Statistics and Data Analysis
Unit 06 – Lecture 04 Notes
Forecasting Fundamentals and ARIMA

Tofik Ali

February 17, 2026

Topic

ARIMA models and forecasting workflow (overview).

How to Use These Notes

These notes are written for students who are seeing the topic for the first time. They follow the slide order, but add the missing 'why', interpretation, and common mistakes. If you get stuck, look at the worked exercises and then run the Python demo.

Course repository (slides, demos, datasets): `https://github.com/tali7c/Statistics-and-Data-Analysis`

Time Plan (55 minutes)

- 0–10 min: Attendance + recap of previous lecture
- 10–35 min: Core concepts (this lecture's sections)
- 35–45 min: Exercises (solve 1–2 in class, rest as practice)
- 45–50 min: Mini demo + interpretation of output
- 50–55 min: Buffer / wrap-up (leave 5 minutes early)

Slide-by-slide Notes

Title Slide

State the lecture title clearly and connect it to what students already know. Tell students what they will be able to do by the end (not just what you will cover).

Quick Links / Agenda

Explain the structure of the lecture and where the exercises and demo appear.

- Overview

- ARIMA
- Differencing
- Exercises
- Demo
- Summary

Learning Outcomes

- Define ARIMA(p,d,q) at a high level
- Explain differencing (d) to remove trend
- Explain p and q meaning (AR and MA orders)
- Describe time-based train/test split for forecasting

Why these outcomes matter. **Trend** is a long-term upward or downward movement. Trend changes the mean over time, which often creates non-stationarity. Many forecasting models handle trend by differencing or by explicitly modeling trend. **Differencing** transforms a series by subtracting the previous value: $y_t - y_{t-1}$. It removes trend and can help achieve stationarity. Over-differencing can add noise, so use the smallest differencing order that works.

ARIMA: Key Points

- p: AR order
- d: differencing order
- q: MA order

Explanation. **Differencing** transforms a series by subtracting the previous value: $y_t - y_{t-1}$. It removes trend and can help achieve stationarity. Over-differencing can add noise, so use the smallest differencing order that works.

Differencing: Key Points

- First difference: y_t - y_t-1
- Often stabilizes mean
- Over-differencing adds noise

Explanation. **Differencing** transforms a series by subtracting the previous value: $y_t - y_{t-1}$. It removes trend and can help achieve stationarity. Over-differencing can add noise, so use the smallest differencing order that works.

Exercises (with Solutions)

Attempt the exercise first, then compare with the solution. Focus on interpretation, not only arithmetic.

Exercise 1: Meaning of d

What does d=1 mean?

Solution

- First differencing once.

Walkthrough. **Differencing** transforms a series by subtracting the previous value: $y_t - y_{t-1}$. It removes trend and can help achieve stationarity. Over-differencing can add noise, so use the smallest differencing order that works.

Exercise 2: Chronological split

Why not random split in time series?

Solution

- Random split leaks future information.

Walkthrough. A **time series** is data indexed by time (daily sales, hourly sensor readings). The key difference from 'normal' data is that order matters and observations are often correlated over time. Many standard ML assumptions (IID, random split) break for time series.

Exercise 3: Trend fix

Series has strong upward trend. Name one simple step.

Solution

- First differencing.

Walkthrough. **Trend** is a long-term upward or downward movement. Trend changes the mean over time, which often creates non-stationarity. Many forecasting models handle trend by differencing or by explicitly modeling trend. **Differencing** transforms a series by subtracting the previous value: $y_t - y_{t-1}$. It removes trend and can help achieve stationarity. Over-differencing can add noise, so use the smallest differencing order that works.

Mini Demo (Python)

Run from the lecture folder:

```
python demo/demo.py
```

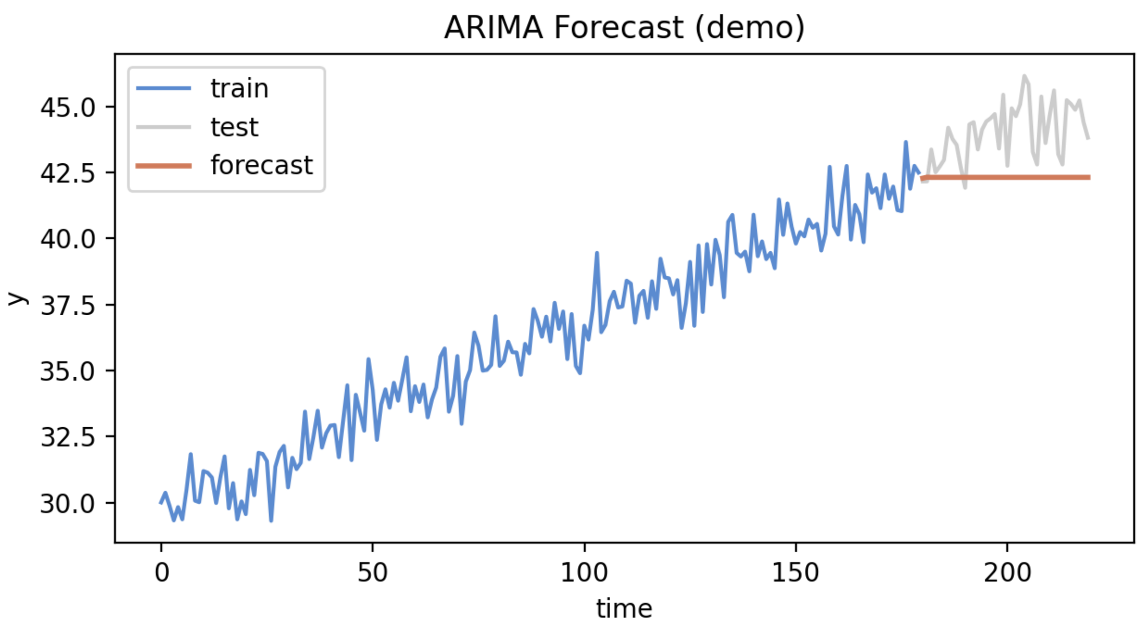
Output files:

- `images/demo.png`
- `data/results.txt`

What to show and say.

- Creates a trending series, differences it, and fits an ARIMA-style model.
- Shows why differencing helps stabilize the mean (stationarity).
- Use residual output to motivate diagnostics after fitting.

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why do we check residuals after fitting an ARIMA model?

Suggested answer (for revision). We check residuals to see if remaining structure (autocorrelation/patterns) is still unexplained; good residuals look like white noise.

References

- Montgomery, D. C., & Runger, G. C. *Applied Statistics and Probability for Engineers*, Wiley.
- Devore, J. L. *Probability and Statistics for Engineering and the Sciences*, Cengage.
- McKinney, W. *Python for Data Analysis*, O'Reilly.

Appendix: Slide Deck Content (Reference)

The material below is a reference copy of the slide deck content. Exercise solutions are explained in the main notes where applicable.

Title Slide

https://github.com/tali7c/Statistics-and-Data-Analysis

Quick Links

Overview ARIMA Differencing Exercises Demo Summary

Agenda

- Overview
- ARIMA
- Differencing
- Exercises
- Demo
- Summary

Learning Outcomes

- Define ARIMA(p,d,q) at a high level
- Explain differencing (d) to remove trend
- Explain p and q meaning (AR and MA orders)
- Describe time-based train/test split for forecasting

ARIMA: Key Points

- p: AR order
- d: differencing order
- q: MA order

Differencing: Key Points

- First difference: y_t - y_t-1
- Often stabilizes mean
- Over-differencing adds noise

Exercise 1: Meaning of d

What does d=1 mean?

Solution 1

- First differencing once.

Exercise 2: Chronological split

Why not random split in time series?

Solution 2

- Random split leaks future information.

Exercise 3: Trend fix

Series has strong upward trend. Name one simple step.

Solution 3

- First differencing.

Mini Demo (Python)

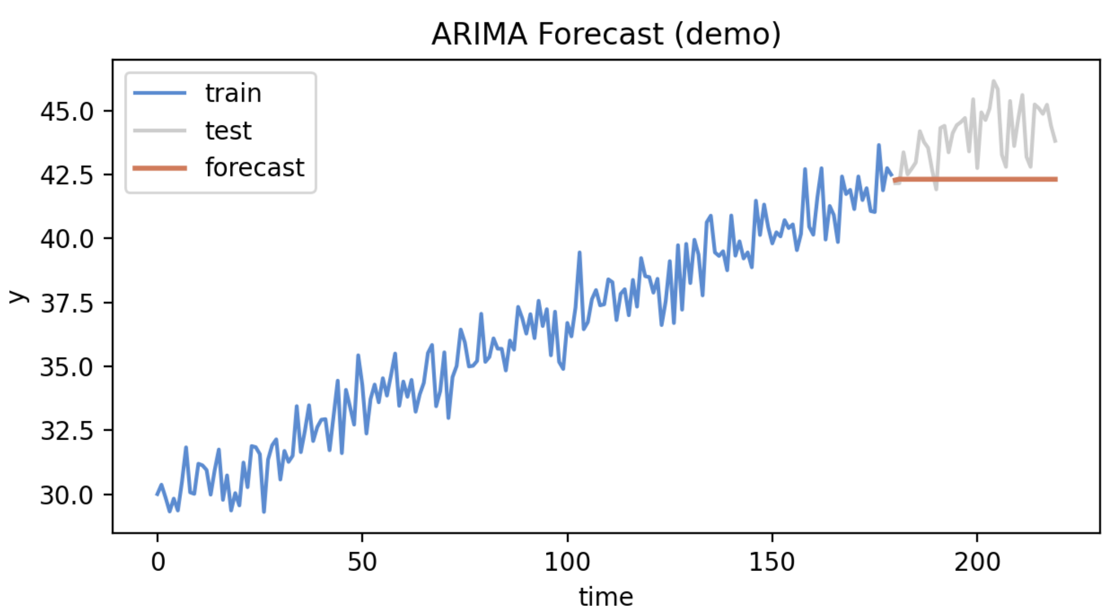
Run from the lecture folder:

```
python demo/demo.py
```

Outputs:

- `images/demo.png`
- `data/results.txt`

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why do we check residuals after fitting an ARIMA model?