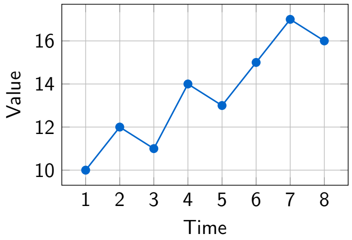
What is a Time Series?

Definition

A **time series** is a sequence of data points collected over time, typically at regular intervals.

Examples:

- Daily temperature readings
- Monthly sales figures
- Yearly population counts
- Hourly stock prices
- Weekly COVID-19 cases



Why Study Time Series?

Main Goal: Forecasting the Future!

Business

- Sales prediction
- Inventory planning
- Budget allocation

Science

- Weather forecasting
- Climate modeling
- Disease outbreaks

Finance

- Stock prices
- Interest rates
- Risk management

Key Insight

Understanding the **past patterns** helps us predict **future values**!

Time Series Notation

Basic Notation

We denote a time series as: $\{Y_t\}$ where $t = 1, 2, 3, \ldots, n$

Example: Ice Cream Sales (units)

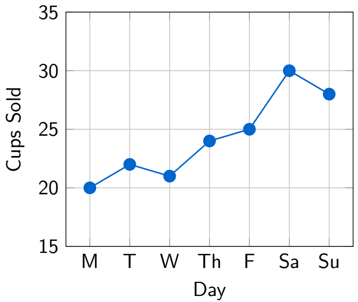
Month (t)	1	2	3	4	5	6	7
Sales (Y_t)	100	110	125	140	155	165	180

- $Y_1 = 100$ (sales in month 1)
- $Y_3 = 125$ (sales in month 3)
- Y_{t-1} means the **previous** observation (lag 1)
- Y_{t-2} means **two periods ago** (lag 2)

A Simple Example: Lemonade Stand

Scenario: You run a lemonade stand and track daily sales.

Day	Cups Sold
Monday	20
Tuesday	22
Wednesday	21
Thursday	24
Friday	25
Saturday	30
Sunday	28



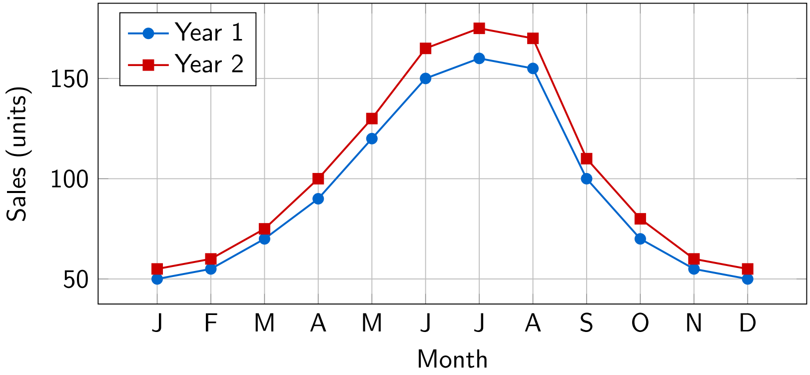
Question: How many cups will you sell next Monday?

Four Components of Time Series

1. **Trend (T)**: Long-term direction
   - Upward, downward, or flat
2. **Seasonality (S)**: Regular patterns
   - Weekly, monthly, yearly cycles
3. **Cyclical (C)**: Irregular fluctuations
   - Economic cycles
4. **Random/Noise (ϵ)**: Unpredictable
   - What we cannot explain

Visual Examples:
Trend: Straight line up or down
Seasonality: Wave pattern
Noise: Random jumps
Combined: Real-world data!

Real-World Example: Ice Cream Sales



- **Trend**: Sales are gradually increasing (Year 2 ¿ Year 1)
- **Seasonality**: Peak in summer (June-August), low in winter

What is Stationarity?

Definition (Simple)

A time series is **stationary** if its statistical properties (mean, variance) don't change over time.

Stationary Series:

- Constant mean
- Constant variance
- No trend
- ✓ Good for ARIMA

Non-Stationary Series:

- Changing mean (trend)
- May have changing variance
- Upward/downward drift
- × Needs transformation

Why Does Stationarity Matter?

Important!

ARIMA models **require stationarity** to work properly!

Think of it this way:

- If a series is stationary, past patterns will **repeat** in the future
- If non-stationary, the patterns keep **changing**, making prediction harder

Analogy

Imagine predicting the weather:

- **Stationary**: Temperature in a stable climate - patterns repeat yearly
- **Non-stationary**: Temperature with global warming - getting hotter each year

Making Data Stationary: Differencing

Differencing

Subtract consecutive observations to remove trends.

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

Example:

t	1	2	3	4	5	6
Y_t	10	12	15	19	24	30
$Y_t - Y_{t-1}$	–	2	3	4	5	6

Original (non-stationary):
Shows upward trend

After differencing:
Still not constant, but closer!
(May need 2nd differencing)

What is an AR Model?

Key Idea

Today's value depends on **yesterday's value** (and maybe days before).

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

AR(1) Model Components:

- Y_t = today's value
- Y_{t-1} = yesterday's value (lag 1)
- ϕ_1 = how much yesterday influences today (coefficient)
- c = constant (baseline level)
- ϵ_t = random error (unpredictable part)

Think of it like...

Tomorrow's weather is similar to today's weather, plus some randomness!

AR(1) Example: Daily Temperature

Model: $Y_t = 5 + 0.8Y_{t-1} + \epsilon_t$

Meaning:

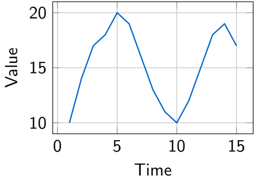
- Baseline: 5°C
- 80% of yesterday's temperature carries over
- Plus some random variation

Calculation Example:

- If yesterday's temperature was $Y_{t-1} = 20C$
- Expected today: $Y_t = 5 + 0.8 \times 20 = 5 + 16 = 21C$
- Actual might be $21 \pm \epsilon$ (e.g., 19°C or 23°C)

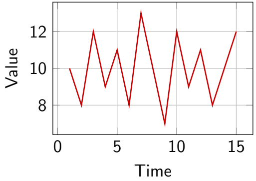
AR(1): Visual Understanding

$\phi_1 = 0.9$ **(high persistence)**



- Slow, smooth changes
- High correlation between neighbors

$\phi_1 = 0.3$ **(low persistence)**



- Quick, jumpy changes
- Low correlation between neighbors

Interpretation

Higher ϕ_1 = stronger memory = smoother series

AR(2) Model: Looking Back Two Steps

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

Today depends on:

- Yesterday (Y_{t-1}) with weight ϕ_1
- Two days ago (Y_{t-2}) with weight ϕ_2

Example: $Y_t = 10 + 0.5Y_{t-1} + 0.3Y_{t-2} + \epsilon_t$

If $Y_{t-1} = 20$ and $Y_{t-2} = 18$:

$$\begin{aligned} Y_t &= 10 + 0.5(20) + 0.3(18) \\ &= 10 + 10 + 5.4 \\ &= 25.4 \end{aligned}$$

General AR(p) Model

AR(p) Formula

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

- p = **order** of the AR model
- p = how many past values we use

Model	Uses	Example
AR(1)	1 past value	Stock prices
AR(2)	2 past values	Quarterly GDP
AR(3)	3 past values	Some climate data
AR(p)	p past values	Depends on data

AR Practice Problem

Exercise

Given the AR(1) model: $Y_t = 2 + 0.7Y_{t-1} + \epsilon_t$

If $Y_{100} = 15$, what is the expected value of Y_{101}?

AR Practice Problem

Exercise

Given the AR(1) model: $Y_t = 2 + 0.7Y_{t-1} + \epsilon_t$

If $Y_{100} = 15$, what is the expected value of Y_{101}?

Solution:

$$
\begin{aligned}
Y_{101} &= 2 + 0.7 \times Y_{100} + \epsilon_{101} \\
E[Y_{101}] &= 2 + 0.7 \times 15 + 0 \quad \text{(expected error is 0)} \\
E[Y_{101}] &= 2 + 10.5 \\
E[Y_{101}] &= \boxed{12.5}
\end{aligned}
$$

AR Practice Problem

Exercise

Given the AR(1) model: $Y_t = 2 + 0.7Y_{t-1} + \epsilon_t$

If $Y_{100} = 15$, what is the expected value of Y_{101}?

Solution:

$$\begin{aligned} Y_{101} &= 2 + 0.7 \times Y_{100} + \epsilon_{101} \\ E[Y_{101}] &= 2 + 0.7 \times 15 + 0 \quad \text{(expected error is 0)} \\ E[Y_{101}] &= 2 + 10.5 \\ E[Y_{101}] &= \boxed{12.5} \end{aligned}$$

Note

The actual value will be $12.5 \pm \epsilon$, where ϵ is random noise.

What is an MA Model?

Key Idea

Today's value depends on **past random shocks** (errors), not past values!

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

MA(1) Model Components:

- Y_t = today's value
- μ = mean (average level)
- ϵ_t = today's random shock
- ϵ_{t-1} = yesterday's random shock
- θ_1 = how much yesterday's shock affects today

Think of it like...

An unexpected event yesterday still affects you today!

MA vs AR: What's the Difference?

AR: AutoRegressive

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

Depends on **past values**

Example: If it was hot yesterday, it's probably hot today.

MA: Moving Average

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

Depends on **past shocks**

Example: If there was a surprise storm yesterday, today might still be affected.

Key Difference

- **AR**: "History of the series itself"
- **MA**: "History of the surprises/shocks"

MA(1) Example: Unexpected Events

Model: $Y_t = 100 + \epsilon_t + 0.6\epsilon_{t-1}$

Scenario: Daily sales with average of 100 units

Suppose:

- Yesterday there was a surprise promotion: $\epsilon_{t-1} = +20$ (sold 20 extra)
- Today's random event: $\epsilon_t = +5$

Today's sales:

$$\begin{aligned} Y_t &= 100 + 5 + 0.6 \times 20 \\ &= 100 + 5 + 12 \\ &= 117 \text{ units} \end{aligned}$$

Interpretation

The surprise from yesterday (+20) still affects today (+12), but with a smaller impact (60%).

MA(q) General Model

MA(q) Formula

$$Y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$$

- q = **order** of the MA model
- q = how many past shocks we consider

Model	Description
MA(1)	Today affected by shock from 1 period ago
MA(2)	Today affected by shocks from 1 and 2 periods ago
MA(q)	Today affected by shocks from 1, 2, ..., q periods ago

Key Property

MA(q) has "short memory" - effects die out after q periods!

ARMA Model: Best of Both Worlds

ARMA(p,q) Model

$$Y_t = c + \underbrace{\phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p}}_{\text{AR part}} + \epsilon_t + \underbrace{\theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}}_{\text{MA part}}$$

ARMA combines:

- Past values (AR)
- Past shocks (MA)

Notation:

- p = AR order
- q = MA order

Special Cases

- ARMA(1,0) = AR(1)
- ARMA(0,1) = MA(1)
- ARMA(1,1) = AR(1) + MA(1)

ARMA(1,1) Example

Model: $Y_t = 5 + 0.7Y_{t-1} + \epsilon_t + 0.4\epsilon_{t-1}$

Given:

- $Y_{t-1} = 20$ (yesterday's value)
- $\epsilon_{t-1} = 3$ (yesterday's shock)
- $\epsilon_t = -1$ (today's shock)

Calculate Y_t:

$$\begin{aligned} Y_t &= 5 + 0.7(20) + (-1) + 0.4(3) \\ &= 5 + 14 - 1 + 1.2 \\ &= \boxed{19.2} \end{aligned}$$

Breakdown

Constant (5) + AR effect (14) + Today's shock (-1) + MA effect (1.2) = 19.2

The "I" in ARIMA: Integration

The Problem

ARMA models only work for **stationary** data!
Real-world data often has trends (non-stationary).

The Solution: Differencing

Transform non-stationary data by taking differences!

First difference: $\nabla Y_t = Y_t - Y_{t-1}$
Second difference: $\nabla^2 Y_t = \nabla Y_t - \nabla Y_{t-1}$

Differencing Example

Original Data (with trend):

t	1	2	3	4	5	6	7	8
Y_t	10	13	17	22	28	35	43	52

First Difference ($d = 1$):

t	2	3	4	5	6	7	8
$Y_t - Y_{t-1}$	3	4	5	6	7	8	9

Second Difference ($d = 2$):

t	3	4	5	6	7	8
$\nabla^2 Y_t$	1	1	1	1	1	1

✓ After 2nd differencing: constant! (Stationary)

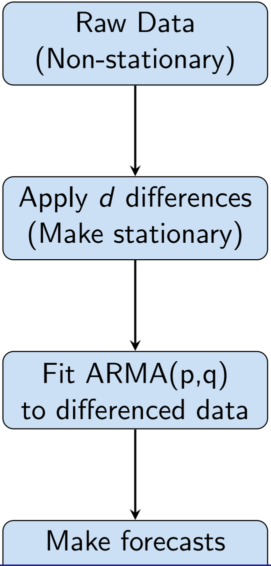
ARIMA(p,d,q) Model

ARIMA Notation

$$\text{ARIMA}(\underbrace{p}_{\text{AR}}, \underbrace{d}_{\text{diff}}, \underbrace{q}_{\text{MA}})$$

Parameter	Meaning	Typical Values
p	Number of AR terms (past values)	0, 1, 2, 3
d	Number of differences (to make stationary)	0, 1, 2
q	Number of MA terms (past shocks)	0, 1, 2, 3

ARIMA Process: Step by Step



ARIMA Examples: Model Interpretation

ARIMA(1,1,0):

- Difference once ($d = 1$)
- Then fit AR(1) to differenced data
- Formula: $\nabla Y_t = c + \phi_1 \nabla Y_{t-1} + \epsilon_t$

ARIMA(0,1,1):

- Difference once ($d = 1$)
- Then fit MA(1) to differenced data
- Formula: $\nabla Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1}$
- This is called **Exponential Smoothing**!

ARIMA(1,1,1):

- Difference once ($d = 1$)
- Then fit ARMA(1,1) to differenced data
- Formula: $\nabla Y_t = c + \phi_1 \nabla Y_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1}$

Autocorrelation: Correlation with Yourself

Definition

Autocorrelation measures how correlated a time series is with its own past values.

Autocorrelation at lag k: How related is Y_t to Y_{t-k}?

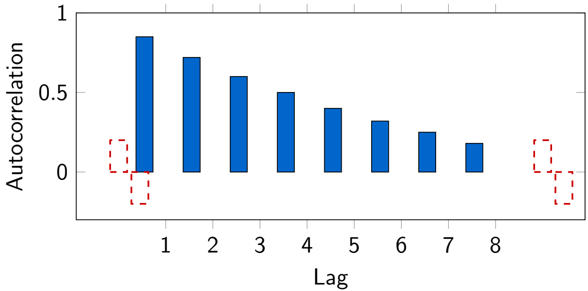
Lag	**Correlation between...**
1	Today and Yesterday
2	Today and 2 days ago
7	Today and 1 week ago
12	Today and 1 year ago (monthly data)

Notation: ρ_k or r_k for autocorrelation at lag k

ACF: Autocorrelation Function

ACF

Plot of autocorrelations at different lags.

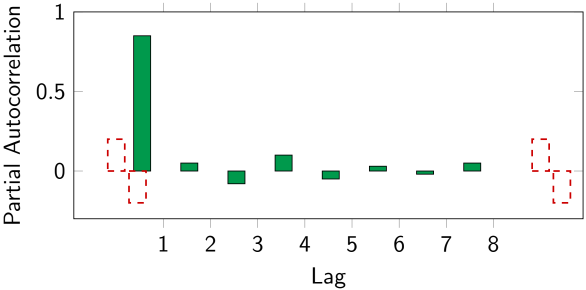


- Red dashed lines: Significance bounds (95%)
- Bars outside bounds are **significant**

PACF: Partial Autocorrelation Function

PACF

Measures correlation at lag k **after removing** effects of lags 1 to $k-1$.



- Only lag 1 is significant $\Rightarrow$ suggests **AR(1)** model!

Using ACF and PACF to Choose p and q

Model	ACF Pattern	PACF Pattern
AR(p)	Decays gradually	Cuts off after lag p
MA(q)	Cuts off after lag q	Decays gradually
ARMA(p,q)	Decays gradually	Decays gradually

Practical Rules

- If **PACF cuts off** at lag p, and ACF decays $\Rightarrow$ use AR(p)
- If **ACF cuts off** at lag q, and PACF decays $\Rightarrow$ use MA(q)
- If **both decay** gradually $\Rightarrow$ use ARMA(p,q)

ACF/PACF Pattern Summary

AR(1) Pattern:

- ACF: Slowly decays (exponentially)
- PACF: Only lag 1 significant, then cuts off

AR(2) Pattern:

- ACF: Slowly decays (may oscillate)
- PACF: Lags 1 and 2 significant, then cuts off

MA(1) Pattern:

- ACF: Only lag 1 significant, then cuts off
- PACF: Slowly decays

MA(2) Pattern:

- ACF: Lags 1 and 2 significant, then cuts off
- PACF: Slowly decays

ACF/PACF Visual Examples

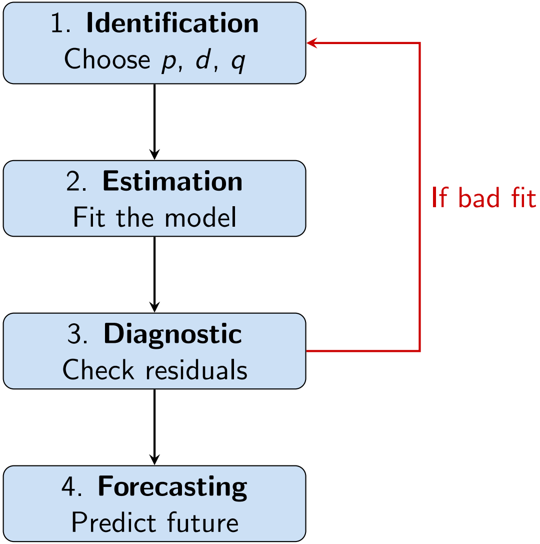
How to read ACF/PACF plots:

What you see	ACF	PACF	Model
ACF decays, PACF spike at 1	Decay	Cut at 1	AR(1)
ACF decays, PACF spikes at 1,2	Decay	Cut at 2	AR(2)
ACF spike at 1, PACF decays	Cut at 1	Decay	MA(1)
ACF spikes at 1,2, PACF decays	Cut at 2	Decay	MA(2)
Both decay	Decay	Decay	ARMA

Important Note

In practice, patterns are rarely this clean! Use this as a starting point, then refine.

The Box-Jenkins Methodology



Step 1: Identification

Goal: Determine values of p, d, and q

1. **Check for stationarity**
   - Plot the data
   - Look for trends or changing variance
   - If non-stationary: difference the data ($d = 1$ or $d = 2$)
2. **Examine ACF and PACF**
   - ACF cuts off at lag $q \Rightarrow$ MA(q)
   - PACF cuts off at lag $p \Rightarrow$ AR(p)
3. **Start simple!**
   - Begin with low values: ARIMA(1,1,1) or ARIMA(1,1,0)
   - Add complexity only if needed

Step 2: Estimation

Goal: Estimate model parameters (ϕ, θ, etc.)

Methods

- **Maximum Likelihood Estimation (MLE)** - most common
- **Least Squares** - simpler alternative

In practice: Software does this automatically!

Software Options

- **R:** `arima()`, `auto.arima()`
- **Python:** `statsmodels.ARIMA`
- **Excel:** Various add-ins

Step 3: Diagnostic Checking

Goal: Verify the model fits well

What to Check

Residuals should behave like **white noise**:

- Mean $= 0$
- Constant variance
- No autocorrelation (ACF of residuals all near 0)
- Approximately normal distribution

Step 4: Forecasting

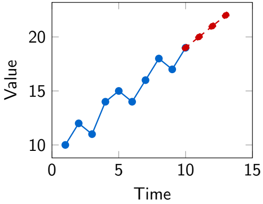
Goal: Predict future values

Point Forecast:

- Single best prediction
- Example: "Sales will be 150 units"

Interval Forecast:

- Range of likely values
- Example: "95% CI: 130 to 170 units"



Model Selection Criteria

How to choose between different ARIMA models?

AIC (Akaike Information Criterion)

$$\mathrm{AIC} = -2\log(L) + 2k$$

- L = likelihood (how well model fits)
- k = number of parameters
- **Lower AIC = Better model**

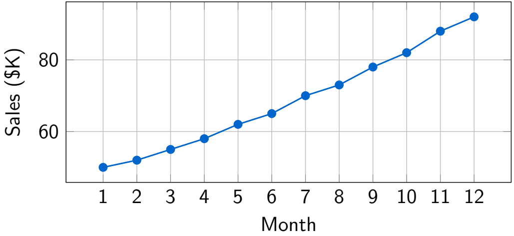
BIC (Bayesian Information Criterion)

$$\mathrm{BIC} = -2\log(L) + k\log(n)$$

Example: Monthly Sales Data

Scenario: A store tracks monthly sales (in thousands of dollars)

Month	1	2	3	4	5	6	7	8	9	10	11	12
Sales	50	52	55	58	62	65	70	73	78	82	88	92

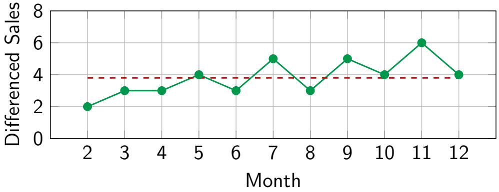


Observation: Clear upward **trend** $\Rightarrow$ non-stationary!

Example: Step 1 - Make Stationary

Apply first differencing $(d = 1)$:

Month	2	3	4	5	6	7	8	9	10	11	12
∇Y_t	2	3	3	4	3	5	3	5	4	6	4



✓ Now looks more stationary (fluctuates around mean ≈ 3.8)

Example: Step 2 - Analyze ACF/PACF

Assume ACF and PACF analysis suggests:

- PACF: Significant at lag 1, then cuts off
- ACF: Gradual decay

$\Rightarrow$ This suggests an **AR(1)** model on differenced data
$\Rightarrow$ Overall model: **ARIMA(1,1,0)**

Model Selection

$$\text{ARIMA}(1, 1, 0)$$

$p = 1$ (one AR term), $d = 1$ (one difference), $q = 0$ (no MA terms)

Example: Step 3 - Estimate Parameters

Fitting ARIMA(1,1,0):

Model equation: $\nabla Y_t = c + \phi_1 \nabla Y_{t-1} + \epsilon_t$

Suppose estimation gives:

- $c = 2.5$
- $\phi_1 = 0.3$

Fitted model:

$$\nabla Y_t = 2.5 + 0.3 \nabla Y_{t-1} + \epsilon_t$$

Or equivalently:

$$Y_t - Y_{t-1} = 2.5 + 0.3(Y_{t-1} - Y_{t-2}) + \epsilon_t$$

Example: Step 4 - Forecast

Forecast for Month 13:

Given:

- $Y_{12} = 92$
- $Y_{11} = 88$
- $\nabla Y_{12} = Y_{12} - Y_{11} = 4$

Calculate:

$$\begin{aligned}\nabla Y_{13} &= 2.5 + 0.3 \times \nabla Y_{12} \\ &= 2.5 + 0.3 \times 4 \\ &= 2.5 + 1.2 = 3.7\end{aligned}$$

Convert back:

$$Y_{13} = Y_{12} + \nabla Y_{13} = 92 + 3.7 = \boxed{95.7}$$

Forecast: Sales in month 13 $\approx$ \$95,700

Example: Forecast Multiple Periods

Continuing the forecast...

Month 14:

$$\nabla Y_{14} = 2.5 + 0.3 \times 3.7 = 2.5 + 1.11 = 3.61$$
$$Y_{14} = 95.7 + 3.61 = 99.31$$

Month 15:

$$\nabla Y_{15} = 2.5 + 0.3 \times 3.61 = 2.5 + 1.08 = 3.58$$
$$Y_{15} = 99.31 + 3.58 = 102.89$$

Month	13	14	15
Forecast	95.7	99.3	102.9

Tips for Beginners

Start Simple

- Try ARIMA(1,1,0), ARIMA(0,1,1), or ARIMA(1,1,1) first
- Only add complexity if diagnostics suggest it

Use Automatic Selection

- R: `auto.arima()` function
- Python: `pmdarima.auto_arima()`
- Let software suggest initial values

Always Visualize

- Plot your data first
- Plot ACF/PACF
- Plot residuals after fitting

Software: R Example

R Code for ARIMA

```
# Load data
sales <- c(50, 52, 55, 58, 62, 65, 70, 73, 78, 82, 88, 92)
ts_sales <- ts(sales, frequency=12)

# Automatic ARIMA selection
library(forecast)
model <- auto.arima(ts_sales)
summary(model)

# Or manual specification
model <- arima(ts_sales, order=c(1,1,0))

# Forecast next 3 periods
forecast(model, h=3)

# Plot forecast
plot(forecast(model, h=3))
```

Software: Python Example

Python Code for ARIMA

```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt

# Create data
sales = [50, 52, 55, 58, 62, 65, 70, 73, 78, 82, 88, 92]

# Fit ARIMA(1,1,0)
model = ARIMA(sales, order=(1,1,0))
fitted = model.fit()
print(fitted.summary())

# Forecast
forecast = fitted.forecast(steps=3)
print(forecast)
```

Interpreting Software Output

Key things to look for:

Output	**What to check**
Coefficients	Are they significant? (p-value ¡ 0.05)
AIC/BIC	Lower is better (compare models)
Residual diagnostics	Should be white noise
Ljung-Box test	p-value ¿ 0.05 (no autocorrelation)

What's Next? Beyond ARIMA

Extensions of ARIMA:

- **SARIMA**: Seasonal ARIMA for data with seasonal patterns
  - SARIMA(p,d,q)(P,D,Q)$_m$
  - Example: Monthly data with yearly seasonality
- **ARIMAX**: ARIMA with external variables (predictors)
- **GARCH**: For modeling changing variance
- **VAR**: Vector AR for multiple related time series
- **Machine Learning**: LSTM, Prophet, etc.