



# TIME SERIES 501

Lesson 5: ARMA and ARIMA Models

# Learning Objectives

You will be able to do the following:

- Review and recall key ideas from previous lessons.
- Describe an ARMA model and its stages.
- Describe the ARIMA and SARIMA models and choose parameters.
- List ARMA, ARIMA, and SARIMA model assumptions.
- Use Python\* to construct ARMA, ARIMA, and SARIMA models.

# ARMA MODEL

# ARMA Model

The ARMA model (also known as the Box-Jenkins approach) combines two models:

- An autoregressive model
- A moving-average model

# ARMA Model Notes

Some things to keep in mind when dealing with ARMA models:

- The time series is assumed to be stationary.
- A good rule of thumb is to have at least 100 observations when fitting an ARMA model.

# ARMA Model Stages

There are three stages in building an ARMA model:

- Model identification
- Model estimation
- Model evaluation

# ARMA Model Identification

Confirm the following:

- The time series is stationary.
- Whether the time series contains a seasonal component.

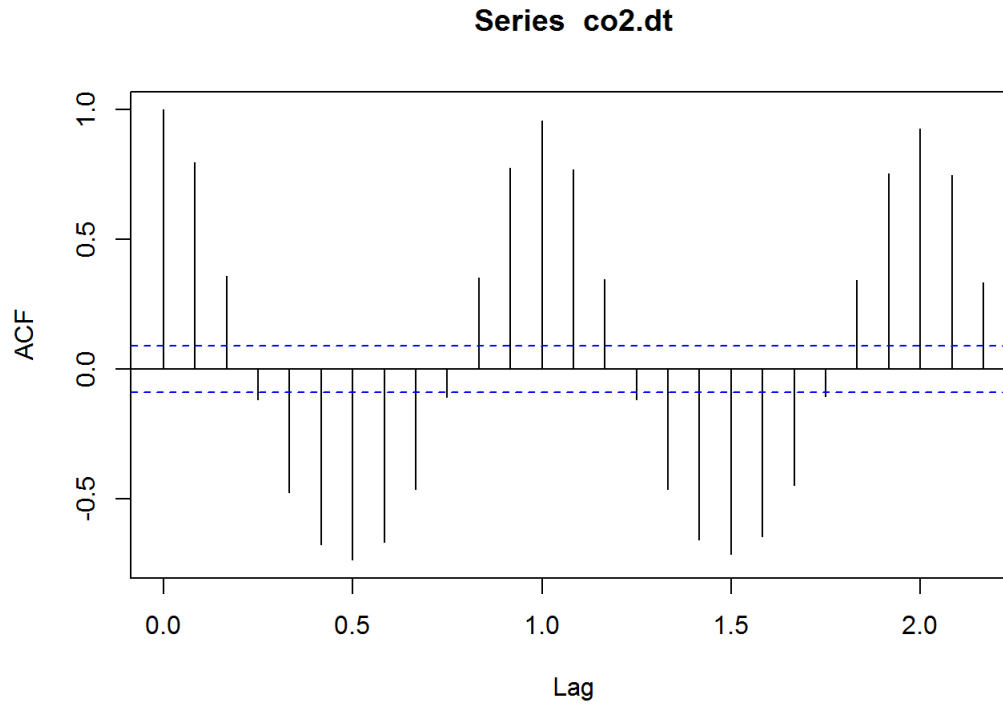
# Determine Seasonality

You can determine whether seasonality is present by using the following:

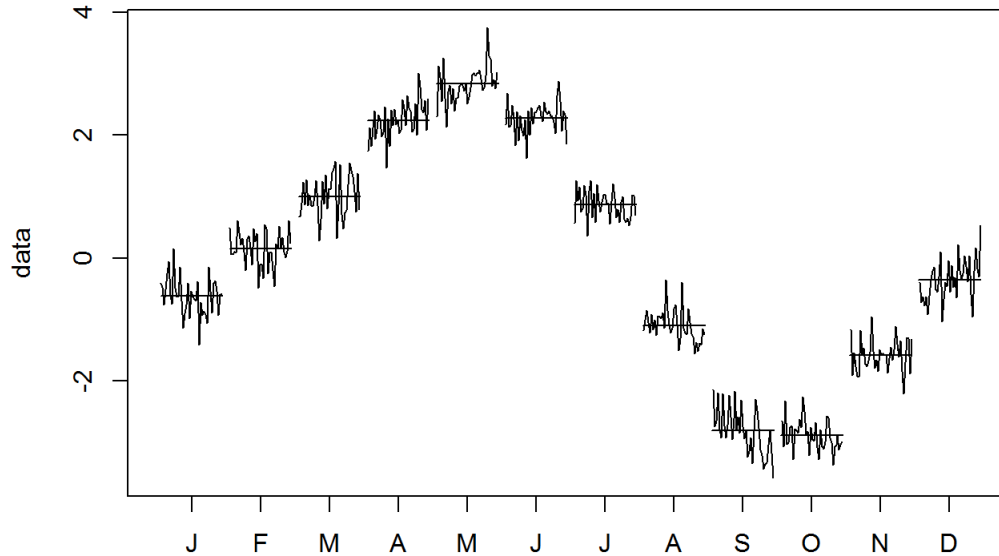
- Autocorrelation plot
- Seasonal subseries plot
- Spectral plot



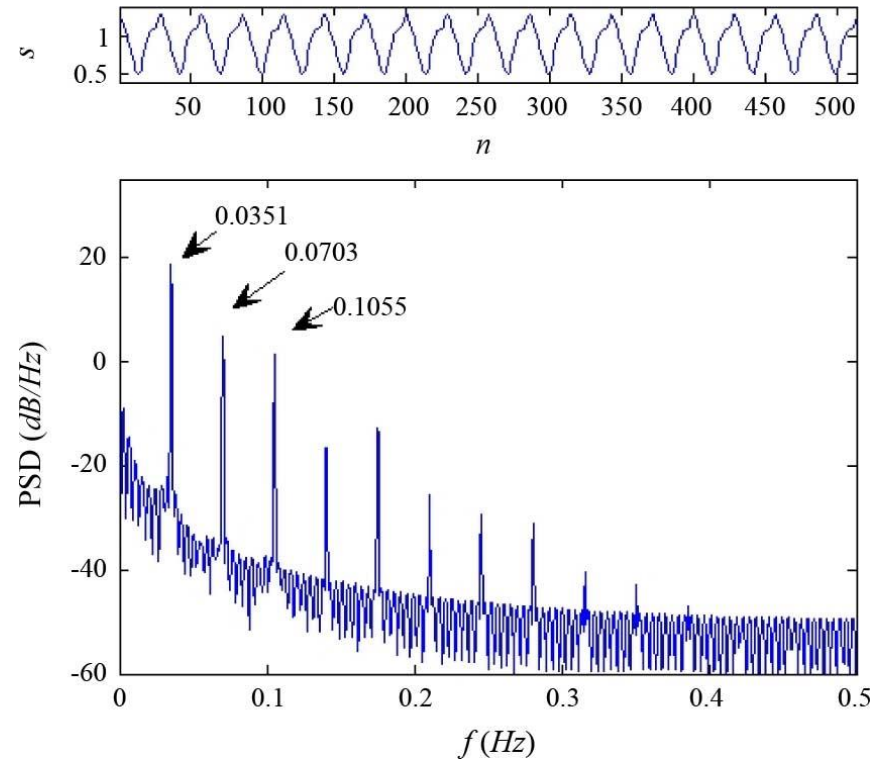
# Autocorrelation Plot



# Seasonal Subseries Plot



# Spectral Plot



# Identifying $p$ and $q$

After the time series is stationary, you must identify the order of the AR and MA models. As you learned in Lesson 4, you can do this by looking at the following:

- Autocorrelation plot
- Partial autocorrelation plot

# AR(p)

How do you determine the order  $p$  of the AR model?

- Plot 95% confidence interval on the partial autocorrelation plot. (Most modern software does this automatically.)
- Choose lag  $p$  such that partial autocorrelation becomes statistically zero for  $p+1$  and beyond.

# MA(q)

How do you determine the order  $q$  of the MA model?

- Plot 95% confidence interval on the autocorrelation plot. (Most modern software does this automatically.)
- Choose lag  $q$  such that autocorrelation becomes statistically zero for  $q+1$  and beyond.

# Guidelines

SHAPE	MODEL
Exponential decaying to zero	AR models
Alternating positive and negative decaying to zero	AR models
One or more spikes, the rest are close to zero	MA model
Decay after a few lags	Mixed AR and MA
All zero or close to zero	Data is random
High values at fixed intervals	Include seasonal AR term
No decay to zero	Series is not stationary

# ARMA Model Estimation

Estimating the parameters of an ARMA model can be a complicated, nonlinear problem.

- Nonlinear least squares and maximum likelihood estimation (MLE) are common approaches.
- Many modern software programs will fit the ARMA model for you.



# ARMA Model Validation

How do you know if your ARMA model is any good?

- The residuals will approximate a Gaussian distribution (aka white noise).
- Otherwise, you'll need to iterate to obtain a better model.

# ARIMA & SARIMA MODELS

# ARIMA Model

ARIMA stands for autoregressive integrated moving average. ARIMA models have three components:

- AR model
- Integrated component (more on this shortly)
- MA model

# ARIMA Model Details

There are a few things you should know about ARIMA models:

- The ARIMA model is denoted  $\text{ARIMA}(p, d, q)$ .
- $p$  is the order of the AR model.
- $d$  is the number of times to difference the data.
- $q$  is the order of the MA model.
- $p$ ,  $d$ , and  $q$  are nonnegative integers.

# Differencing

It turns out that differencing nonstationary time series data one or more times can make it stationary. That's the integrated (I) component of ARIMA.

- d is the number of times to perform a lag-1 difference on the data.
- d=0: no differencing
- d=1: difference once
- d=2: difference twice

$$Y_i = Z_i - Z_{i-1}$$

# SARIMA Model

SARIMA is short for seasonal ARIMA. This model is used to remove seasonal components.

- The SARIMA model is denoted SARIMA(p, d, q)(P, D, Q).
- P, D, and Q represent the same as p, d, and q but they are applied across a season (for example, yearly).

$$Y_i = Z_i - Z_{i-M}$$

- M = one season

# Choosing ARIMA/SARIMA Parameters

How do you choose  $p$ ,  $d$ ,  $q$  and  $P$ ,  $D$ ,  $Q$ ?

- Visually inspect a run sequence plot for trend and seasonality.
- Generate an ACF Plot.
- Generate a PACF Plot.
- Rule of thumb:  $p + q \leq 3$ .

# Automated Selection

Some modern software automatically selects model parameters for you.

- The software tests a wide range and combination of parameters.
- An optimization metric is used to determine the optimal combination.
- Beware that different software implementations may provide a different selection of parameters due to the way the algorithms are implemented under the hood.



# ARIMA Summary

Here's what you should have learned about ARIMA:

- Flexible family of models that capture autocorrelation.
- Based on strong statistical foundation.
- Requires stationary time series.
- Choosing optimal parameters manually requires care.
- Some software finds parameters automatically.
- Can be challenging to explain and interpret.
- Can be prone to overfitting.

# MODEL ASSUMPTIONS

# Assumptions

ARMA, ARIMA, SARIMA assumptions:

- Time-series data is stationary.
- If nonstationary, remove trend, seasonality, apply differencing, and so on.
- Remember that stationary data has no trend, seasonality, constant mean, and constant variance.
- Therefore, the past is assumed to represent what will happen in the future in a probabilistic sense.

# **APPLICATIONS IN PYTHON**

# Use Python to Fit ARMA, ARIMA, and SARIMA Models

Next up is a look at applying these concepts in Python.

- See notebook entitled *ARIMA\_SARIMA\_student.ipynb*

# Learning Objectives Recap

In this session you have done the following:

- Reviewed key ideas from previous lessons.
- Learned about the ARMA model and its stages.
- Learned what the ARIMA and SARIMA models are and how to choose parameters.
- Saw ARMA, ARIMA, and SARIMA model assumptions.
- Used Python to construct ARMA, ARIMA, and SARIMA models.

# Legal Notices and Disclaimers

This presentation is for informational purposes only. INTEL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at [intel.com](https://www.intel.com).

Sample source code is released under the [Intel Sample Source Code License Agreement](#).

Intel, the Intel logo, the Intel. Experience What's Inside logo, and Intel. Experience What's Inside are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All rights reserved.

