



# Time Series Analysis and Forecasting

Bucharest University of Economic Studies

Academic Year 2025-2026



# Outline

- 1 Introduction and Lag Operator
- 2 Autoregressive (AR) Models
- 3 Moving Average (MA) Models
- 4 ARMA Models
- 5 Model Identification
- 6 Parameter Estimation
- 7 Model Diagnostics
- 8 Forecasting with ARMA
- 9 Practical Implementation
- 10 Summary

## Recap: Stationarity

**From Chapter 1:** A process  $\{X_t\}$  is **weakly stationary** if:

- ①  $\mathbb{E}[X_t] = \mu$  (constant mean)
- ②  $\text{Var}(X_t) = \sigma^2 < \infty$  (constant, finite variance)
- ③  $\text{Cov}(X_t, X_{t+h}) = \gamma(h)$  (covariance depends only on lag  $h$ )

**Why stationarity matters for ARMA:**

- ARMA models assume the underlying process is stationary
- Non-stationary data must be differenced first (ARIMA)
- Stationarity ensures stable model parameters

**Today:** We build models for stationary time series using past values and past errors.

# The Lag Operator (Backshift Operator)

## Definition 1 (Lag Operator)

The **lag operator**  $L$  (or backshift operator  $B$ ) shifts a time series back by one period:

$$LX_t = X_{t-1}$$

### Properties:

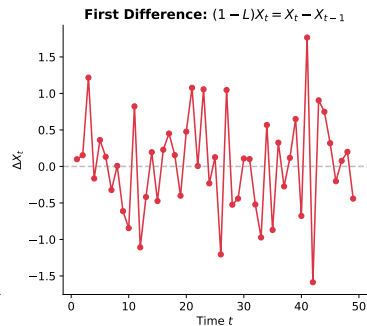
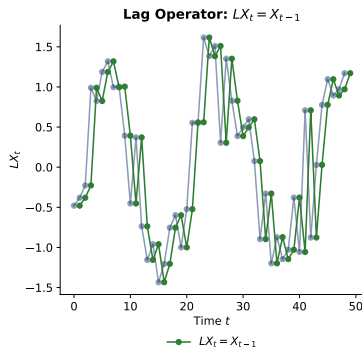
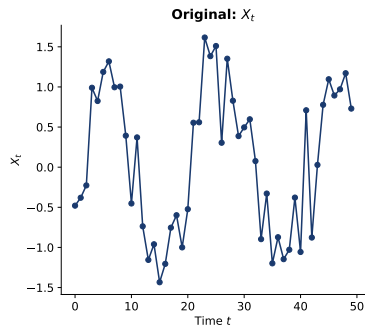
- $L^k X_t = X_{t-k}$  (shift back  $k$  periods)
- $L^0 X_t = X_t$  (identity)
- $(1 - L)X_t = X_t - X_{t-1} = \Delta X_t$  (first difference)
- $(1 - L)^d X_t = \Delta^d X_t$  ( $d$ -th difference)

### Lag Polynomials:

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p$$

$$\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \dots + \theta_q L^q$$

# Lag Operator: Visual Illustration



**Key insight:** The lag operator is the foundation of ARMA model notation

## Definition 2 (White Noise)

A process  $\{\varepsilon_t\}$  is **white noise**, denoted  $\varepsilon_t \sim WN(0, \sigma^2)$ , if:

- ①  $\mathbb{E}[\varepsilon_t] = 0$  for all  $t$
- ②  $\text{Var}(\varepsilon_t) = \sigma^2$  for all  $t$
- ③  $\text{Cov}(\varepsilon_t, \varepsilon_s) = 0$  for all  $t \neq s$

## Properties:

- White noise is the “building block” of ARMA models
- ACF:  $\rho(0) = 1$ ,  $\rho(h) = 0$  for  $h \neq 0$
- PACF: same pattern
- **Gaussian white noise:** additionally  $\varepsilon_t \sim N(0, \sigma^2)$

**Note:** White noise is *not* predictable — it's pure randomness.

## AR(1) Model: Definition

### Definition 3 (AR(1) Process)

An **autoregressive process of order 1** is:

$$X_t = c + \phi X_{t-1} + \varepsilon_t$$

where  $\varepsilon_t \sim WN(0, \sigma^2)$  and  $|\phi| < 1$  for stationarity.

### Interpretation:

- $c$ : constant (intercept)
- $\phi$ : autoregressive coefficient — measures persistence
- $\varepsilon_t$ : innovation (unpredictable shock)

### Using lag operator:

$$(1 - \phi L)X_t = c + \varepsilon_t$$

$$\phi(L)X_t = c + \varepsilon_t \quad \text{where } \phi(L) = 1 - \phi L$$

## AR(1) Stationarity Condition

**For AR(1) to be stationary:**  $|\phi| < 1$

### Intuition:

- If  $|\phi| < 1$ : shocks decay over time  $\rightarrow$  stationary
- If  $|\phi| = 1$ : random walk  $\rightarrow$  non-stationary (unit root)
- If  $|\phi| > 1$ : explosive process  $\rightarrow$  non-stationary

### Characteristic equation:

$$\phi(z) = 1 - \phi z = 0 \implies z = \frac{1}{\phi}$$

Stationarity requires the root  $z = 1/\phi$  to lie **outside the unit circle**, i.e.,  $|z| > 1$ , which means  $|\phi| < 1$ .



## AR(1) Properties

For a stationary AR(1) with  $|\phi| < 1$ :

**Mean:**

$$\mu = \mathbb{E}[X_t] = \frac{c}{1 - \phi}$$

**Variance:**

$$\gamma(0) = \text{Var}(X_t) = \frac{\sigma^2}{1 - \phi^2}$$

**Autocovariance:**

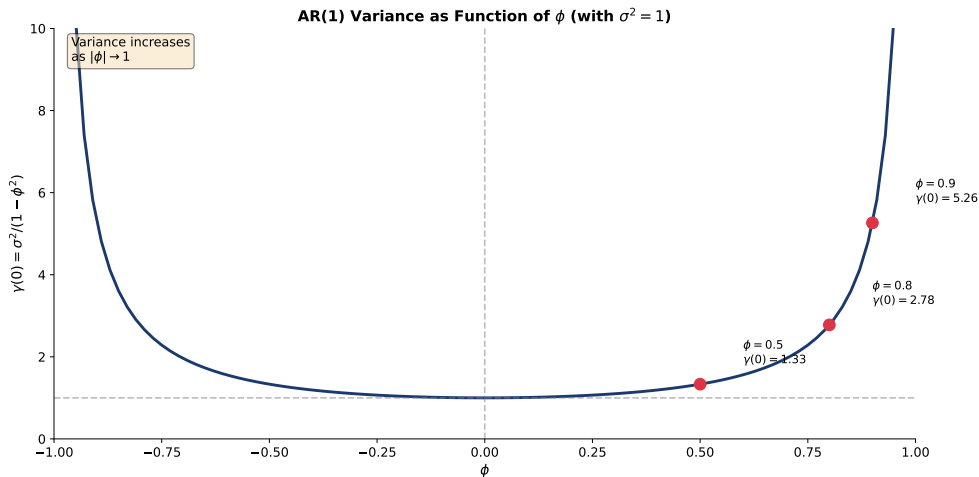
$$\gamma(h) = \phi^h \gamma(0) = \frac{\phi^h \sigma^2}{1 - \phi^2}$$

**Autocorrelation (ACF):**

$$\rho(h) = \phi^h$$

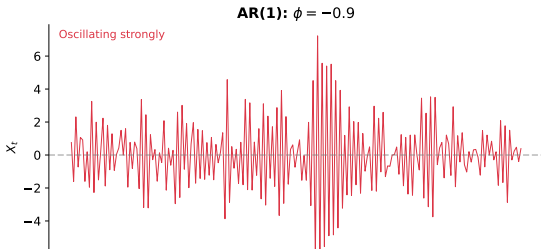
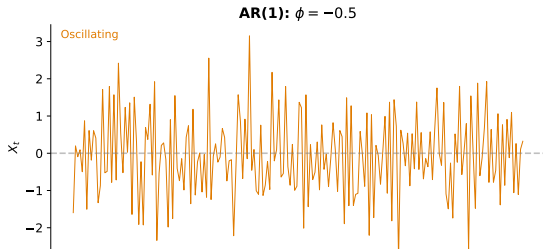
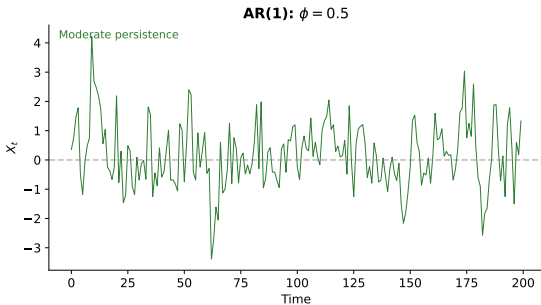
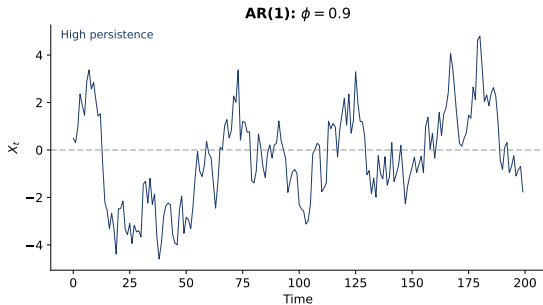
**Key insight:** ACF decays exponentially at rate  $\phi$

## AR(1) Variance as Function of $\phi$

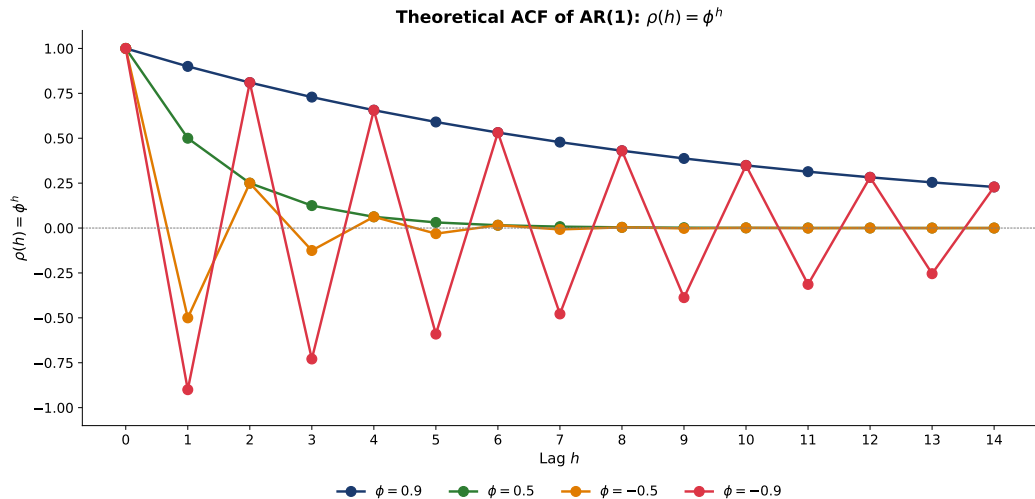


**Key insight:** As  $|\phi| \rightarrow 1$ , variance explodes  $\rightarrow$  non-stationarity

# AR(1) Simulations: Effect of $\phi$



# AR(1) Theoretical ACF



**Pattern:**  $\rho(h) = \phi^h$  — exponential decay (or alternating for  $\phi < 0$ )

## AR(1) ACF and PACF Patterns

### ACF of AR(1):

- Decays exponentially:  $\rho(h) = \phi^h$
- If  $\phi > 0$ : all positive, gradual decay
- If  $\phi < 0$ : alternating signs, decay in magnitude

### PACF of AR(1):

- **Cuts off after lag 1**
- $\pi_1 = \phi$ ,  $\pi_k = 0$  for  $k > 1$

ACF		PACF
AR(1)	Exponential decay	Cuts off at lag 1

**This is the key identification pattern for AR(1)!**

## AR(p) Model: General Form

### Definition 4 (AR(p) Process)

An autoregressive process of order  $p$  is:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t$$

Using lag operator:

$$\phi(L)X_t = c + \varepsilon_t$$

where  $\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p$

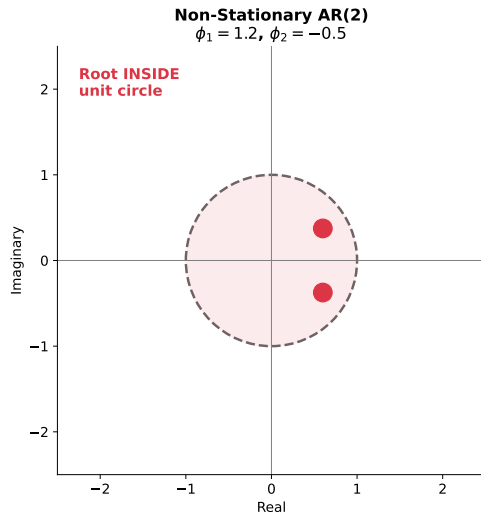
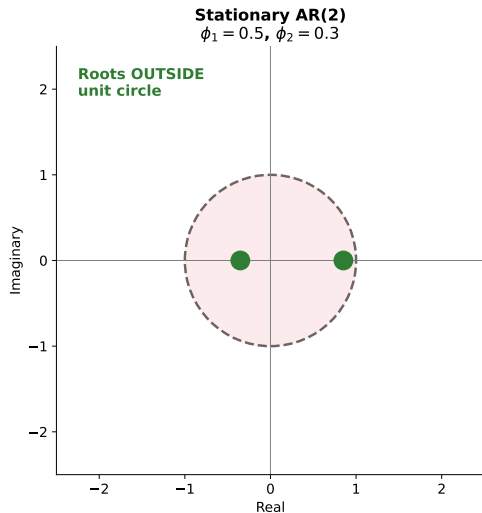
**Stationarity condition:**

- All roots of  $\phi(z) = 0$  must lie **outside** the unit circle
- Equivalently: all roots have modulus  $> 1$

**PACF pattern:**

- PACF cuts off after lag  $p$
- ACF decays (exponentially or with damped oscillations)

## AR(2) Stationarity: Unit Circle Visualization



**Rule:** All roots of  $\phi(z) = 0$  must lie **outside** the shaded unit circle

### Definition 5 (AR(2) Process)

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t$$

#### Stationarity conditions for AR(2):

- ❶  $\phi_1 + \phi_2 < 1$
- ❷  $\phi_2 - \phi_1 < 1$
- ❸  $|\phi_2| < 1$

#### ACF behavior depends on roots:

- **Real roots:** mixture of two exponential decays
- **Complex roots:** damped sinusoidal pattern (pseudo-cycles)

**PACF:** Cuts off after lag 2 ( $\pi_k = 0$  for  $k > 2$ )



## MA(1) Model: Definition

### Definition 6 (MA(1) Process)

A **moving average process of order 1** is:

$$X_t = \mu + \varepsilon_t + \theta\varepsilon_{t-1}$$

where  $\varepsilon_t \sim WN(0, \sigma^2)$ .

### Interpretation:

- $\mu$ : mean of the process
- $\theta$ : MA coefficient — measures impact of past shock
- Current value depends on current and one past shock

### Using lag operator:

$$X_t = \mu + \theta(L)\varepsilon_t$$

where  $\theta(L) = 1 + \theta L$

**Key property:** MA processes are **always stationary** for any finite  $\theta$

## MA(1) Properties

For MA(1):  $X_t = \mu + \varepsilon_t + \theta\varepsilon_{t-1}$

**Mean:**

$$\mathbb{E}[X_t] = \mu$$

**Variance:**

$$\gamma(0) = \text{Var}(X_t) = \sigma^2(1 + \theta^2)$$

**Autocovariance:**

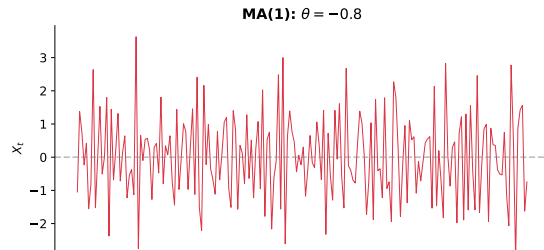
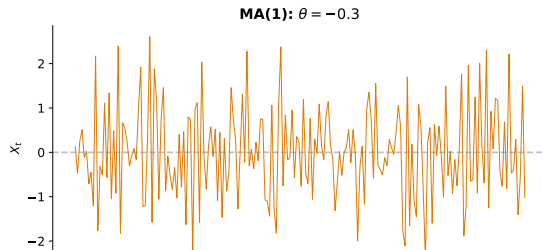
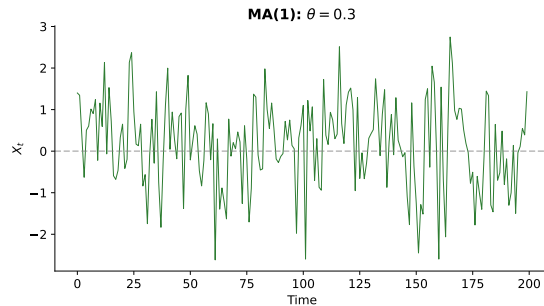
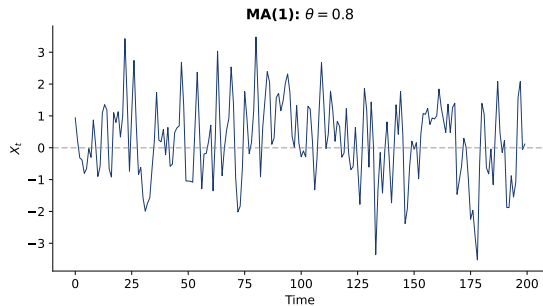
$$\gamma(1) = \theta\sigma^2, \quad \gamma(h) = 0 \text{ for } h > 1$$

**Autocorrelation (ACF):**

$$\rho(1) = \frac{\theta}{1 + \theta^2}, \quad \rho(h) = 0 \text{ for } h > 1$$

**Key insight:** ACF **cuts off** after lag 1

## MA(1) Simulations: Effect of $\theta$



## MA(1) ACF and PACF Patterns

### ACF of MA(1):

- Cuts off after lag 1
- $\rho(1) = \frac{\theta}{1+\theta^2}$ ,  $\rho(h) = 0$  for  $h > 1$
- Note:  $|\rho(1)| \leq 0.5$  always (maximum at  $\theta = \pm 1$ )

### PACF of MA(1):

- Decays exponentially (or with alternating signs)
- Does *not* cut off

	ACF	PACF
MA(1)	Cuts off at lag 1	Exponential decay

**This is the opposite pattern from AR(1)!**

# Invertibility of MA Models

## Definition 7 (Invertibility)

An MA process is **invertible** if it can be written as an infinite AR process:

$$X_t = \mu + \sum_{j=1}^{\infty} \pi_j (X_{t-j} - \mu) + \varepsilon_t$$

**For MA(1):** Invertible if  $|\theta| < 1$

**For MA(q):** All roots of  $\theta(z) = 0$  must lie outside the unit circle

**Why invertibility matters:**

- Ensures unique representation
- Required for forecasting and estimation
- Creates correspondence:  $AR(\infty) \leftrightarrow MA(q)$

**Note:** Stationarity is for AR, Invertibility is for MA

### Definition 8 (MA(q) Process)

A moving average process of order  $q$  is:

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

Using lag operator:

$$X_t = \mu + \theta(L)\varepsilon_t$$

where  $\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q$

**Properties:**

- Always stationary (finite variance)
- ACF cuts off after lag  $q$ :  $\rho(h) = 0$  for  $h > q$
- PACF decays gradually
- Invertible if all roots of  $\theta(z) = 0$  lie outside unit circle

## ARMA(p,q) Model: Definition

### Definition 9 (ARMA(p,q) Process)

An **autoregressive moving average process** of order (p,q) is:

$$X_t = c + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

**Compact form using lag operators:**

$$\phi(L)(X_t - \mu) = \theta(L)\varepsilon_t$$

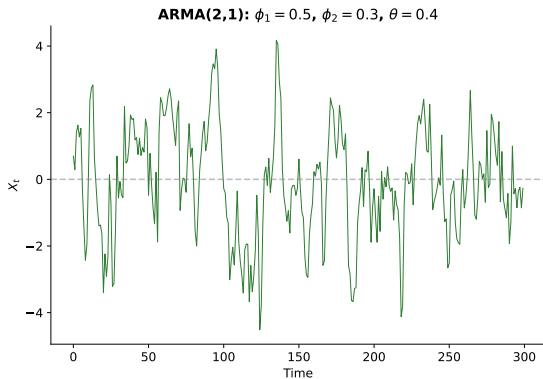
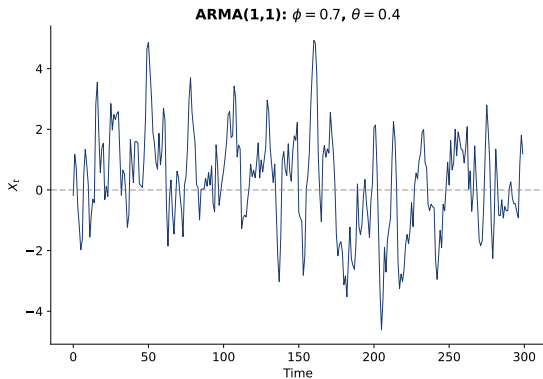
or equivalently:

$$\phi(L)X_t = c + \theta(L)\varepsilon_t$$

where  $\mu = \frac{c}{1 - \phi_1 - \cdots - \phi_p}$

**Key idea:** Combines AR and MA components for more flexible modeling

# ARMA Examples





## ARMA(1,1) Model

### Definition 10 (ARMA(1,1) Process)

$$X_t = c + \phi X_{t-1} + \varepsilon_t + \theta \varepsilon_{t-1}$$

**Properties (assuming stationarity and invertibility):**

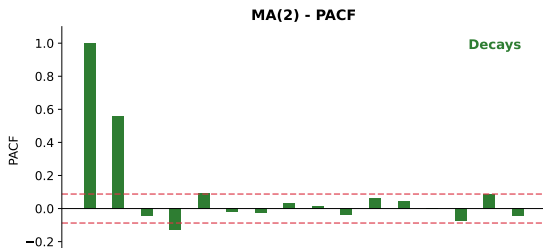
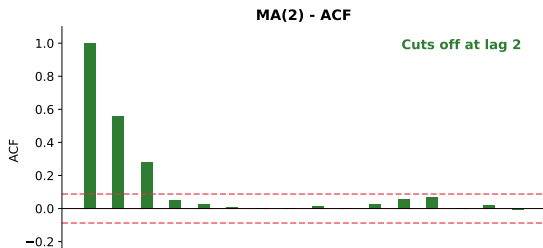
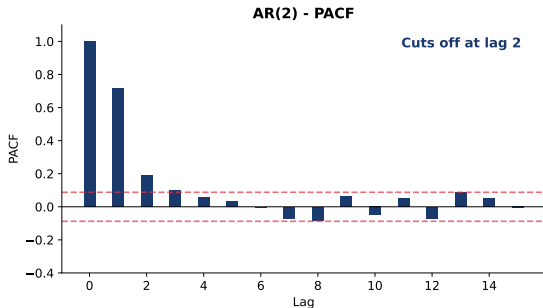
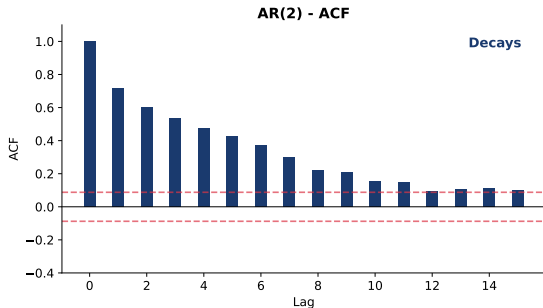
- Mean:  $\mu = \frac{c}{1-\phi}$
- Variance:  $\gamma(0) = \frac{(1+2\phi\theta+\theta^2)\sigma^2}{1-\phi^2}$

**ACF:**

$$\rho(1) = \frac{(1 + \phi\theta)(\phi + \theta)}{1 + 2\phi\theta + \theta^2}$$
$$\rho(h) = \phi \cdot \rho(h-1) \quad \text{for } h \geq 2$$

**Pattern:** ACF decays exponentially after lag 1 (like AR), but starting point depends on both  $\phi$  and  $\theta$

# ACF/PACF Patterns: AR vs MA vs ARMA



## ARMA ACF and PACF Patterns

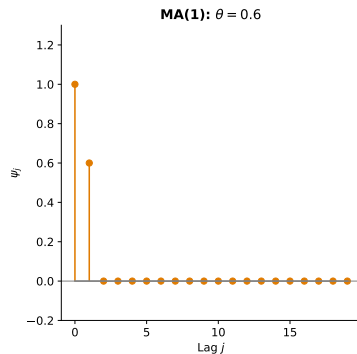
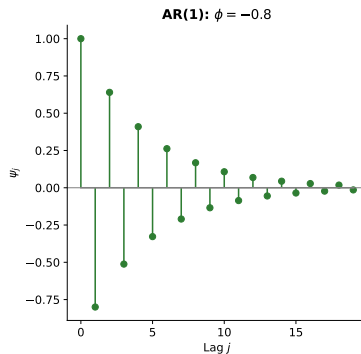
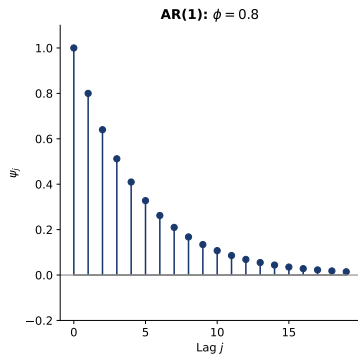
Model	ACF	PACF
AR( $p$ )	Decays (exp./damped)	Cuts off at lag $p$
MA( $q$ )	Cuts off at lag $q$	Decays (exp./damped)
ARMA( $p, q$ )	Decays after lag $q - p$	Decays after lag $p - q$

### Key identification rule:

- **PACF cuts off**  $\rightarrow$  AR process (order = cutoff lag)
- **ACF cuts off**  $\rightarrow$  MA process (order = cutoff lag)
- **Both decay**  $\rightarrow$  ARMA process

**Caution:** In practice, sample ACF/PACF are noisy; use confidence bands

# Impulse Response Functions



**Interpretation:** Shows how a unit shock propagates through the system over time

# Stationarity and Invertibility Summary

For ARMA(p,q) to be well-behaved:

Condition	Requirement
Stationarity	Roots of $\phi(z) = 0$ outside unit circle
Invertibility	Roots of $\theta(z) = 0$ outside unit circle

Implications:

- **Stationarity:** Can write as MA( $\infty$ ):  $X_t = \mu + \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$
- **Invertibility:** Can write as AR( $\infty$ ):  $X_t = \mu + \sum_{j=1}^{\infty} \pi_j (X_{t-j} - \mu) + \varepsilon_t$

**Causal representation:**  $X_t$  depends only on *past* shocks (not future)

# The Box-Jenkins Methodology

## Box-Jenkins Methodology

**Step 1: IDENTIFICATION**  
Analyze ACF/PACF  
Select  $p$ ,  $d$ ,  $q$



**Step 2: ESTIMATION**  
Estimate parameters  
(MLE or Yule-Walker)



**Step 3: DIAGNOSTIC**  
Check residuals  
Ljung-Box test



Residuals  
White Noise?

No  
Revise  
model



Yes

**Step 4: FORECASTING**  
Generate predictions  
with confidence intervals

# Model Identification Summary Table

## Model Identification: ACF/PACF Patterns

Model	ACF Pattern	PACF Pattern
<b>AR(p)</b>	Exponential decay or damped oscillation	Cuts off after lag p
<b>MA(q)</b>	Cuts off after lag q	Exponential decay or damped oscillation
<b>ARMA(p,q)</b>	Exponential decay after lag q-p	Exponential decay after lag p-q

Practical tip: Start simple (low p, q), increase if diagnostic fail

## ACF/PACF Identification Rules

**Theoretical patterns for stationary processes:**

Model	ACF Pattern	PACF Pattern
AR(1)	Exponential decay	Spike at lag 1, then 0
AR(2)	Damped exponential/sine	Spikes at lags 1-2, then 0
AR( $p$ )	Decays gradually	Cuts off after lag $p$
MA(1)	Spike at lag 1, then 0	Exponential decay
MA(2)	Spikes at lags 1-2, then 0	Damped exponential/sine
MA( $q$ )	Cuts off after lag $q$	Decays gradually
ARMA( $p,q$ )	Decays	Decays



## Information Criteria

**Purpose:** Balance goodness-of-fit against model complexity

**Akaike Information Criterion (AIC):**

$$\text{AIC} = -2\ln(\hat{L}) + 2k$$

**Bayesian Information Criterion (BIC/SBC):**

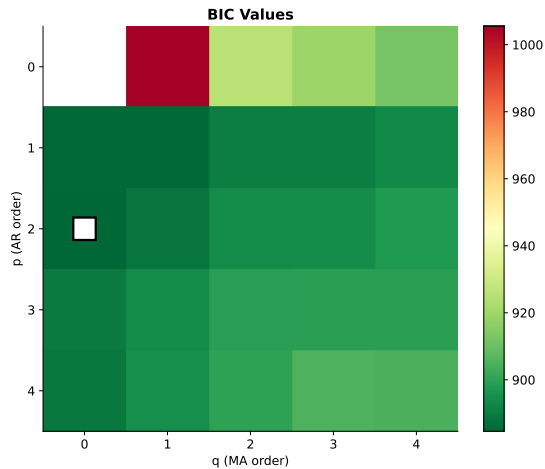
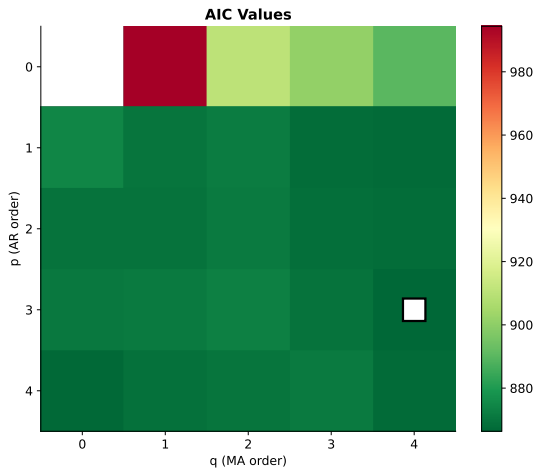
$$\text{BIC} = -2\ln(\hat{L}) + k \ln(n)$$

where  $\hat{L}$  = maximized likelihood,  $k$  = number of parameters,  $n$  = sample size

**Usage:**

- Lower values are better
- BIC penalizes complexity more strongly than AIC
- AIC tends to choose larger models; BIC more parsimonious
- Compare models fit to the *same data*

## AIC vs BIC: Model Selection



**Note:** White square marks the best model; lower values (green) are better

# Automatic Model Selection

## Grid search approach:

- 1 Fit ARMA( $p, q$ ) for  $p = 0, 1, \dots, p_{max}$  and  $q = 0, 1, \dots, q_{max}$
- 2 Select model with lowest AIC or BIC
- 3 Verify with diagnostic checks

## In Python (statsmodels):

- `pm.auto_arima()` from `pmdarima` package
- Automatically tests stationarity, searches over orders
- Returns best model by AIC/BIC

## Caution:

- Automatic selection is a starting point, not final answer
- Always check diagnostics
- Consider domain knowledge

## Three main approaches:

### 1. Method of Moments / Yule-Walker (AR only)

- Match sample autocorrelations to theoretical values
- Simple, closed-form for AR models
- Not efficient for MA components

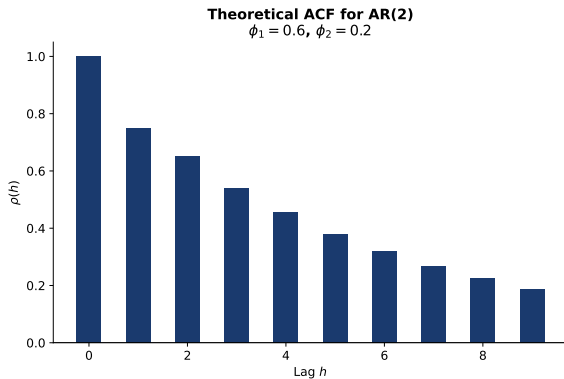
### 2. Maximum Likelihood Estimation (MLE)

- Most common approach
- Requires distributional assumption (usually Gaussian)
- Efficient and consistent

### 3. Conditional Least Squares

- Minimize sum of squared residuals
- Conditioning on initial observations
- Computationally simpler than exact MLE

# Yule-Walker Equations for AR(p)



## Yule-Walker Equations

$$\rho(1) = \phi_1 + \phi_2 \rho(1)$$

$$\rho(2) = \phi_1 \rho(1) + \phi_2$$

Matrix form:  $R \cdot \phi = \rho$

$R$  = autocorrelation matrix

$$\text{Solution: } \hat{\phi} = R^{-1} \rho$$

## Yule-Walker Equations: Matrix Form

For AR(p):  $X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t$

**Yule-Walker equations:**

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2) + \cdots + \phi_p \rho(k-p)$$

for  $k = 1, 2, \dots, p$

**Matrix form:**

$$\begin{pmatrix} \rho(0) & \rho(1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0) \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{pmatrix}$$

**Estimation:** Replace  $\rho(k)$  with sample autocorrelations  $\hat{\rho}(k)$

# Maximum Likelihood Estimation

**Assuming Gaussian errors:**  $\varepsilon_t \sim N(0, \sigma^2)$

**Log-likelihood for ARMA(p,q):**

$$\ell(\phi, \theta, \sigma^2) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^n \varepsilon_t^2$$

where  $\varepsilon_t$  are the innovations computed recursively.

**Estimation procedure:**

- 1 Initialize: use method of moments or OLS for starting values
- 2 Optimize: numerical methods (e.g., BFGS, Newton-Raphson)
- 3 Iterate until convergence

**In practice:** Use `statsmodels.tsa.arima.model.ARIMA`

### Asymptotic distribution of MLE:

$$\hat{\theta} \xrightarrow{d} N\left(\theta_0, \frac{1}{n} \mathbf{I}(\theta_0)^{-1}\right)$$

where  $\mathbf{I}(\theta)$  is the Fisher information matrix.

**Standard errors:** Square root of diagonal of  $\frac{1}{n} \hat{\mathbf{I}}^{-1}$

### Hypothesis testing:

- $H_0 : \phi_j = 0$  (or  $\theta_j = 0$ )
- Test statistic:  $z = \frac{\hat{\phi}_j}{SE(\hat{\phi}_j)} \sim N(0, 1)$  asymptotically
- Reject if  $|z| > 1.96$  at 5% level

**Confidence interval:**  $\hat{\phi}_j \pm 1.96 \cdot SE(\hat{\phi}_j)$



**If model is correctly specified, residuals should be white noise:**

## **1. Plot residuals over time**

- Should fluctuate around zero
- No obvious patterns or trends
- Constant variance (no heteroskedasticity)

## **2. Check ACF of residuals**

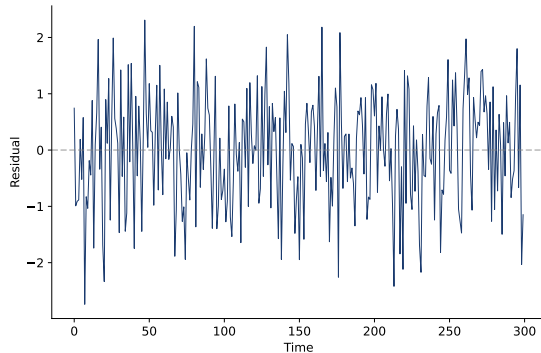
- All correlations should be within confidence bands
- No significant spikes  $\rightarrow$  white noise

## **3. Check histogram / Q-Q plot**

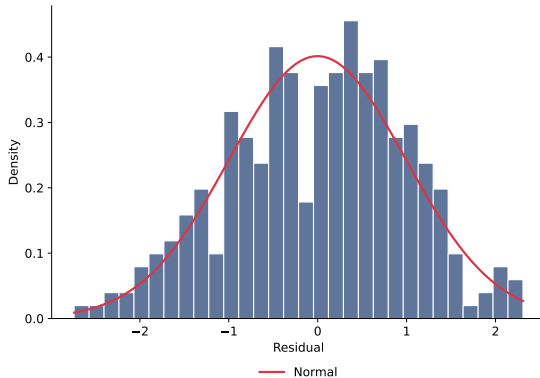
- Should be approximately normal (if assuming Gaussian)
- Heavy tails suggest non-normal errors

# Residual Diagnostics: Example

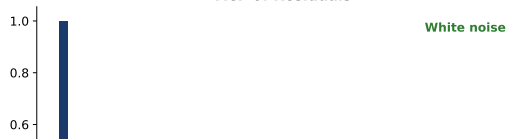
**Residuals over Time**



**Residual Distribution**



**ACF of Residuals**



**Q-Q Plot**



## Definition 11 (Ljung-Box Test)

Tests whether residuals are independently distributed (no autocorrelation).

**Test statistic:**

$$Q(m) = n(n+2) \sum_{k=1}^m \frac{\hat{\rho}_k^2}{n-k}$$

**Hypotheses:**

- $H_0$ : Residuals are white noise (no autocorrelation up to lag  $m$ )
- $H_1$ : Residuals are autocorrelated

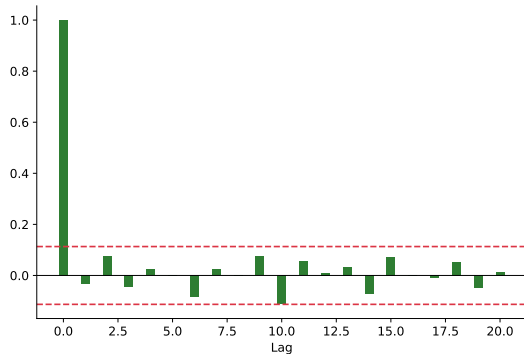
**Distribution:** Under  $H_0$ ,  $Q(m) \sim \chi^2(m-p-q)$  approximately

**Decision:**

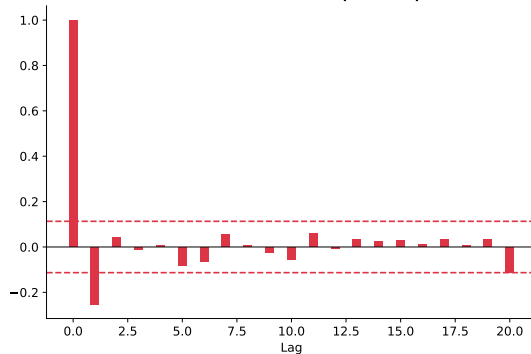
- p-value  $> 0.05 \rightarrow$  fail to reject  $H_0 \rightarrow$  residuals look like white noise (good!)
- p-value  $< 0.05 \rightarrow$  significant autocorrelation remains  $\rightarrow$  model inadequate

# Ljung-Box Test: Good vs Bad Model Fit

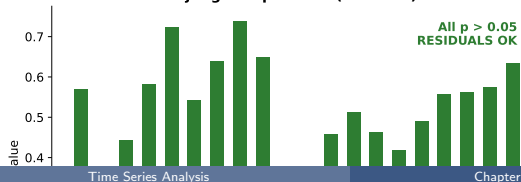
Good Fit: Residual ACF



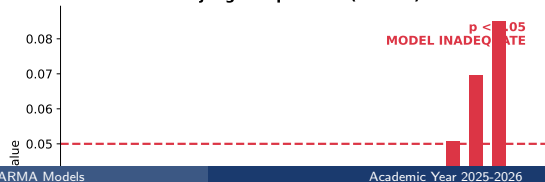
Bad Fit: Residual ACF (Underfit)



Ljung-Box p-values (Good Fit)



Ljung-Box p-values (Bad Fit)



# Diagnostic Checklist

**A good ARMA model should satisfy:**

- ① **Stationarity:** AR roots outside unit circle  
✓ Check with `arroots`
- ② **Invertibility:** MA roots outside unit circle  
✓ Check with `maroots`
- ③ **White noise residuals:** No significant ACF  
✓ ACF plot, Ljung-Box test
- ④ **Normal residuals:** (if assumed)  
✓ Q-Q plot, Jarque-Bera test
- ⑤ **No heteroskedasticity:** Constant variance  
✓ Plot residuals, ARCH test
- ⑥ **Parsimonious:** Lowest AIC/BIC among adequate models

**If diagnostics fail:** Return to identification, try different orders

**Optimal forecast:** Conditional expectation minimizes MSE

$$\hat{X}_{n+h|n} = \mathbb{E}[X_{n+h}|X_n, X_{n-1}, \dots]$$

**For AR(1):**  $X_t = c + \phi X_{t-1} + \varepsilon_t$

$$\hat{X}_{n+1|n} = c + \phi X_n$$

$$\hat{X}_{n+2|n} = c + \phi \hat{X}_{n+1|n} = c(1 + \phi) + \phi^2 X_n$$

$$\hat{X}_{n+h|n} = \mu + \phi^h (X_n - \mu)$$

**Key property:** Forecasts converge to mean  $\mu$  as  $h \rightarrow \infty$

**For MA(1):**  $X_t = \mu + \varepsilon_t + \theta \varepsilon_{t-1}$

$$\hat{X}_{n+1|n} = \mu + \theta \varepsilon_n$$

$$\hat{X}_{n+h|n} = \mu \quad \text{for } h > 1$$

# Forecast Uncertainty

**Forecast error:**

$$e_{n+h|n} = X_{n+h} - \hat{X}_{n+h|n}$$

**Mean squared forecast error (MSFE):**

$$\text{MSFE}(h) = \mathbb{E}[e_{n+h|n}^2] = \sigma^2 \sum_{j=0}^{h-1} \psi_j^2$$

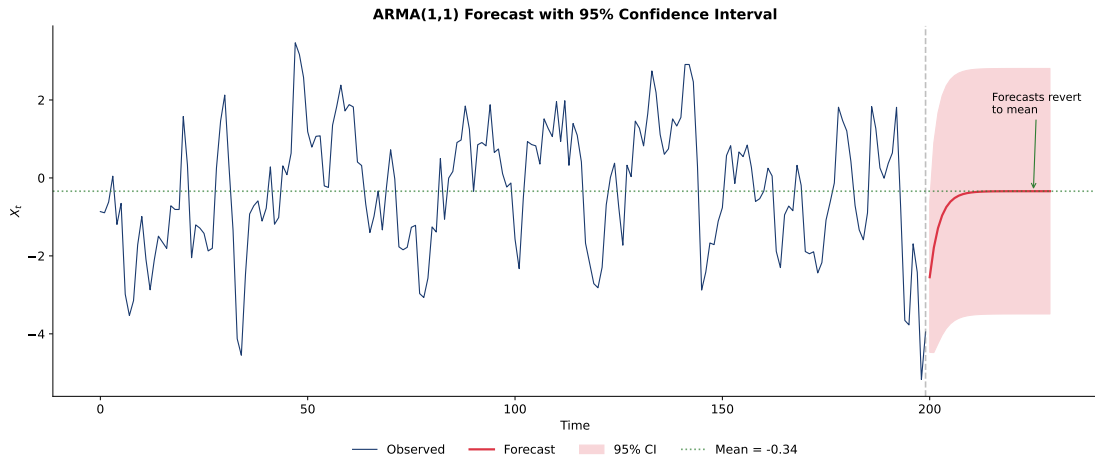
where  $\psi_j$  are the  $\text{MA}(\infty)$  coefficients.

**For AR(1):**  $\psi_j = \phi^j$

$$\text{MSFE}(h) = \sigma^2 \frac{1 - \phi^{2h}}{1 - \phi^2} \rightarrow \frac{\sigma^2}{1 - \phi^2} = \text{Var}(X_t)$$

**Key insight:** Forecast uncertainty increases with horizon, eventually reaching unconditional variance

# ARMA Forecasting with Confidence Intervals





# Confidence Intervals for Forecasts

**Assuming Gaussian errors:**

$$X_{n+h}|X_n, \dots \sim N\left(\hat{X}_{n+h|n}, \text{MSFE}(h)\right)$$

**$(1 - \alpha)$  confidence interval:**

$$\hat{X}_{n+h|n} \pm z_{\alpha/2} \cdot \sqrt{\text{MSFE}(h)}$$

where  $z_{\alpha/2} = 1.96$  for 95% CI.

**Properties:**

- Intervals widen as horizon increases
- Eventually converge to unconditional interval:  $\mu \pm z_{\alpha/2}\sigma_X$
- Width depends on model parameters (AR coefficients, etc.)

**In Python:** `model.get_forecast(h).conf_int()`

# Forecast Evaluation

## Out-of-sample testing:

- 1 Split data: training set (fit model) and test set (evaluate)
- 2 Generate forecasts for test period
- 3 Compare forecasts to actual values

## Metrics (from Chapter 1):

- $MAE = \frac{1}{n} \sum |e_t|$
- $RMSE = \sqrt{\frac{1}{n} \sum e_t^2}$
- $MAPE = \frac{100}{n} \sum \left| \frac{e_t}{\hat{X}_t} \right|$

## Rolling/expanding window:

- Re-estimate model as new data arrives
- More realistic assessment of forecast performance

# Python Implementation: Fitting ARMA

## Using statsmodels:

```
from statsmodels.tsa.arima.model import ARIMA

# Fit ARMA(2,1) --- note: ARIMA(p,d,q) with d=0
model = ARIMA(data, order=(2, 0, 1))
results = model.fit()

# Summary
print(results.summary())

# Forecasting
forecast = results.get_forecast(steps=10)
print(forecast.predicted_mean)
print(forecast.conf_int())
```

**Note:** ARIMA with  $d = 0$  is equivalent to ARMA

### Automatic ARIMA selection:

```
import pmdarima as pm

# Auto ARIMA with AIC criterion
model = pm.auto_arima(data,
                      start_p=0, max_p=5,
                      start_q=0, max_q=5,
                      d=0, # No differencing for stationary data
                      seasonal=False,
                      information_criterion='aic',
                      trace=True)

print(model.summary())
```

**Output:** Best model order and fitted parameters

# Workflow Summary

## ① Data preparation

- Check for missing values, outliers
- Transform if necessary (log, differencing)

## ② Stationarity check

- Visual inspection: time plot, ACF
- Formal tests: ADF, KPSS
- Difference if non-stationary

## ③ Model identification

- ACF/PACF patterns
- Information criteria grid search

## ④ Estimation and diagnostics

- Fit model, check significance
- Residual analysis, Ljung-Box test

## ⑤ Forecasting

- Point forecasts with confidence intervals
- Out-of-sample validation

# Key Takeaways

- ① **AR( $p$ ) models:** Current value depends on  $p$  past values
  - Stationarity: roots of  $\phi(z)$  outside unit circle
  - PACF cuts off at lag  $p$
- ② **MA( $q$ ) models:** Current value depends on  $q$  past shocks
  - Always stationary; invertibility: roots of  $\theta(z)$  outside unit circle
  - ACF cuts off at lag  $q$
- ③ **ARMA( $p,q$ ):** Combines AR and MA for flexible modeling
  - Both ACF and PACF decay
- ④ **Box-Jenkins:** Identify  $\rightarrow$  Estimate  $\rightarrow$  Diagnose  $\rightarrow$  Forecast
- ⑤ **Diagnostics:** Residuals must be white noise
- ⑥ **Forecasts:** Converge to mean; uncertainty increases with horizon

### Chapter 3: ARIMA and Seasonal Models

- ARIMA(p,d,q): Integrated models for non-stationary data
- Seasonal ARIMA: SARIMA(p,d,q)(P,D,Q)<sub>s</sub>
- Seasonal differencing
- Real-world applications with seasonal patterns

#### Reading:

- Hyndman & Athanasopoulos, *Forecasting: Principles and Practice*, Ch. 9
- Box, Jenkins, Reinsel & Ljung, *Time Series Analysis*, Ch. 3-4