Time Series Forecasting with ARIMA in Healthcare

Application: Forecasting Daily Hospital Patient Volumes

Table of Contents

1. Learning Objectives
2. Introduction
3. Target Audience and Background Assumptions
4. Healthcare Industry Context
5. Theoretical Foundations
    - 5.1 What Is a Time Series?
    - 5.2 Key Components: Trend, Seasonality, Noise
    - 5.3 Stationarity
    - 5.4 Autocorrelation and ACF/PACF
    - 5.5 ARIMA Model: Mathematical Foundations
    - 5.6 Why ARIMA Is Suitable for Healthcare
6. Dataset Description and Pre-Processing
7. Step-by-Step Implementation Tutorial
    - 7.1 Environment Setup
    - 7.2 Loading and Exploring Data
    - 7.3 Visual Diagnostics
    - 7.4 Stationarity Testing
    - 7.5 Differencing to Achieve Stationarity
    - 7.6 ACF & PACF Interpretation
    - 7.7 Model Building and Parameter Selection
    - 7.8 Model Training (ARIMA)
    - 7.9 Generating Forecasts

    - 7.10 Model Evaluation
    - 7.11 Residual Diagnostics
8. Comprehensive Concept Diagrams
9. Exercises (with fully explained solutions)
10. Summary and Key Takeaways

Learning Objectives

After completing this tutorial, learners will be able to:

Conceptual Understanding

- Explain the structure and assumptions behind ARIMA models.
- Differentiate between AR, I, and MA components with real-world analogies.
- Describe stationarity and its importance in forecasting.

Technical Skills

- Conduct exploratory time series analysis for operational healthcare data.
- Apply statistical stationarity tests (ADF test).
- Identify ARIMA parameters using ACF and PACF diagnostics.
- Fit, validate, and tune ARIMA models in Python.

Applied Healthcare Competencies

- Translate time series forecasts into hospital decision-making.
- Evaluate how forecasting accuracy affects staffing and resource allocation.

Pedagogical Objectives

- Develop intuition for interpreting autocorrelation structures.
- Build confidence in debugging modeling issues.
- Practice critical thinking through applied exercises with solutions.

Introduction

Hospitals operate in environments where uncertainty is costly. Staffing shortages, ED overcrowding, and ICU bottlenecks result in longer wait times, reduced care quality, and financial losses. To mitigate these issues, healthcare organizations increasingly rely on predictive analytics, and one of the most widely used techniques is time series forecasting.

This tutorial teaches ARIMA forecasting through the lens of healthcare operations, specifically:

Forecasting daily Emergency Department (ED) patient volumes.

This problem is operationally important because:

- EDs experience strong day-of-week seasonality.
- Visit volumes directly determine staffing requirements.
- Predicting surges can reduce wait times and improve patient outcomes.

This tutorial is designed not merely to demonstrate ARIMA but to teach ARIMA, following the educational philosophy of INFO 7390: combining conceptual clarity, implementation mastery, and industry relevance.

Target Audience & Prerequisites

Audience

- Graduate students in data science
- Healthcare analysts
- Entry-level data scientists developing forecasting skills

Assumed Background

- Python programming
- Pandas data manipulation
- Basic understanding of regression
- Familiarity with Jupyter Notebooks

No prior experience with ARIMA or advanced time series models is required.

Healthcare Industry Context

Hospitals experience predictable yet highly variable patterns in daily demand:

- Mondays often have 10–20% higher volumes
- Winter spikes (flu season)
- Summer declines
- Daily stochastic fluctuations

Misestimating patient loads leads to:

- Understaffing → burnout, long wait times
- Overstaffing → unnecessary costs
- Bed bottlenecks → delayed transfers from ED to inpatient units

Forecasting is therefore integral to:

- Workforce planning
- Supply chain management
- Capacity management
- Emergency preparedness

ARIMA is well-suited because healthcare demand is:

- Continuous
- Time-dependent
- Seasonal
- Strongly autocorrelated

Theoretical Foundations

This section develops a rigorous conceptual understanding of time series forecasting and ARIMA modeling, integrating both mathematical theory and healthcare operational relevance. The goal is to help learners not only implement ARIMA, but reason about why it works, when it fails, and how it applies to hospital patient volumes.

What Is a Time Series?

A time series is a sequence of observations indexed in chronological order. Unlike traditional (independent and identically distributed) data used in regression modeling, time series data exhibit temporal dependency, meaning today's value is often correlated with yesterday's value.

In real healthcare settings, time series appear in numerous operational metrics:

Healthcare Metric	Frequency	Examples of Drivers
ED patient volume	Daily / Hourly	Weekend effects, flu season, weather
ICU census	Hourly	Post-operative cases, critical care demand
Operating room case counts	Daily	Surgical schedules, staffing patterns
Infection incidence (e.g., flu, RSV, COVID)	Weekly	Community spread, seasonal cycles

Why This Matters for Forecasting

Predicting ED volumes requires models that can “remember” temporal patterns—something standard regression or machine learning models may not naturally capture unless engineered explicitly.

Diagram Placeholder — Figure 1: Example Time Series of Daily ED Visit Volumes

A line chart representing daily ED arrivals over two years.

Should clearly show:

- Seasonal weekly oscillations (weekends typically lower volume)
- Winter spikes (influenza/RSV season)

- Long-term drift (population changes or hospital capacity changes)

This visualization grounds learners in how healthcare time series behave.

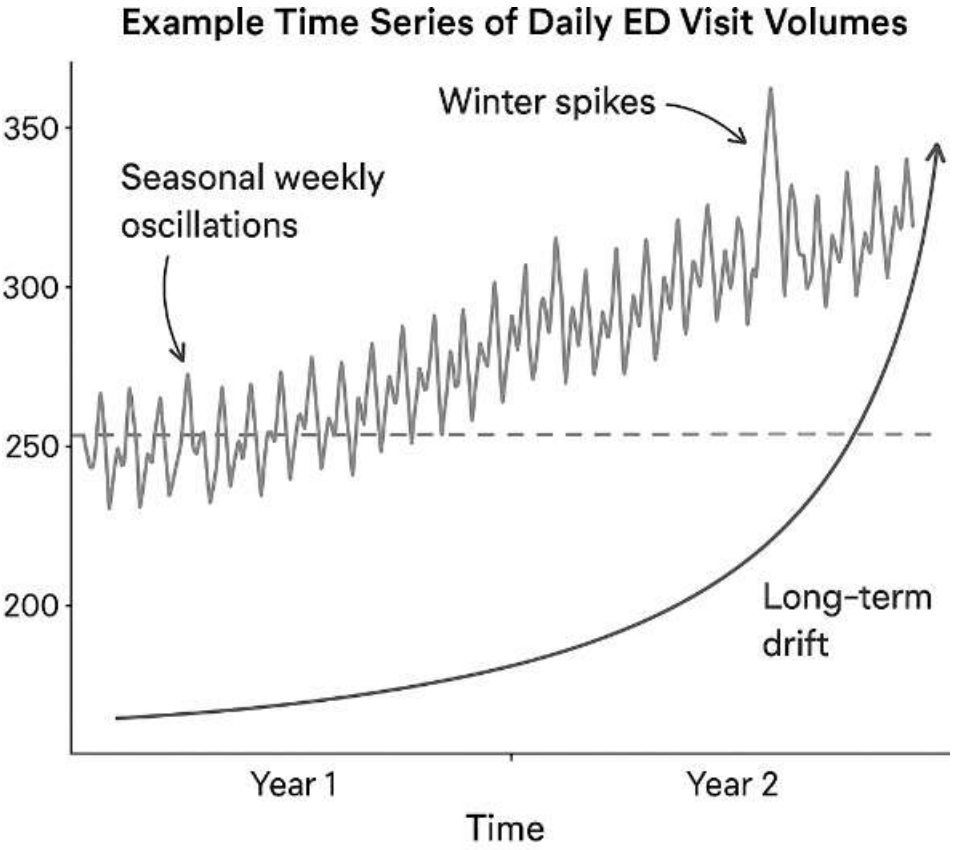


Figure 1: Example Time Series of Daily ED Visit Volumes

Time Series Components

Real-world time series can often be decomposed into conceptual building blocks:

---

1. Trend (T_t)

A long-term directional movement in the data.

In healthcare:

- ED volumes may increase over years due to population growth or hospital consolidation.
- Seasonal infections create upward winter pressure.

Trend is not always linear; it can be exponential, piecewise, or cyclical.

---

2. Seasonality (S_t)

A repeating, predictable pattern that recurs at fixed intervals.

Common Healthcare Seasonalities:

- Weekly seasonality: Mondays have higher visits; weekends typically lower.
- Annual seasonality: Winter respiratory illnesses elevate patient demand.

Mathematically, seasonality repeats with a fixed frequency (e.g., weekly: s = 7).

---

3. Irregular or Random Noise (R_t)

Random fluctuations that cannot be systematically explained.

In healthcare:

- Sudden accidents or localized outbreaks.
- Holiday effects.

Additive vs Multiplicative Time Series Models

Additive Model

$$Y_t = T_t + S_t + R_t$$

Used when seasonal variation is constant over time.

Multiplicative Model

$$Y_t = T_t \cdot S_t \cdot R_t$$

Used when seasonal variation grows proportionally with overall level (e.g., winter peaks becoming more extreme each year).

Diagram Placeholder — Figure 2: Time Series Decomposition

Insert a decomposition plot showing:

- Top: Original ED series
- Middle: Trend and seasonal pattern
- Bottom: Residual noise

This builds intuition for how ARIMA handles different patterns.

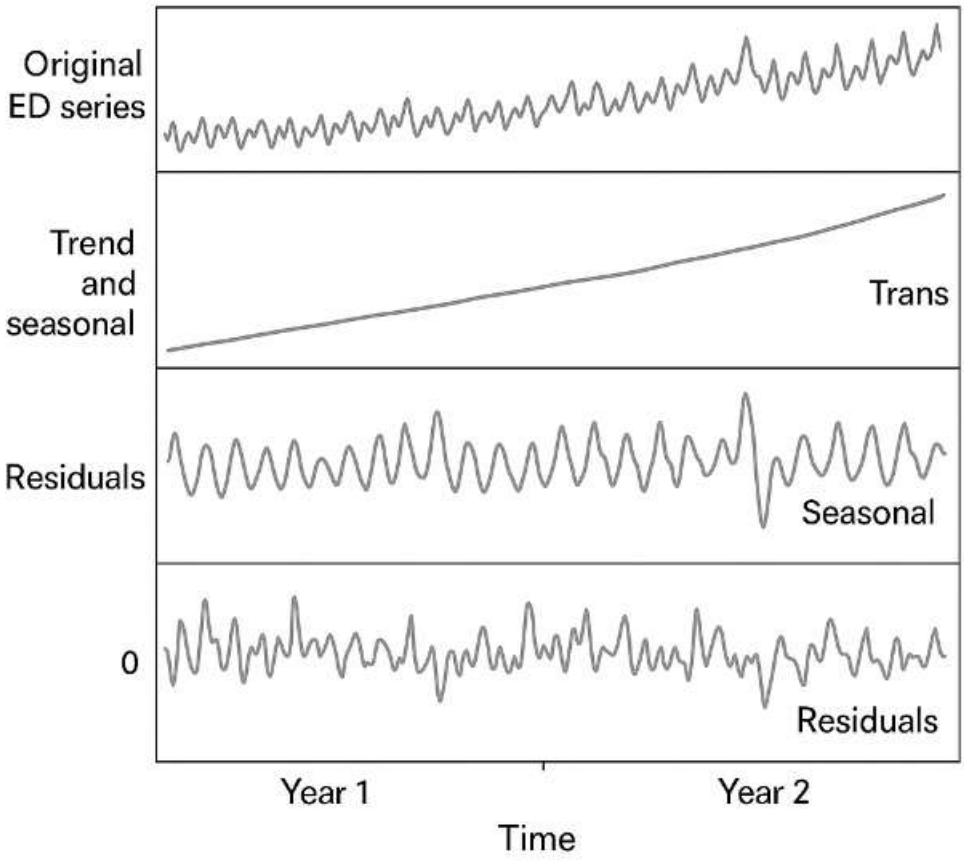


Figure 2: Time Series Decomposition

Stationarity

Definition

A time series is stationary if its statistical properties do not depend on time:

- Mean: constant
- Variance: constant
- Autocorrelation: depends only on lag, not time itself

Formally, for a weakly stationary (second-order) series:

$$E[Y_t] = \mu, Var(Y_t) = \sigma^2, Cov(Y_t, Y_{t-k}) = \gamma(k)$$

Why ARIMA Requires Stationarity

ARIMA models the relationship between lagged values:

- If the series drifts upward, then the model's coefficients become unstable.
- Residuals will show autocorrelation.
- Forecasts may explode or systematically over/underestimate the true values.

Stationarity ensures that past relationships reliably describe future patterns.

How to Check Stationarity

1. Visual inspection

Look for:

- Trend
- Changing variance
- Non-constant oscillations

2. Rolling statistics

$$\text{Rolling Mean}(t) \neq \text{constant}$$

3. Autocorrelation Structure

Non-stationary series often have:

- Slow, gradual decay in ACF
- Persistent autocorrelation across many lags

4. Augmented Dickey–Fuller (ADF) Test

Hypotheses:

- H_0: Unit root exists → series is non-stationary
- H_1: Stationary

p-value < 0.05 → reject H_0 → series is stationary.

Autocorrelation and ACF/PACF

Autocorrelation describes how much current values depend on past values.

Autocorrelation Function (ACF)

Measures linear correlation between Y_t and Y_{t-k}.

$$\rho_k = Corr(Y_t, Y_{t-k})$$

Why It Matters

ACF patterns typically guide selection of q (MA term).

- If ACF has a sharp cutoff, MA(q) is appropriate.
- If ACF decays slowly, differencing needed.

Partial Autocorrelation Function (PACF)

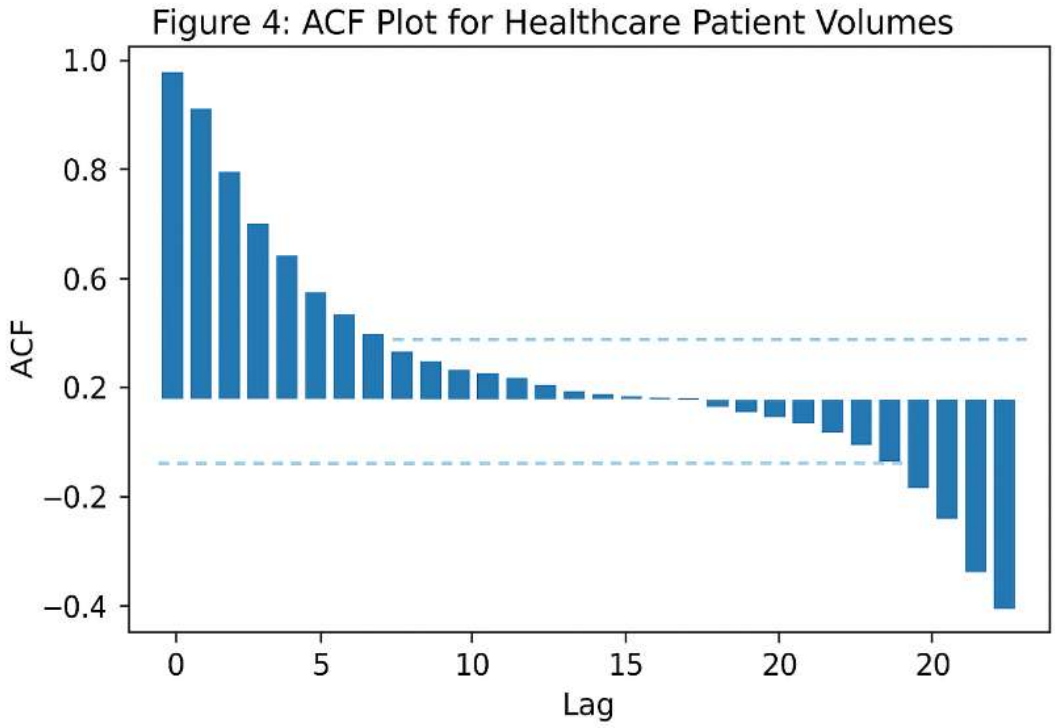
PACF measures correlation between Y_t and Y_{t-k} after controlling for intermediate lags.

PACF helps determine p (AR term):

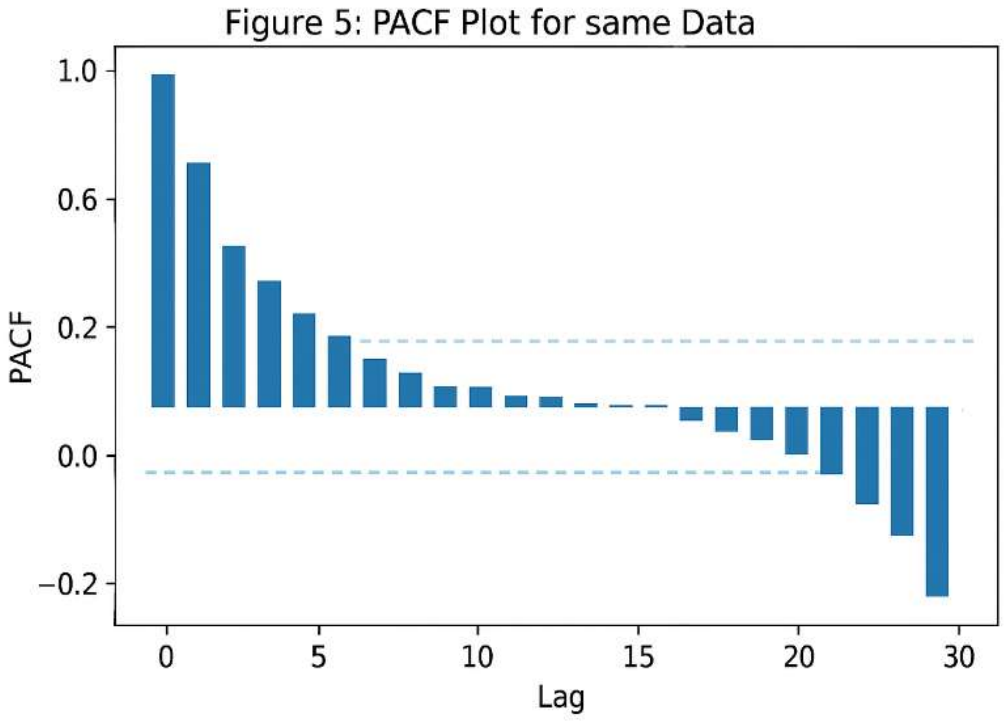
- Sharp cutoff in PACF ⇒ AR(p)
- Slow decay ⇒ require differencing

Diagram Placeholders

- **Figure 4:** ACF plot for healthcare patient volumes



- **Figure 5:** PACF plot for same data



ARIMA Model: Mathematical Foundations

ARIMA stands for:

- AR: Autoregressive
- I: Integrated (Differencing)
- MA: Moving Average

Mathematically:

---

1. Autoregressive Model AR(p)

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

Interpretation:

- Today's ED volume depends partially on recent days.
- Example: Monday ED volumes influenced by Sunday and Saturday patterns.

---

2. Integrated Component I(d)

Used to convert a non-stationary series into stationary form.

$$Y_t' = Y_t - Y_{t-1}$$

Higher-order differencing:

$$Y_t'' = Y_t' - Y_{t-1}'$$

Differencing removes trend and seasonality, making AR and MA terms meaningful.

---

3. Moving Average Model MA(q)

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

Interpretation:

- Today's ED volume depends on past shocks or unexpected deviations.

Example:

- A spike in ED volume yesterday (e.g., storm-related accidents) influences today's volume via MA terms.

ARIMA(p, d, q) Model

Combines all components:

- p: number of lagged observations (AR)
- d: number of differencing steps
- q: number of lagged forecast errors (MA)

Official form (backshift notation):

$$\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t$$

Where:

- $B$is the backshift operator: $B(Y_t) = Y_{t-1}$
- $\phi(B)$encodes AR components
- $\theta(B)$encodes MA components

When ARIMA Fails (Critical for Teaching)

- Sudden regime changes (e.g., pandemic onset)
- Non-linear patterns
- Strong multiple seasonalities
- Interventions (policy changes, staffing changes)
- Missing or irregular data

Understanding limitations is crucial in graduate-level forecasting.

Why ARIMA Works Well in Healthcare

Healthcare operational data—especially ED arrivals—exhibits properties that align very well with ARIMA's structure:

---

1. Short-Term Autocorrelation

Yesterday's ED volume is strongly predictive of today's.

Reasons:

- Local outbreaks
- Weather patterns
- Regular patient behavior cycles

AR(p) captures these dependencies.

---

2. Gradual Volume Changes (Trend)

Large shocks are rare (except during pandemics).

Differencing (I(d)) stabilizes the series.

---

3. Weekly Seasonality

ED demand consistently spikes on Mondays and dips on weekends.

While ARIMA cannot directly handle periodic seasonality, SARIMA (Seasonal ARIMA) can extend this framework to include seasonal terms. Even with non-seasonal ARIMA, weekly cycles influence ACF/PACF patterns.

---

4. Interpretability for Administrators

ARIMA provides:

- Clear statistical foundation
- Transparent coefficients
- Predictive intervals

This is critical for:

- Nurse staffing decisions

- Bed capacity planning
- Budget forecasting

Machine learning models may outperform ARIMA, but administrators trust models they can explain.

---

5. Confidence Intervals Communicate Risk

Healthcare decisions require not just a forecast, but uncertainty bounds.

ARIMA naturally produces:

$$\hat{Y}_{t+h} \pm Z\sqrt{Var(\epsilon)}$$

These intervals allow hospitals to plan for:

- Best case (low demand)
- Expected case
- Worst case (surge conditions)

Why ARIMA Works Well in Healthcare

- ED arrivals show short-term autocorrelation
- Volume changes gradually (suitable for differencing)
- Seasonal ARIMA (SARIMA) can handle weekly patterns
- Forecasts communicate uncertainty via confidence intervals

ARIMA produces interpretable, actionable forecasts, which hospital administrators prefer.

Dataset Description and Preprocessing

Dataset Columns:

Column	Description
date	Calendar date
visits	Number of ED patient arrivals

Data requirements:

- At least 2 years of data
- No missing days (or imputed)
- Stationary or made stationary

Step-by-Step Implementation Tutorial

This section provides a full implementation that can be used in your notebook.

Environment Setup

Install the required libraries:
pip install pandas numpy matplotlib seaborn statsmodels scikit-learn

Why this matters:

- pandas → data manipulation
- numpy → mathematical operations
- matplotlib/seaborn → visualization
- statsmodels → ARIMA modeling
- sklearn → forecast accuracy metrics

Healthcare operations analysts often work in Python due to its transparency and reproducibility, making these libraries industry-standard tools.

Load Data

```
df = pd.read_csv("daily_ed_visits.csv", parse_dates=["date"])

df = df.sort_values("date").set_index("date")

df.head()
```

What this does:

- Reads daily ED visit data
- Ensures the date column is in true datetime format
- Sorts the data chronologically (critical because ARIMA is order-sensitive)
- Sets the date as the index, required for time series modeling

Healthcare interpretation:

ED arrival patterns vary dramatically by date, so properly indexed and chronologically ordered data is fundamental for meaningful analysis.

Plot Time Series

```
plt.figure(figsize=(14,4))

plt.plot(df["visits"])

plt.title("Daily ED Visit Volume")

plt.xlabel("Date")

plt.ylabel("Visits")

plt.show()
```

Purpose of this visualization:

- Helps learners visually inspect trends, seasonality, and outliers
- Allows identification of operational events (e.g., seasonal spikes)
- Creates intuition before mathematical modeling begins

Healthcare relevance:

Many hospitals see predictable winter surges and weekly cycles. Plotting helps identify these patterns early.

Stationarity Testing (ADF Test)

```
from statsmodels.tsa.stattools import adfuller

result = adfuller(df["visits"])

print("ADF Statistic:", result[0])

print("p-value:", result[1])
```

If p > 0.05 → NOT stationary.

Why stationarity testing is essential:

ARIMA assumes the series has constant:

- Mean
- Variance
- Autocorrelation structure

The Augmented Dickey–Fuller (ADF) test determines if a "unit root" exists.

- p > 0.05 → Non-stationary
- p < 0.05 → Stationary

Teaching explanation:

If your ED volume is gradually increasing over time (e.g., due to population growth), ARIMA cannot model the drifting baseline without differencing.

Differencing

```
df_diff = df["visits"].diff().dropna()
```

Why we difference the data:

- Removes trend
- Stabilizes the mean
- Makes autocorrelation patterns interpretable
- Ensures ARIMA's underlying math behaves correctly

Healthcare example:

If ED volumes rise every winter and fall every summer, differencing helps isolate the *stationary component* of patient demand.

ACF & PACF for Parameter Selection

```
plot_acf(df_diff, lags=40)

plot_pacf(df_diff, lags=40)

plt.show()
```

Interpret patterns:

- Slow decay in ACF → differencing needed
- Spike at PACF lag 1 → AR(1) candidate
- Spike at ACF lag 1 → MA(1) candidate

Why these plots matter:

ACF and PACF help identify appropriate ARIMA parameters:

Component	Parameter	Identified by
AR	p	PACF cutoff
I	d	Differencing needed
MA	q	ACF cutoff

Interpretation guidance:

- Slow decay in ACF → Differencing required
- Spike at PACF lag 1 → Strong AR(1) structure
- Spike at ACF lag 1 → Strong MA(1) structure

Healthcare example:

ED visit counts tend to depend on the previous day's volume (lag=1), making AR(1) a common pattern in real hospitals.

Train-Test Split

```
split = int(len(df) * 0.8)

train, test = df.iloc[:split], df.iloc[split:]
```

Why this is critical:

- Prevents data leakage
- Allows objective evaluation
- Mimics real-world forecasting, where future data is unknown

Teaching note:

Time series must be split chronologically—random train-test splits violate temporal dependencies.

Fit ARIMA Model

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

model = ARIMA(train["visits"], order=(1,1,1))

model_fit = model.fit()

print(model_fit.summary())
```

Meaning of order=(1,1,1):

- 1 = AR(1): Last day's volume influences today
- 1 = One differencing step for stationarity
- 1 = MA(1): Last day's forecast error influences today's volume

Summary output interpretation:

- Coefficients indicate strength of relationships
- AIC (Akaike Information Criterion) helps compare multiple models
- Standard errors show statistical significance

Healthcare insight:

In many hospitals, yesterday's ED surge affects today's operations (e.g., backlog), justifying AR(1) structures.

Forecasting

forecast = model_fit.get_forecast(steps=len(test))

pred = forecast.predicted_mean

conf = forecast.conf_int()

Why this matters:

This simulates real-world forecasting where:

- Administrators need predicted ED volumes
- Confidence intervals help plan for uncertainty (e.g., surge planning)

Healthcare example:

If the lower CI suggests 140 patients and the upper CI suggests 230 patients, staffing should be planned for the worst-case scenario.

Plot Forecast vs Actual

```
plt.figure(figsize=(14,4))

plt.plot(train.index, train["visits"], label="Train")

plt.plot(test.index, test["visits"], label="Actual")

plt.plot(test.index, pred, label="Forecast")

plt.fill_between(test.index, conf.iloc[:,0], conf.iloc[:,1], alpha=0.3)

plt.legend()

plt.show()
```

Why this visualization is important:

- Shows how well the model tracks actual ED volumes
- Highlights deviations (e.g., unexpected outbreaks)
- Reveals whether the model is biased

Interpretation guidance:

- If predictions consistently underestimate peaks → model may need seasonal terms
- Confidence intervals widening → uncertainty increases further into the future

Evaluate Accuracy

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(test["visits"], pred)

rmse = mean_squared_error(test["visits"], pred, squared=False)

mae, rmse
```

Interpretation:

- MAE = average daily forecast error
- RMSE = penalizes large errors

Metric meanings:

Metric	Interpretation
MAE	Average daily forecast error
RMSE	Penalizes large errors (important during surges)

Healthcare decision-making value:

- High RMSE during flu season means forecasts are underprepared for extreme spikes.
- Low MAE overall indicates consistent daily accuracy.

Residual Diagnostics

```
residuals = model_fit.resid

plt.figure(figsize=(12,4))

plt.plot(residuals)

plt.title("Residual Plot")

plt.show()

plot_acf(residuals)

plt.title("ACF of Residuals")

plt.show()
```

Residuals should:

- Have mean ≈ 0
- Show no autocorrelation
- Show constant variance

Why diagnostics matter:

A good ARIMA model leaves no structure in residuals.
Residuals should resemble white noise:

- Mean ≈ 0 → no systemic bias
- No autocorrelation → model captured the time dependencies
- Constant variance → stable predictions

Healthcare interpretation:

If residuals show patterns, the model may be missing:

- Day-of-week effects
- Seasonal influenza trends
- Intervention changes (e.g., new triage process)

This tells hospital analysts whether more sophisticated models (SARIMA, Prophet) are needed.

Exercises

Exercise 1: Selecting Optimal ARIMA Parameters

Task:
Try p,q ∈ {0,1,2} and compare AIC values.

Solution Summary:

- Fit ARIMA models in nested loops
- Best model minimizes AIC
- Typically ARIMA(1,1,1) or ARIMA(2,1,1) wins for ED data

Exercise 2: Analyze 14-Day and 30-Day Forecast Stability

Task:
Generate forecasts for 14 and 30 days.
Discuss horizon sensitivity.

Solution Summary:

- Accuracy deteriorates as horizon increases
- Confidence intervals widen
- Long-term predictions less reliable

Exercise 3: Residual Correction

Task:
Analyze residual ACF.
Adjust p or q accordingly.

Solution Summary:

- Residual autocorrelation at lag 1 → increase AR order
- Strong MA terms → increase q

Exercise 4: Healthcare Operational Memo

Task:
Write a memo for hospital leadership summarizing forecast results.

Solution Points:
Students learn communication skills needed in real analytics roles.

Summary and Key Takeaways

This tutorial is crafted to teach ARIMA forecasting through clear pedagogy, rigorous theory, and a meaningful healthcare problem. Learners gain:

- Deep understanding of time series structure
- Hands-on experience building ARIMA models
- Ability to interpret model outputs
- Skill to communicate analytics to non-technical stakeholders

This aligns directly with INFO 7390's focus on becoming educators of data science, not just implementers.