Lecture 3 – ACF / PACF, ARMA Identification, Estimation, Forecasting, and Transition to Nonstationarity

Jiajing Sun

School of Economics and Management
University of Chinese Academy of Sciences

Econometrics and Time Series Methods
Spring 2026

Textbook sequence for this lecture

This lecture is organized to follow the textbook sequence as closely as possible, while matching the course plan for Lecture 3.

1. **Chapter 2:** *Autocorrelation Structure of the ARMA model*
   - autocorrelation coefficients,
   - partial autocorrelation coefficients,
   - using ACF and PACF to determine the order of ARMA models.
2. **Chapter 2:** *Estimation of the ARMA model* and *Statistical Inference for ARMA Model*
   - Yule–Walker estimation,
   - conditional maximum likelihood,
   - AIC/BIC and parsimony.
3. **Chapter 2:** *How to Predict Using the ARMA(p, q) Process?*
   - one-step and multi-step forecasts,
   - residual diagnostics,
   - R workflow.
4. **Chapter 8:** *Deterministic Trend* and the opening of *Unit Root Process*
   - deterministic trend versus stochastic trend,
   - trend-stationary versus difference-stationary thinking.

Lecture goals

- Read ACF and PACF plots as model-identification tools rather than as decorations.
- Connect textbook heuristics (cut-off versus tail-off) to the algebra of AR and MA models.
- Move from identification to estimation, order selection, residual checking, and forecasting in one coherent workflow.
- End by seeing why slow decay, permanent shocks, and drifting levels force us to leave the stationary ARMA world.

Roadmap for the three-hour block

Block 1

ACF / PACF, order selection, and model identification.

Block 2

R-integrated workflow: ARMA estimation, residual diagnostics, and forecasting.

Block 3

Transition to nonstationarity: deterministic trend versus stochastic trend.

Lecture 3 map

ARMA(p, q) as the working model

The textbook introduces the ARMA(p, q) model as

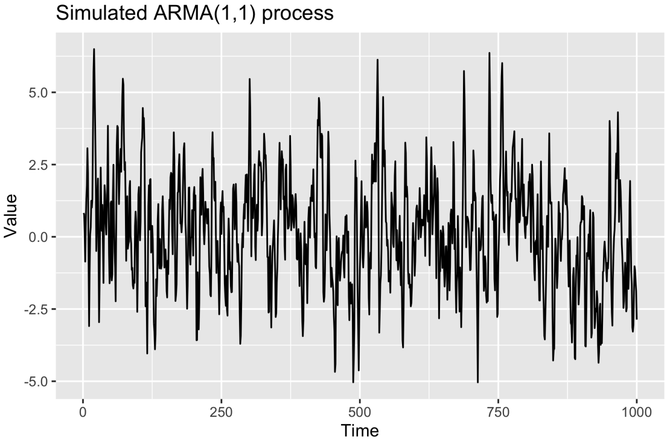
$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}.$$

- ARMA combines the autoregressive and moving-average components in one model.
- It can describe more complex dependence than a pure AR or pure MA model alone.
- The lecture question is practical: *how do we recognize a plausible low-order AR, MA, or ARMA model from data?*

Guiding idea

Model identification starts with second-order dependence: autocovariances, autocorrelations, and partial autocorrelations.

A simulated ARMA(1,1) series



Textbook simulation: $\phi_1 = 0.6$, $\theta_1 = 0.7$, and $\varepsilon_t \sim N(0, 1)$. Persistence is visible, but the order is not.

Autocovariance and autocorrelation

The textbook defines, for any integer k,

$$\gamma_k = \mathrm{Cov}(Y_t, Y_{t-k}) = \mathrm{E}\big[(Y_t - \mu)(Y_{t-k} - \mu)\big], \qquad \rho_k = \frac{\gamma_k}{\gamma_0}.$$

- $\gamma_0 = \mathrm{Var}(Y_t)$ is the variance.
- ρ_k lies between -1 and 1.
- Viewing ρ_k as a function of k gives the **autocorrelation function (ACF)**.
- For weakly stationary series, $\gamma_{-k} = \gamma_k$ and $\rho_{-k} = \rho_k$.

Interpretation

The ACF describes how strongly the series at time t is related to its past values, and therefore how long the impact of a shock lasts in the series.

Autocorrelation structure of AR(1)

For the AR(1) model

$$Y_t = c + \phi_1 Y_{t-1} + \varepsilon_t, \qquad |\phi_1| < 1, \qquad \varepsilon_t \sim WN(0, \sigma^2),$$

stationarity implies

$$\gamma_0 = \frac{\sigma^2}{1 - \phi_1^2}, \qquad \rho_1 = \phi_1, \qquad \rho_k = \phi_1^k.$$

- The ACF decays geometrically.
- The sign of ϕ_1 determines whether the decay alternates in sign.
- The magnitude of $|\phi_1|$ determines how slowly or quickly the decay happens.

Visual lesson

A stationary AR process has an ACF that *tails off*; it does not cut off exactly at a finite lag.

Yule–Walker logic for AR(2) and AR(p)

For

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \varepsilon_t,$$

the textbook derives

$$\gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + \sigma^2, \qquad \gamma_1 = \phi_1\gamma_0 + \phi_2\gamma_1, \qquad \gamma_2 = \phi_1\gamma_1 + \phi_2\gamma_0.$$

For AR(p), taking covariance with Y_{t-k} yields the Yule–Walker recursion

$$\gamma_k = \phi_1\gamma_{k-1} + \phi_2\gamma_{k-2} + \cdots + \phi_p\gamma_{k-p}, \qquad k = 1, 2, 3, \ldots$$

Takeaway

For AR models, the ACF follows a recursive decay pattern determined by the AR coefficients; this is why it tails off rather than cuts off.

Autocorrelation structure of MA models

For the MA(1) model

$$y_t = \theta\varepsilon_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN(0, \sigma^2),$$

we have

$$\rho_1 = \frac{\theta}{1+\theta^2}, \qquad \rho_2 = \rho_3 = \cdots = 0.$$

For the MA(2) model

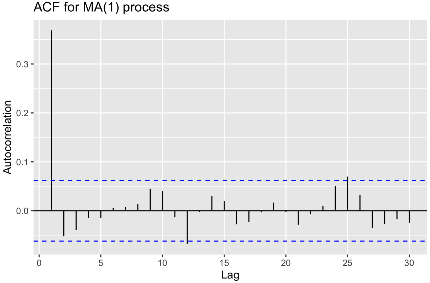
$$y_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2},$$

$$\rho_1 = \frac{\theta_1(1+\theta_2)}{1+\theta_1^2+\theta_2^2}, \qquad \rho_2 = \frac{\theta_2}{1+\theta_1^2+\theta_2^2}, \qquad \rho_3 = \rho_4 = \cdots = 0.$$

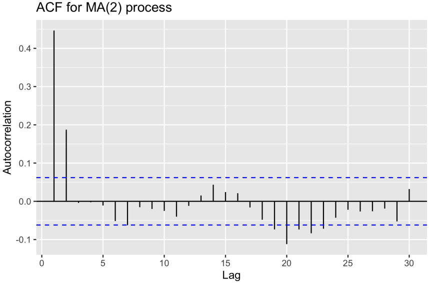
Visual lesson

For MA(q), the ACF is nonzero up to lag q and then **cuts off**.

Empirical ACFs for MA(1) and MA(2)



MA(1)



MA(2)

The textbook uses simulated examples to show the exact cut-off idea in finite samples: after the relevant lag, the sample ACF bars wobble around zero inside the confidence bands.

Partial autocorrelation coefficients (PACF)

The textbook defines the k-th partial autocorrelation coefficient as the last coefficient in an AR(k) regression:

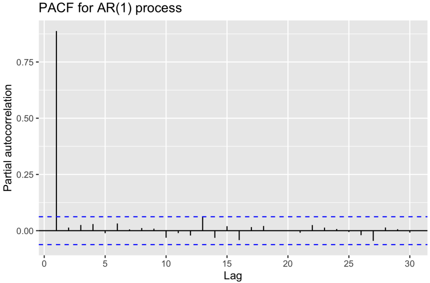
$$\begin{aligned}
\text{AR}(1)\text{:} \quad & Y_t = \delta + \boxed{\phi_1} Y_{t-1} + \varepsilon_t, \quad \widehat{\phi}_{11}; \\
\text{AR}(2)\text{:} \quad & Y_t = \delta + \phi_1 Y_{t-1} + \boxed{\phi_2} Y_{t-2} + \varepsilon_t, \quad \widehat{\phi}_{22}; \\
\text{AR}(k)\text{:} \quad & Y_t = \delta + \phi_1 Y_{t-1} + \cdots + \boxed{\phi_k} Y_{t-k} + \varepsilon_t, \quad \widehat{\phi}_{kk}.
\end{aligned}$$

- PACF measures the dependence between Y_t and Y_{t-k} after controlling for intermediate lags.
- For an AR(p) model, the PACF should be close to zero for $k > p$.

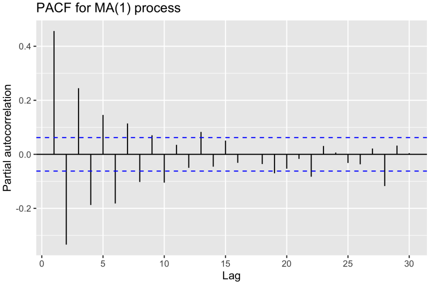
Visual lesson

For AR(p), the PACF cuts off after lag p.

PACFs for AR(1) and MA(1)



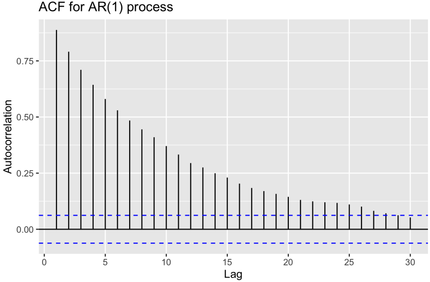
AR(1)



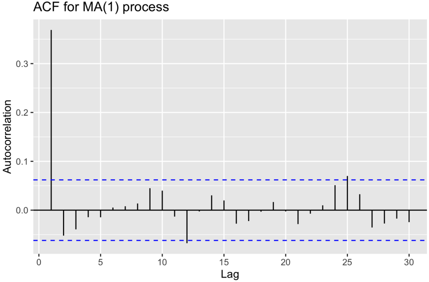
MA(1)

The AR(1) PACF has one dominant spike and then becomes insignificant, while the MA(1) PACF tails off because an MA model corresponds to an infinite-order AR representation.

ACFs for AR(1) and MA(1)



AR(1)



MA(1)

This is the textbook pair to memorize:

- AR(1): ACF tails off, PACF cuts off.
- MA(1): ACF cuts off, PACF tails off.

Using ACF and PACF to determine the order of ARMA models

Model	ACF	PACF
AR(p)	Tails off	Cuts off after lag p
MA(q)	Cuts off after lag q	Tails off
ARMA(p, q)	Tails off	Tails off
Unsuitable model	Cuts off	Cuts off

- The textbook emphasizes that we should read ACF and PACF *together*.
- For ARMA models, both functions usually tail off.
- So ACF/PACF give *candidate* orders, not a final answer.

Why does the sample PACF not literally hit zero?

The textbook explicitly raises the question: if the theoretical PACF of an AR(1) process is zero for $k > 1$, why do sample PACFs not equal zero exactly?

- Sample PACFs are random variables, so even when their expectation is zero, the realized sample values are almost never exactly zero.
- What matters empirically is not literal equality to zero, but whether the bars beyond the cut-off lie inside the approximate significance bands.
- This is why model identification is always heuristic and sample-size dependent.

Practical reading rule

A theoretical cut-off appears in data as "one or two important spikes, then mostly insignificant bars."

Why ARMA order cannot be pinned down from ACF/PACF alone

The textbook's answer is direct:

- An AR model can be written as an infinite-order MA model.
- An MA model can be written as an infinite-order AR model.
- Therefore, in a genuine ARMA process, both ACF and PACF generally tail off.

Implication

Once both functions tail off, order selection must be completed with estimation, information criteria, and residual diagnostics rather than with pictures alone.

A practical identification workflow

1. Start from a plot of the series and ask whether stationarity is plausible.
2. Inspect the sample ACF and PACF.
3. Use cut-off versus tail-off to propose a small set of candidate AR, MA, or ARMA models.
4. Estimate those candidates.
5. Compare AIC/BIC.
6. Check whether residuals look like white noise.
7. Only then keep the model for forecasting.

This is the lecture's main arc

Identification $\rightarrow$ estimation $\rightarrow$ diagnostics $\rightarrow$ forecasting.

Lecture 3 map

Estimation of the ARMA model: two textbook routes

This section of the textbook focuses on two estimation routes:

1. **Yule–Walker estimation** for AR models, using the autocovariance equations directly.
2. **Maximum likelihood estimation (MLE)** for ARMA models, using the joint density of the implied innovation sequence.

Why two routes?

Yule–Walker is algebraically transparent for AR models; MLE is the general workhorse for ARMA models in practice.

Estimating AR models using Yule–Walker equations

For AR(p), the Yule–Walker equations can be written as

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

- Replace population autocovariances by sample autocovariances or sample ACFs.
- Solve the linear system for $\widehat{\phi}_1, \ldots, \widehat{\phi}_p$.

Limitation

This route is natural for AR models, but not for the full ARMA class.

R block: AR(2) via Yule–Walker

```
set.seed(123)
ts_data <- arima.sim(n = 1000,
                     model = list(ar = c(0.6, -0.4)))

acf_vals <- acf(ts_data, plot = FALSE, lag.max = 2)$acf
gamma0 <- acf_vals[1]
gamma1 <- acf_vals[2]
gamma2 <- acf_vals[3]

mat <- matrix(c(gamma0, gamma1,
                gamma1, gamma0), nrow = 2)
vec <- c(gamma1, gamma2)
phi <- solve(mat, vec)

ar.yw(ts_data, order.max = 2)$ar
```

The textbook's point is that the direct linear-system solution and `ar.yw()` both recover coefficients close to the true AR(2) parameters.

MLE warm-up: what the likelihood is doing

The textbook first reviews likelihood in a simple normal-sample setting before turning to ARMA.

Likelihood idea

Given data $x = (x_1, \ldots, x_n)$ and unknown parameters $\boldsymbol{\theta}$, the likelihood is the joint density viewed as a function of $\boldsymbol{\theta}$:

$$\mathcal{L}(\boldsymbol{\theta} \mid x) = f(x_1, \ldots, x_n \mid \boldsymbol{\theta}).$$

- We maximize the likelihood, or equivalently the log-likelihood.
- Taking logs turns products into sums.
- That is computationally easier and is also the right form for asymptotic theory.

Conditional likelihood for ARMA(p, q)

For

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q},$$

with $\varepsilon_t \sim$ i.i.d. $N(0, \sigma^2)$, the parameter vector is

$$\boldsymbol{\theta} = (c, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma^2)'.$$

Given starting values for past y's and past ε's, we recursively compute

$$\varepsilon_t = y_t - c - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q}.$$

Key step

The unobserved shocks become a computable residual sequence once the parameters are guessed.

Conditional log-likelihood for ARMA(p, q)

Treating the initials as given or approximated, the conditional log-likelihood is

$$\ell(\boldsymbol{\theta}) = -\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma^2) - \sum_{t=1}^{T} \frac{\varepsilon_t^2}{2\sigma^2}.$$

- Numerically maximize $\ell(\boldsymbol{\theta})$ over admissible parameters.
- The textbook notes that initial values are not observed, but their influence vanishes asymptotically when p and q are fixed.
- In practice, functions such as `arima()` or `Arima()` do this optimization for us.

Why MLE is useful

It gives a unified estimation route for AR, MA, and ARMA models.

R block: hand-coded ARMA(1,1) likelihood

```
set.seed(123)
data <- arima.sim(n = 500, model = list(ar = 0.5, ma = 0.4))

loglik_arma11 <- function(par, data) {
  phi    <- par[1]
  theta  <- par[2]
  sigma2 <- par[3]^2
  n   <- length(data)
  eps <- numeric(n)
  for (t in 2:n) {
    eps[t] <- data[t] - phi * data[t - 1] - theta * eps[t - 1]
  }
  sum_ll <- -n / 2 * log(2 * pi * sigma2) - sum(eps^2) / (2 * sigma2)
  -sum_ll
}

optim(c(0.5, 0.5, 1), loglik_arma11, data = data, method = "BFGS")
```

The textbook includes this version mainly for pedagogy: write down the likelihood once, and the structure of ARMA estimation becomes transparent.

R block: practical MLE with built-in functions

```
set.seed(123)
data <- arima.sim(n = 500, model = list(ar = 0.5, ma = 0.4))

fit <- arima(data, order = c(1, 0, 1), method = "ML")
print(fit)
```

The practical message is simple: in applied work, we rarely program the likelihood from scratch unless we need a nonstandard model or want to understand the mechanics.

Model order selection with AIC and BIC

The textbook gives

$$\text{AIC} = -2\ell(\widehat{\boldsymbol{\theta}}) + 2k, \qquad \text{BIC} = -2\ell(\widehat{\boldsymbol{\theta}}) + k \log T,$$

where k is the number of estimated parameters and T is the sample size.

- Smaller AIC or BIC means a better trade-off between fit and complexity.
- Both criteria penalize extra parameters.
- BIC penalizes complexity more strongly than AIC.

Link back to identification

ACF/PACF propose candidates; AIC/BIC help choose among them.

The principle of parsimony

The textbook explicitly ties AIC/BIC to **Occam's Razor**:

- among models that fit reasonably well, prefer the simpler one;
- avoid adding lags unless they provide a meaningful improvement;
- simpler models are easier to interpret, estimate, and forecast.

ARMA lesson

A low-order model that captures the main dependence and leaves white-noise residuals is usually better than a high-order model that memorizes noise.

R block: automatic order selection

```
set.seed(123)
data <- arima.sim(n = 500, model = list(ar = 0.5, ma = 0.4))

library(forecast)
fit_aic <- auto.arima(data, ic = "aic",
                      stepwise = FALSE, approximation = FALSE)
fit_bic <- auto.arima(data, ic = "bic",
                      stepwise = FALSE, approximation = FALSE)

print(fit_aic)
print(fit_bic)
```

In the textbook simulation, both AIC and BIC correctly recover an ARMA(1,1) specification.

Lecture 3 map

After estimation, what do we still need to check?

An estimated model is not automatically an adequate model.

1. Are the fitted coefficients economically and statistically sensible?
2. Do the residuals look like white noise?
3. Are there remaining serial correlations in the residuals?
4. Does the model produce stable, interpretable forecasts?

Box–Jenkins spirit

Fit the model, check the residuals, refine the specification, and only then use it for forecasting.

Residual diagnostics: what “good residuals” look like

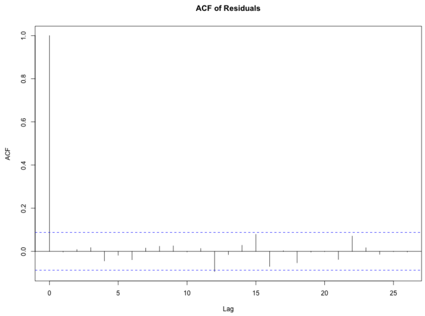
For a well-specified ARMA model, residuals should behave approximately like white noise.

- Residual ACF and PACF should show no systematic remaining structure.
- Most bars should lie inside the approximate significance bands.
- Portmanteau tests such as the Ljung–Box test should not reject residual whiteness too strongly.

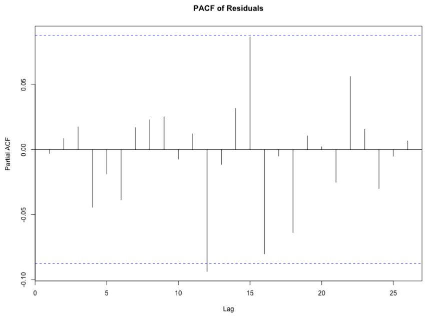
Interpretation

If residual autocorrelation remains, the model has not absorbed all the serial dependence in the data.

Residual ACF and PACF



Residual ACF



Residual PACF

This is the visual target after fitting: no obvious cut-off, no sustained tail-off, and no cluster of significant spikes.

R block: residual diagnostics workflow

```
library(forecast)

fit <- Arima(data, order = c(1, 0, 1))
res <- residuals(fit)

Acf(res)
Pacf(res)
Box.test(res, lag = 20, type = "Ljung-Box", fitdf = 2)
```

The logic is straightforward: once the candidate order is chosen, residual plots and a Ljung–Box check tell us whether the model has left behind serial dependence.

Unconditional mean of ARMA(p, q)

Before forecasting, the textbook computes the unconditional mean. For

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q},$$

stationarity implies

$$\mu = \mathrm{E}(Y_t) = \frac{c}{1 - \phi_1 - \phi_2 - \cdots - \phi_p}.$$

Rewriting around the mean gives

$$(1 - \phi_1 L - \cdots - \phi_p L^p)(Y_t - \mu) = (1 + \theta_1 L + \cdots + \theta_q L^q)\varepsilon_t.$$

Why this matters

Forecasting formulas are simplest when written in deviations from the unconditional mean.

One-step-ahead forecasting

Let $\widehat{Y}_{t+1|t}$ be the forecast of Y_{t+1} using information up to time t. The textbook gives

$$\begin{aligned}\widehat{Y}_{t+1|t} - \mu = \phi_1(Y_t - \mu) + \phi_2(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p+1} - \mu) \\ + \theta_1\widehat{\varepsilon}_t + \theta_2\widehat{\varepsilon}_{t-1} + \cdots + \theta_q\widehat{\varepsilon}_{t-q+1},\end{aligned}$$

where

$$\widehat{\varepsilon}_t = Y_t - \widehat{Y}_{t|t-1}.$$

- The AR part uses past observed values.
- The MA part uses past forecast errors because the true innovations are unobserved.

s-step-ahead forecasts and long-horizon behavior

For horizons $s = 1, 2, \ldots, q$, the textbook writes

$$\begin{aligned}\widehat{Y}_{t+s|t} - \mu = \phi_1(\widehat{Y}_{t+s-1|t} - \mu) + \cdots + \phi_p(\widehat{Y}_{t+s-p|t} - \mu) \\ + \theta_s\widehat{\varepsilon}_t + \theta_{s+1}\widehat{\varepsilon}_{t-1} + \cdots + \theta_q\widehat{\varepsilon}_{t+s-q}.\end{aligned}$$

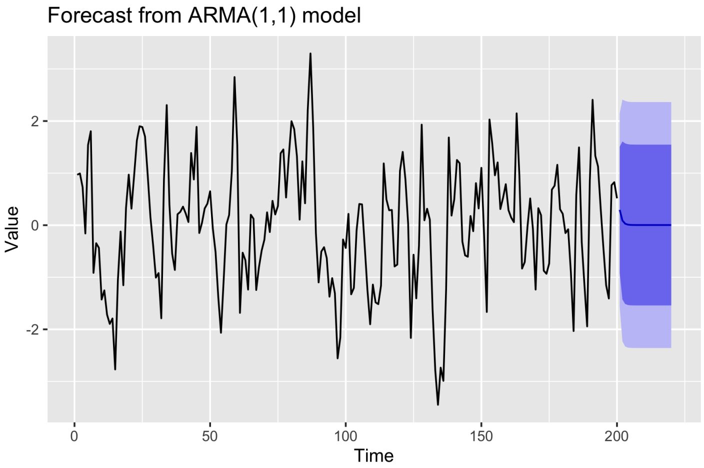
For $s > q$, the MA terms disappear and the forecast obeys the pure AR recursion

$$\widehat{Y}_{t+s|t} - \mu = \phi_1(\widehat{Y}_{t+s-1|t} - \mu) + \cdots + \phi_p(\widehat{Y}_{t+s-p|t} - \mu).$$

Forecast intuition

At longer horizons, the MA component fades away and the AR structure dominates forecast dynamics.

Forecast example from a fitted ARMA(1,1)



The textbook's forecasting example shows the familiar pattern: short-horizon uncertainty widens quickly, and the point forecast moves toward the stationary mean as the horizon increases.

R block: fit and forecast

```
library(forecast)

set.seed(123)
simulated_data <- arima.sim(list(ar = 0.5, ma = 0.3), n = 200)
fit <- Arima(simulated_data, order = c(1, 0, 1))
forecast_result <- forecast(fit, h = 20)
autoplot(forecast_result)
```

This is the full stationary ARMA workflow in compact form: simulate or observe data, estimate the model, diagnose residuals, then forecast.

A stationary-ARMA mindset before we move on

Everything so far relies on the stationary ARMA paradigm.

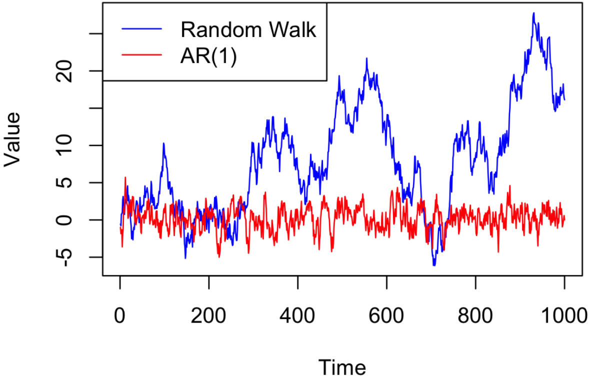
- The unconditional mean exists and is time-invariant.
- The ACF of a stationary AR model tails off.
- Forecasts eventually revert toward a fixed mean or deterministic level implied by the stationary model.
- Shocks are ultimately transitory under stationarity.

Bridge to Chapter 8

What happens when the mean drifts over time, the variance grows over time, or shocks have permanent effects? Then the stationary ARMA language is no longer enough.

Lecture 3 map

Why stationary ARMA is not enough



Deterministic trend versus stochastic trend

The textbook opens Chapter 8 by distinguishing two main sources of nonstationarity:

- **Deterministic trend:** the conditional mean varies over time in a deterministic way, for example a linear or polynomial trend.
- **Stochastic trend (unit root):** the conditional mean evolves stochastically over time, typically because of a unit root.

Why the distinction matters

The two cases behave very differently under shocks, detrending, forecasting, and statistical inference.

Polynomial trend model and trend stationarity

The textbook starts with the additive polynomial trend model

$$y_t = \beta_0 + \beta_1 t + \cdots + \beta_p t^p + u_t = Q_p(t;\beta) + u_t,$$

where $\{u_t\}$ is stationary with mean zero.

- The deterministic component $Q_p(t;\beta)$ is sometimes called a strong or global trend.
- The disturbances u_t are **transitory**: shocks move the series away from trend temporarily, but the process returns toward the trend path.
- This is why such a series is often called **trend stationary**.

Mean, variance, and the source of nonstationarity

Under

$$y_t = Q_p(t;\beta) + u_t, \qquad \mathrm{E}(u_t) = 0,$$

we have

$$\mathrm{E}(y_t) = \beta_0 + \beta_1 t + \cdots + \beta_p t^p, \qquad \mathrm{Var}(y_t) = \sigma_u^2.$$

- Nonstationarity enters through the **mean**, not through an exploding variance.
- After removing the deterministic trend, the remaining component is stationary.

Econometric meaning

For trend-stationary series, detrending is the natural transformation; differencing is not always necessary.

Unmodelled deterministic trend can create spurious persistence

A very important Chapter 8 message is that even if u_t is i.i.d., ignoring the time-varying mean can make the sample ACF look highly persistent. For the simple linear-trend case

$y_t = t + u_t$,

$$\bar{y} = \frac{1}{T}\sum_{t=1}^{T} t = \frac{T+1}{2},$$

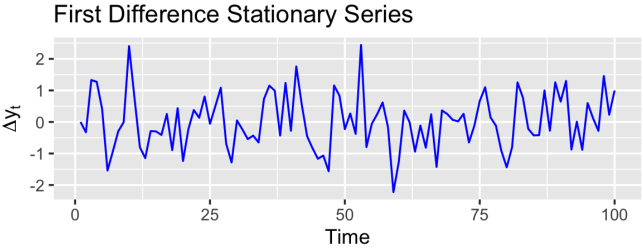
and the sample autocovariance at lag k satisfies

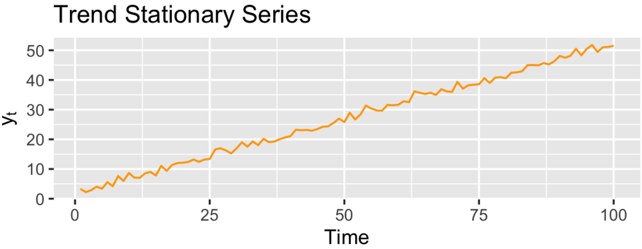
$$\widehat{\gamma}(k) = \frac{1}{T}\sum_{t=1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y}) \longrightarrow \int_0^1 (u - 0.5)^2\, du > 0.$$

Connection back to ACF/PACF

If you forget to remove a deterministic trend, ACF-based identification can falsely suggest strong persistence.

Trend-stationary versus difference-stationary language





How to read the two labels

- A **trend-stationary** series becomes stationary after removing a deterministic trend.
- A **difference-stationary** series becomes stationary after differencing because the level contains a stochastic trend.
- This is the exact bridge from Chapter 2's stationary ARMA language to Chapter 8's unit-root language.

Unit root and random walk as stochastic trend

The textbook then turns to the unit-root process. For the AR(1) model

$$Y_t = \theta Y_{t-1} + \varepsilon_t,$$

a unit root occurs when $\theta = 1$. Then

$$Y_t = Y_{t-1} + \varepsilon_t,$$

which is the **random walk**.

- The unconditional variance does not stay bounded.
- Random shocks have lasting effects.
- The process is nonstationary because the usual weak-stationarity conditions fail.

Random walk with drift: both deterministic and stochastic trend

The textbook's key example is

$$Y_t = \delta + Y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim i.i.d.(0, \sigma^2).$$

Iterating forward gives

$$Y_t = Y_0 + \delta t + \sum_{i=1}^{t} \varepsilon_i.$$

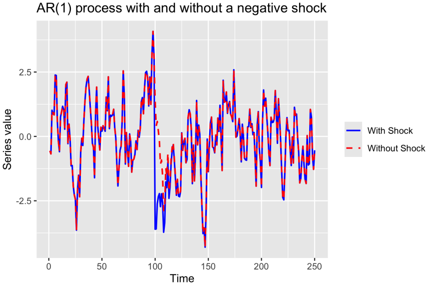
Hence

$$\mathrm{E}(Y_t) = Y_0 + \delta t, \qquad \mathrm{Var}(Y_t) = \sigma^2 t.$$

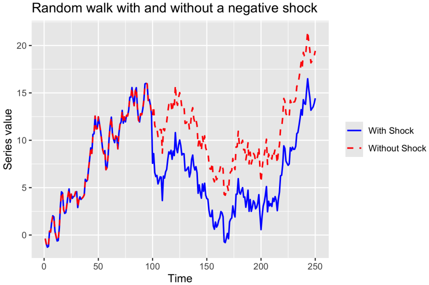
Interpretation

This process contains a deterministic drift in the mean *and* a stochastic trend because the variance grows with time.

Permanent versus transitory shocks



AR(1): shock effect fades



Random walk: shock effect persists

This is the most intuitive distinction in the transition to Chapter 8:

- stationary AR models absorb shocks and return toward their long-run path;
- random walks continue from the new level after the shock.

Deterministic trend versus stochastic trend: compact comparison

	Deterministic trend	Stochastic trend
Prototype	$y_t = \delta + \gamma t + u_t$	$y_t = y_{t-1} + \varepsilon_t$
Source of nonstationarity	Mean changes with time	Level accumulates shocks
Variance	Typically bounded after detrending	Grows with time
Shock effect	Transitory	Permanent
Natural transformation	Detrend	Difference
Forecast behavior	Reverts to trend path	Follows current level plus drift

Why this transition matters for model identification

Everything in the first half of today's lecture should now be read with one warning label:

- ACF/PACF heuristics are designed for stationary analysis.
- A deterministic trend can fake persistence if it is not removed first.
- A unit root can generate very slow decay and permanent shock effects that do not fit the stationary ARMA logic.

So the next step is unavoidable

Before choosing between AR, MA, and ARMA models, we must learn how to tell trend-stationary behavior from unit-root behavior.

Preview of where Chapter 8 goes next

After the transition slides in this lecture, the textbook continues with:

- unit-root processes and ARIMA language,
- Dickey–Fuller and augmented Dickey–Fuller tests,
- KPSS as the complementary stationarity test,
- empirical applications to macro and financial time series.

Course implication

Lecture 4 can now start from the question we created today: is apparent persistence coming from a stationary ARMA structure, a deterministic trend, or a stochastic trend?

Lecture 3 map

Lecture 3 takeaways

1. **ACF and PACF are identification heuristics:** AR(p) means ACF tails off and PACF cuts off; MA(q) means ACF cuts off and PACF tails off.
2. **ARMA identification is not finished by looking at plots:** use estimation, AIC/BIC, and residual diagnostics.
3. **Forecasting in stationary ARMA models is mean-reverting:** the MA component matters in the short run, while the AR recursion dominates longer horizons.
4. **The transition to nonstationarity changes everything:** deterministic trends require detrending, while stochastic trends imply permanent shocks and often call for differencing and unit-root testing.

Bottom line

Today's lecture closes Chapter 2's stationary ARMA workflow and opens Chapter 8's nonstationary perspective.