the **multi-step** forecasts $\{\hat{x}_{T+1|T}, \ldots \hat{x}_{V|T}\}$, the errors can be aggregated as follows:

- Mean absolute (scaled) error MAE/MASE: $\frac{1}{T-V} \sum_{i=T+1}^{V} |e_i|$ → robust to outliers
- Root mean squared (scaled) error RMSE/RMSSE: $\sqrt{\frac{1}{T-V} \sum_{i=T+1}^{V} e_i^2}$ → sensitive to outliers

When units has a **meaningful zero**, consider the mean absolute **percentage errors** MAPE $\frac{1}{T-V} \sum_{i=T+1}^{V} \left| \frac{100\left(x_i - \hat{x}_{i|T}\right)}{x_i} \right|$

# Rolling forecast performance

When the validation set encompasses multiple forecast horizons $\{x_{T+1}, \dots, x_h, \dots x_V\}$,

1. With $i = 0$ for the first iteration, forecast $\{\hat{x}_{T+i+1|T}, \dots \hat{x}_{T+i+h|T}\}$.
2. Compute performance metrics of the forecast.
3. Increment $i = i + k$ with $k$ the chosen step-size, typically $k = h$.
4. Refit model with the newly available values $\{x_{T+i+1}, \dots, \dots x_{T+i+h}\}$.
   - On the test set choosing between **refit** vs **update** strategy depends on the training objective.
   - Update: recalculate model internal state given new data points **without refitting its parameters**.
5. Repeat until the end of the validation set and then aggregate performance.

# Back-transforms

("just be aware of this, but library will do it for you")

To obtain forecasts on the **original scale**, we need to **reverse transformations** applied to the data.

The back-transform for differencing is

- First-order: $\hat{x}_{T+h|T} = x_T + \sum_{i=1}^{h} \hat{y}_{T+i|T}$ with $\mathrm{y_t} = \nabla x_t$
- Seasonal differencing: $\hat{x}_{T+(kP+n)|T} = x_{T+n} + \sum_{i=1}^{k} \hat{y}_{T+iP+n|T}$ with $\mathrm{y_t} = \nabla_{\mathrm{P}} x_t$

Reversing non-linear transforms does **not** preserve the **mean** from the transformed scale but the **median** (assuming the distribution on the transformed scale is symmetric)

- Considering a log-normal distribution $\mathrm{y_t} = \log(x_t)$, the mean of $y_t$ corresponds to the median of $x_t$.
- **Bias correction** is needed to account for the variance in the transformed space.

| Reverse Box-Cox transform | Bias adjusted reverse Box-Cox transform |
|---|---|
| $\hat{x}_{T+h\|T} = \begin{cases} \exp\left(\hat{y}_{T+h\|T}\right) & if\ \lambda = 0 \\ sign\left(\lambda\hat{y}_{T+h\|T}+1\right)\left\|\lambda\hat{y}_{T+h\|T}+1\right\|^{\frac{1}{\lambda}} & if\ \lambda \neq 0 \end{cases}$ | $\hat{x}^{*}_{T+h\|T} = \begin{cases} \hat{x}_{T+h\|T}\left(1+\sigma_h^2/2\right) & if\ \lambda = 0 \\ \hat{x}_{T+h\|T}\left(1+\dfrac{\sigma_h^2(1-\lambda)}{2\left(\lambda\hat{y}_{T+h\|T}+1\right)^2}\right) & if\ \lambda \neq 0 \end{cases}$ |

# Exercise

**Forecasting**

- Split real-world time series into train/test sets.
- Fit ARIMA on training, forecast test observations.
- Plot forecasts vs. actuals, what patterns does your model capture or miss?
- Perform residual analysis
- Generate 80% / 95% forecast intervals. Are values within the intervals? What does this imply?

**Evaluation workflow**

- Use cross-validation to generate different validation folds
- Compute rolling forecast performance. Which metric is best suited for your data?
- Compare ARIMA with baseline models (mean, naïve, seasonal naïve).
- Review sktime forecasting notebook and test different forecasting approaches.
- Select best model and evaluate performance on the test set.