

# Time Series Analysis

## Understanding Patterns in Sequential Data

Prof. Dr. Jörg Osterrieder

- 1 Foundations
- 2 Classical Time Series Models
- 3 Volatility Models
- 4 Forecasting and Evaluation
- 5 Advanced Topics
- 6 Summary and Key Takeaways

After completing this lesson, you will be able to:

1. Identify and decompose time series components (trend, seasonality, noise) and test for stationarity
2. Apply ARIMA and SARIMA models using the Box-Jenkins methodology for non-stationary and seasonal data
3. Model volatility clustering with GARCH for financial time series
4. Evaluate forecasts using accuracy metrics and time series cross-validation

---

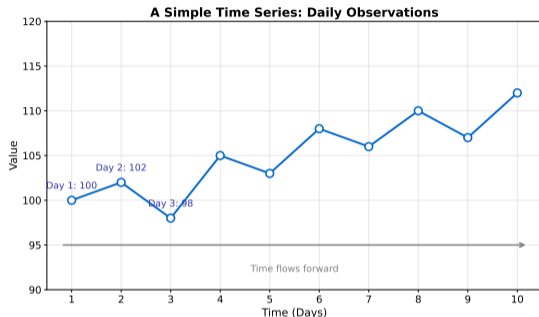
**These four skills cover the complete time series workflow from decomposition to forecasting**

# Foundations

# Why Time Series Analysis?

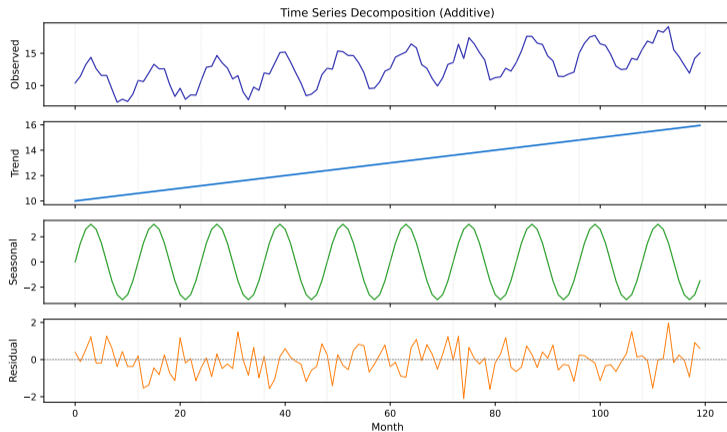
## Sequential data is everywhere

- Sequential data is everywhere: stock prices, weather, sales, sensor readings
- Past values help predict the future — yesterday affects today (autocorrelation)
- Patterns repeat at regular intervals (seasonality) and can be exploited
- Specialized methods handle the temporal dependence that standard models ignore



Sequential data requires specialized methods — the order of observations carries information

# Components of a Time Series



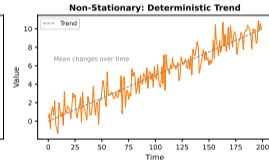
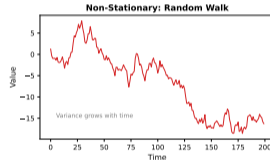
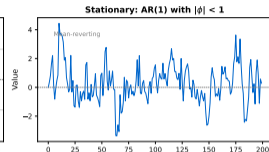
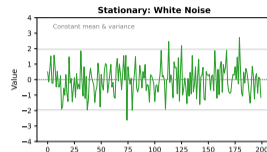
## Key Observations:

- Trend captures the long-term direction (e.g., rising population)
- Seasonality captures regular repeating patterns (e.g., summer ice cream sales)
- Noise (residual) is the unpredictable random component left after removing trend and season

Observed = Trend + Seasonal + Noise — decomposition reveals the hidden structure

## Constant statistical properties

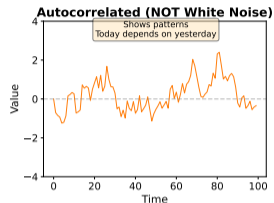
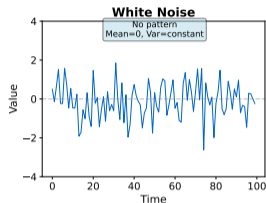
- A stationary series has constant mean and variance over time
- Example: Room temperature stays at  $20^{\circ}\text{C} \pm 2$  (stationary) vs. global warming (non-stationary)
- Most time series models require stationarity as a prerequisite
- Non-stationary data must be transformed (differencing) before modeling



Stable mean and variance are prerequisites for most time series models

## The baseline of randomness

- White noise has mean = 0, constant variance, and zero autocorrelation at all lags
- $ACF(1) = 0$ ,  $ACF(2) = 0$ , ... — no memory, no pattern to exploit
- Good model residuals should look like white noise
- If residuals are not white noise, the model missed a pattern

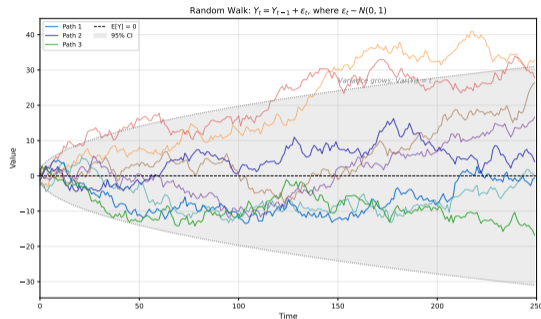


White noise means no pattern left — if residuals are white noise, the model captured everything

## Today plus a random shock

$$Y_t = Y_{t-1} + \epsilon_t$$

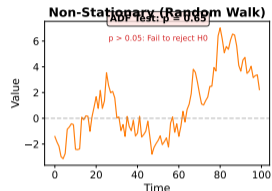
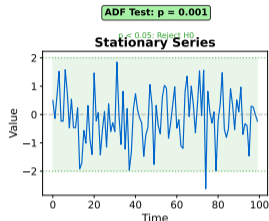
- A random walk adds a random shock to yesterday's value
- Best forecast for tomorrow is simply today's value
- Stock prices often behave like random walks — same start, wildly different paths



Stock prices often behave like random walks — the best forecast is simply today's value

## Visual and statistical tests

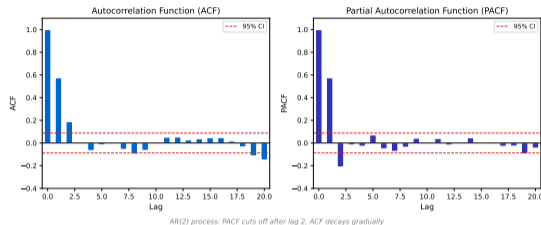
- Visual inspection: Does the plot look stable in level and spread?
- ADF test: Low p-value  $\rightarrow$  evidence of stationarity (reject unit root)
- KPSS test: Low p-value  $\rightarrow$  evidence of non-stationarity (reject stationarity)
- Best practice: Use both ADF and KPSS together — they test opposite null hypotheses



Use both ADF and KPSS tests together — they test opposite null hypotheses

## Measuring temporal dependence

- ACF measures how today correlates with each past lag (e.g.,  $ACF(1) = 0.8$  means 80% correlated with yesterday)
- PACF measures the direct effect of a specific lag, controlling for intermediate lags
- ACF/PACF patterns are the primary tool for identifying AR and MA model orders
- Significant spikes outside the confidence bands indicate meaningful temporal dependence



ACF/PACF patterns are the primary tool for identifying AR and MA model orders

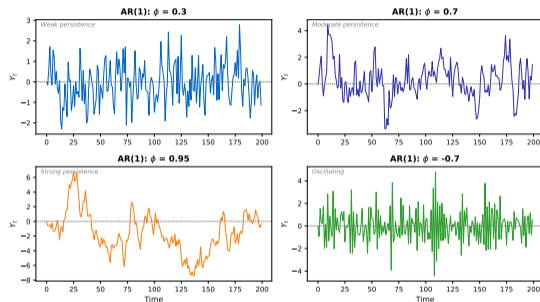
## Classical Time Series Models

## Autoregressive model

$$Y_t = c + \phi_1 Y_{t-1} + \epsilon_t$$

- AR(p) models predict today from the past p values — higher  $\phi$  means stronger persistence
- Example:  $\phi = 0.9$  and  $Y_t = 100 \rightarrow$  next value  $\approx 90 +$  noise
- ACF decays slowly; PACF cuts off at lag p — this identifies the AR order

AR(1) Process:  $Y_t = \phi Y_{t-1} + \epsilon_t$



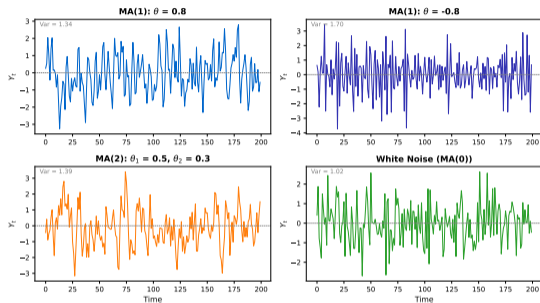
Higher phi means stronger persistence — AR models capture momentum in sequential data

## Moving average model

$$Y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1}$$

- MA(q) models capture the effect of past shocks — the effect fades after q lags
- Example:  $\theta = 0.5$ , shock of 10  $\rightarrow$  effect at  $t+1$  is 5, at  $t+2$  is 0
- ACF cuts off at lag q; PACF decays — the mirror image of AR identification

MA Process:  $Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$

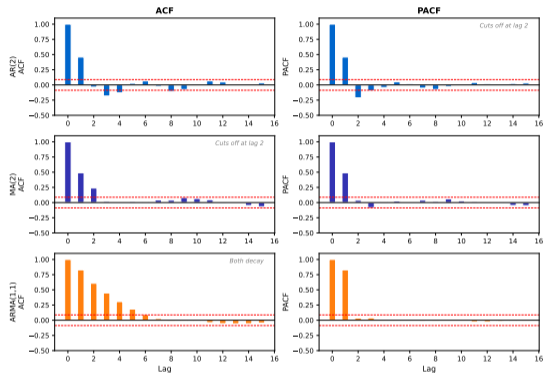


Shocks fade after q lags — MA models capture short-lived effects that dissipate quickly

Model	ACF	PACF
AR(p)	Decays	Cuts at p
MA(q)	Cuts at q	Decays
ARMA	Decays	Decays

- ARMA combines AR persistence with MA shock effects:  

$$Y_t = \phi Y_{t-1} + \epsilon_t + \theta \epsilon_{t-1}$$
- When both ACF and PACF decay gradually, an ARMA model is indicated

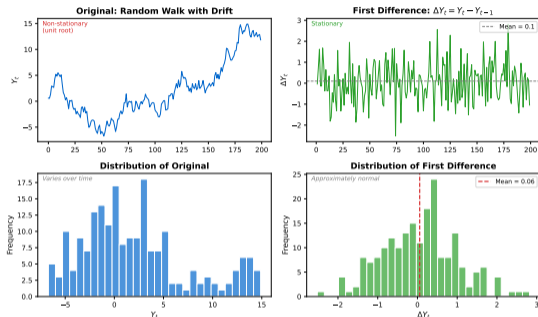


Match the observed ACF/PACF pattern to the identification table to select the right model

## Remove trends by looking at changes

$$\Delta Y_t = Y_t - Y_{t-1}$$

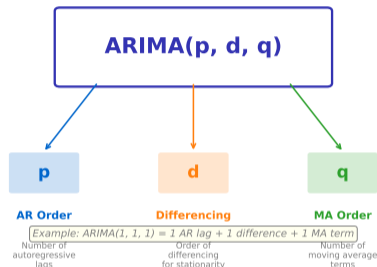
- Differencing converts non-stationary data to stationary by looking at changes, not levels
- Example: Original 100, 102, 105, 109 → Differenced: 2, 3, 4 (trend removed)
- Most series need  $d = 1$  differencing; seasonal differencing removes seasonal patterns



Difference until stationary, then model — most real-world series need  $d = 1$

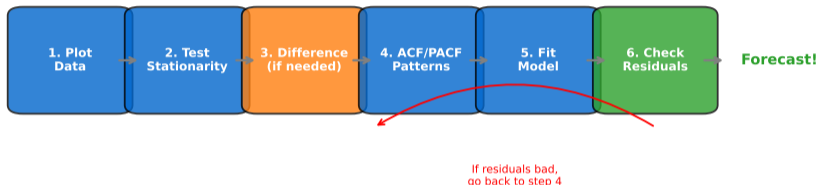
## Unified framework

- $ARIMA(p,d,q)$ :  $p$  = AR order (past values),  $d$  = differencing order,  $q$  = MA order (past shocks)
- Example:  $ARIMA(1,1,1)$  means 1 AR lag + difference once + 1 MA term
- First difference to achieve stationarity, then fit ARMA to the differenced series
- The integrated (I) component handles the non-stationarity that AR and MA cannot



ARIMA unifies differencing (I) with autoregression (AR) and moving average (MA)

## Box-Jenkins Methodology



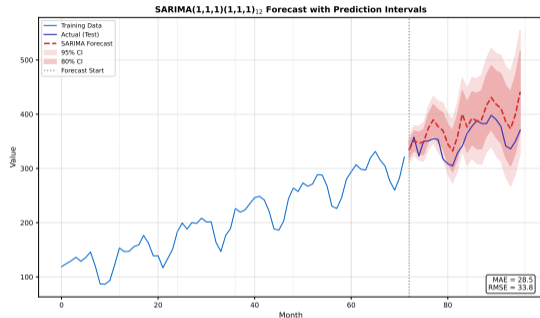
### Key Observations:

- Step 1: Plot the data and visually inspect for trend, seasonality, and outliers
- Step 2: Test stationarity (ADF/KPSS); difference if needed
- Step 3: Examine ACF/PACF to identify model orders ( $p$ ,  $q$ )
- Step 4: Fit model, check residuals are white noise, then forecast

Box-Jenkins provides a systematic four-step approach from raw data to validated forecasts

## Extending ARIMA for seasonality

- $SARIMA(p,d,q)(P,D,Q)_s$  extends ARIMA with seasonal AR, differencing, and MA components
- $s$  = season length (12 for monthly, 4 for quarterly, 7 for daily with weekly pattern)
- Example:  $SARIMA(1,1,1)(1,1,1)_{12}$  captures both short-term dynamics and monthly seasonality
- Seasonal differencing ( $Y_t - Y_{t-s}$ ) removes the repeating seasonal component

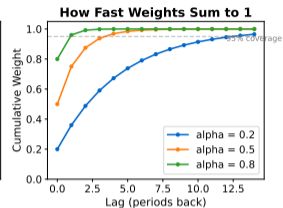
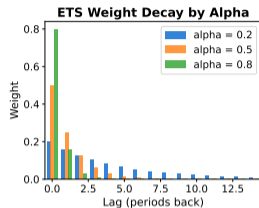


SARIMA extends ARIMA to capture both short-term dynamics and seasonal repeating patterns

# Exponential Smoothing (ETS)

## Weighted recent observations

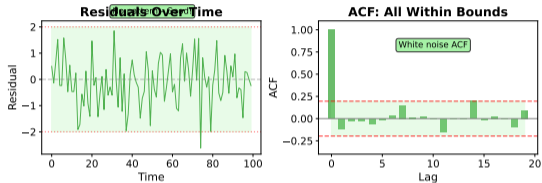
- Exponential smoothing weights recent observations more heavily — weight decays exponentially
- ETS (Error, Trend, Seasonal) is a flexible framework that includes many common methods
- High  $\alpha$  = fast adaptation to recent data; Low  $\alpha$  = smooth, slow-changing forecasts
- Select models using AIC/BIC (lower = better); try both ARIMA and ETS, pick by forecast error



Try both ETS and ARIMA — pick whichever produces lower forecast error on held-out data

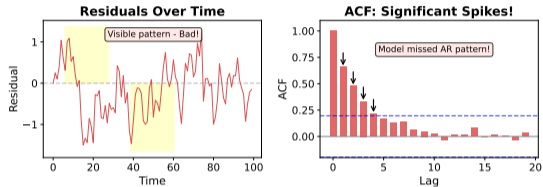
## Good Residuals

GOOD Residual Diagnostics



## Bad Residuals

BAD Residual Diagnostics - Need Better Model

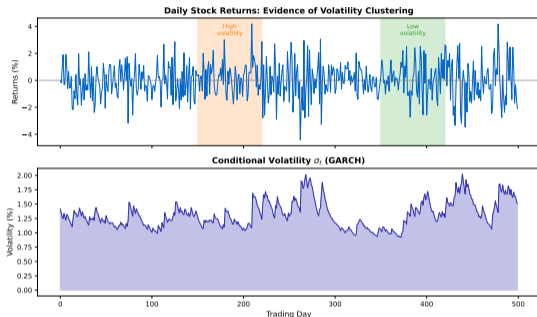


Good residuals look like white noise; patterned residuals mean the model missed something

## Volatility Models

## Time-varying variance in finance

- Financial returns show volatility clustering: big changes follow big changes, small follow small
- ARIMA assumes constant variance — it cannot capture this time-varying volatility
- Volatility clustering is visible in stock returns: calm periods alternate with turbulent ones
- GARCH models extend ARIMA by making the variance itself a time series



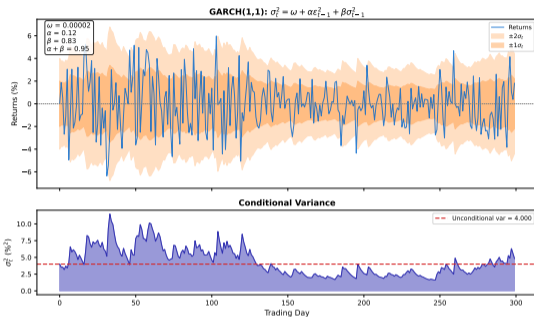
Stylized Facts: Large returns cluster together | Volatility is persistent | Returns are approximately uncorrelated

After big moves, expect more big moves — volatility clustering is the hallmark of financial data

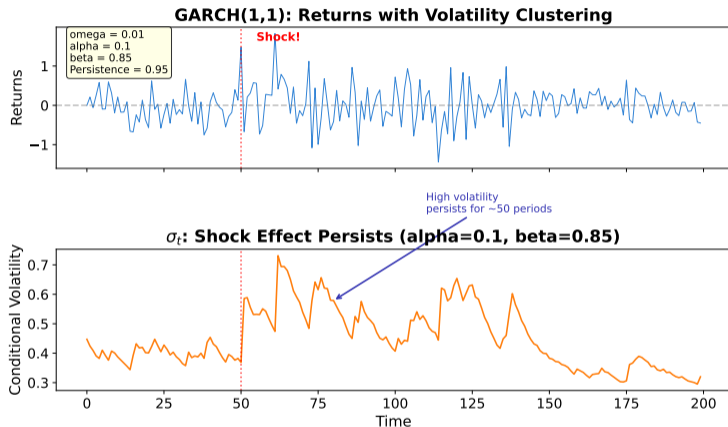
## Conditional variance model

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

- $\alpha$  controls reaction to shocks: how fast volatility responds to new information
- $\beta$  controls persistence: how slowly elevated volatility decays back to normal
- Typical financial values:  $\alpha \approx 0.1$ ,  $\beta \approx 0.85$  ( $\alpha + \beta < 1$  ensures stationarity)



GARCH makes the variance itself a time series — bands expand in turbulent periods



## Key Observations:

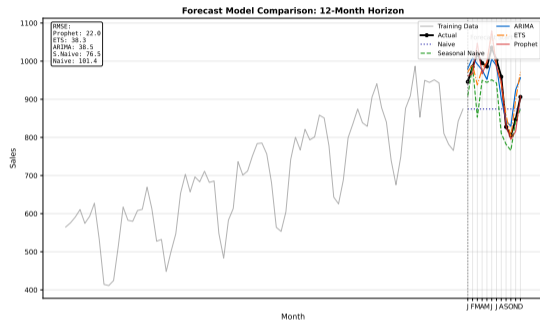
- A single large shock can keep volatility elevated for 50+ periods when  $\beta$  is high
- GARCH confidence bands widen during turbulent periods and narrow during calm ones
- Used for risk management (Value at Risk), option pricing, and portfolio optimization

A single large shock keeps volatility elevated for dozens of periods when beta is high

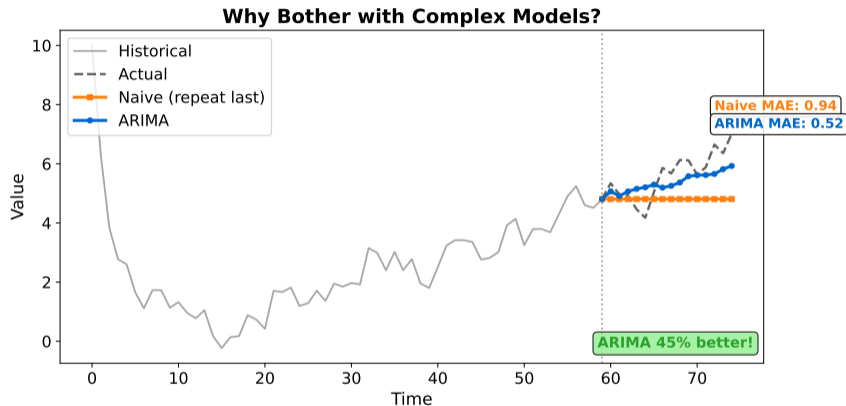
## Forecasting and Evaluation

## Benchmarks and comparisons

- Point forecast: the single best guess for the future value
- Naive benchmark: tomorrow = today; seasonal naive: tomorrow = same day last year
- If a sophisticated model cannot beat the naive benchmark, it adds no value
- Always compare your model's forecast error against at least one benchmark



**Always compare your model against a naive benchmark — if it cannot beat naive, it adds no value**



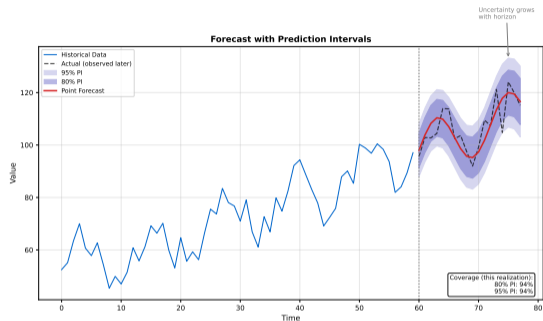
## Key Observations:

- ARIMA captures underlying patterns that naive forecasting misses
- The improvement is especially visible when data contains trend or seasonal structure
- If ARIMA cannot beat naive, the data may be a random walk — no pattern to exploit

Sophisticated models shine when data has real structure — random walks defeat all methods equally

## Quantifying forecast uncertainty

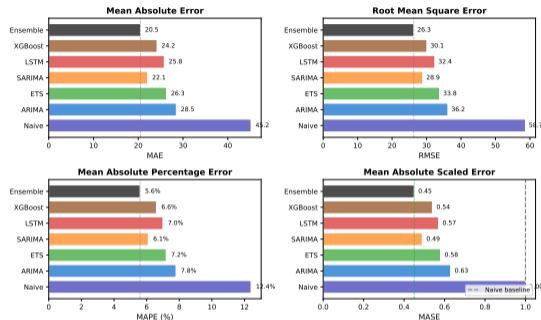
- Point forecasts alone are not enough — always report uncertainty ranges
- 80% prediction interval: the likely range; 95% interval: the conservative range
- Intervals widen with forecast horizon — uncertainty grows the further ahead we predict
- Narrow intervals may indicate overconfidence; always validate interval coverage



Forecast uncertainty grows with horizon — always report prediction intervals alongside point estimates

## Measuring forecast quality

- MAE (Mean Absolute Error): average absolute forecast miss — easy to interpret
- RMSE (Root Mean Squared Error): penalizes large errors more than MAE
- MAPE (Mean Absolute Percentage Error): scale-free but undefined when actuals are zero
- MASE (Mean Absolute Scaled Error): compares to naive — MASE < 1 means better than naive

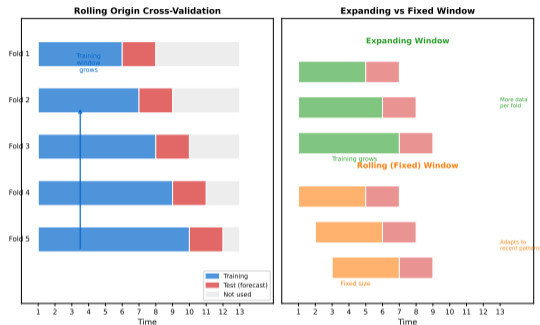


Interpretation: Lower values = better forecast accuracy | MASE < 1 means better than naive | Ensemble typically wins

**MASE less than 1 means better than naive — use it as the universal performance benchmark**

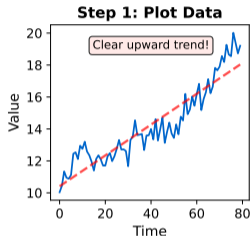
## Respecting temporal order

- Standard cross-validation shuffles data — this destroys temporal order in time series
- Rolling origin CV: train up to time  $t$ , forecast  $h$  steps ahead, move window forward, repeat
- Expanding window uses all past data; rolling window uses a fixed-length training set
- CV gives honest out-of-sample error estimates across many forecast origins



Time series CV respects temporal order — never shuffle sequential data

## Complete Example: Steps 1-3



## Step 2: Test Stationarity

### ADF Test Results:

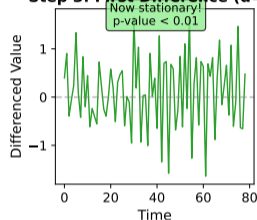
Test statistic: -1.24

**p-value: 0.65**

Critical (5%): -2.89

**Conclusion: NON-STATIONARY**

## Step 3: First Difference (d=1)



## Key Observations:

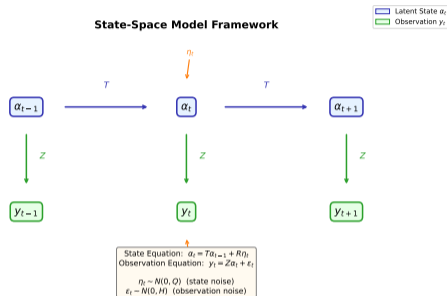
- Step 1: Plot the data and identify visual patterns (trend, seasonality, outliers)
- Step 2: Test stationarity and difference as needed until the ADF test rejects
- Step 3: Examine ACF/PACF, fit candidate models, select by AIC
- Step 4: Verify residuals are white noise, then generate forecasts with prediction intervals

The complete workflow: plot, test stationarity, identify orders, fit, diagnose, forecast

## Advanced Topics

## Separating observation from truth

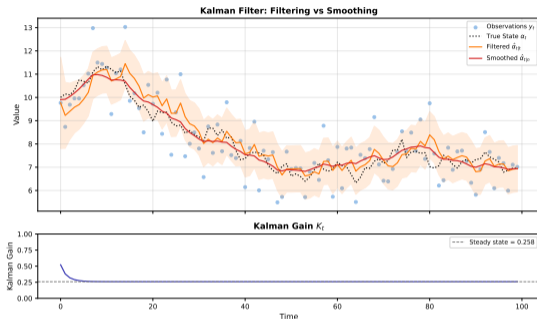
- State-space models separate the observed data (with noise) from the hidden true state
- Two equations: observation equation (what we see) and state equation (how the hidden state evolves)
- Many classical models (ARIMA, ETS) can be written in state-space form
- Example: a thermometer reading (observed) vs. the true temperature (hidden state)



State-space models separate what we observe from the hidden truth — a powerful general framework

## Predict, then update

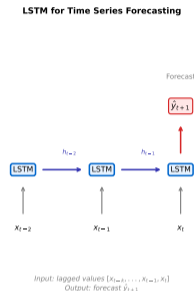
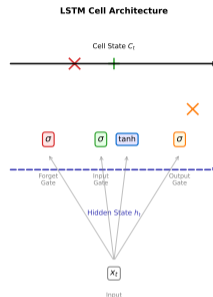
- The Kalman filter is a two-step algorithm: predict the next state, then update with new data
- Prediction step: use the state equation to forecast the hidden state forward
- Update step: adjust the prediction using the observed data (weighted by uncertainty)
- Applications: GPS navigation, robotics, economics, and real-time signal processing



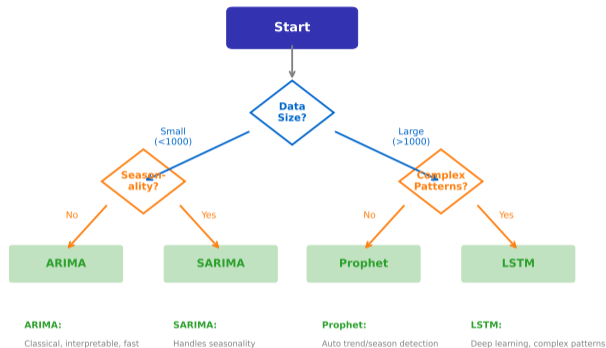
The Kalman filter optimally tracks hidden states by balancing prediction with observation

## Deep learning for sequences

- LSTM (Long Short-Term Memory) networks are deep learning models designed for sequences
- Gates control what to remember and what to forget — capturing long-range dependencies
- Advantages: learn non-linear patterns automatically; handle complex, multivariate inputs
- Trade-off: need large datasets, longer training, and less interpretable than ARIMA



**LSTM networks learn non-linear patterns automatically but need large data and sacrifice interpretability**



## Key Observations:

- Small data + linear patterns → ARIMA/SARIMA (interpretable, fast)
- Moderate data + seasonality → ETS or SARIMA (automatic with AIC)
- Large data + non-linear patterns → LSTM or ensemble methods
- Start simple, add complexity only when justified by performance

Start simple (ARIMA), add complexity only when justified by better out-of-sample performance

## Summary and Key Takeaways

## Strengths

- + ARIMA/SARIMA: Interpretable, fast, works well for linear trends and seasonal data with moderate sample sizes
- + ETS: Automatic, robust, excellent for production forecasting when interpretability is secondary

## Limitations

- GARCH: Essential for financial volatility but adds complexity — use only when variance is clearly time-varying
- LSTM: Powerful for non-linear patterns but needs large data, careful tuning, and sacrifices interpretability

---

Start with ARIMA for interpretability; use GARCH for finance; try LSTM only with large, complex data

## Key Takeaways

- Time series analysis requires stationarity — always test (ADF + KPSS) and difference before modeling
- The Box-Jenkins methodology provides a systematic workflow: plot → test → identify → fit → diagnose → forecast
- Never trust a model that cannot beat the naive benchmark — use  $MASE < 1$  as the minimum bar
- Forecast uncertainty grows with horizon — always report prediction intervals alongside point forecasts

---

Forecasting is a blend of statistical rigor and domain knowledge — always validate, never trust blindly