

TIME SERIES ANALYSIS FOR DATA SCIENTISTS (FORGY)



Charles Forgy

Preface

Welcome to *Time Series Analysis for Data Scientists*! This online textbook grew out of the course notes for a time series analysis course at the University of Maryland, College Park given in the fall semester of 2025. This book aims to be largely self-contained with references to more advanced texts when necessary.

Background

I was motivated to write this book due to a frustration over not being able to find a text at a level I considered appropriate for data science students. On the one hand, there are multiple outstanding statistics texts covering the topic in great depth. However, the statistical and mathematical rigor of these texts can often cause students to become lost in the details without a solid understanding of what is being done and why. On the other hand, texts aimed at business students often go into too little depth for the purposes of data science and focus narrowly on econometrics applications at the expense of broader use cases relevant to data scientists.

This text aims to chart a middle course. Statistical and mathematical derivations are provided with the aim of equipping students with an understanding of the motivations, mechanisms, and limitations of various methods. However, we stop short of the level of rigor required for academic statisticians looking to derive novel methodologies. Where necessary, this book contains references to classics such as [Shumway & Stoffer \(2025\)](#) or [Brockwell & Davis \(1991\)](#) for more advanced students looking for additional depth. This book also contains topics not commonly covered in many time series classes that can be valuable for practicing data scientists such as SIR models from epidemiology and deep learning methodologies.

Most chapters contain one or more problems. Some are conceptual questions designed to help you put what you've learned into a broader context. Others are mathematical or statistical, designed to reinforce the underlying concepts. In the latter case, answers are generally provided in a hidden dropdown. I strongly encourage you to attempt the problems before looking at the solutions in order to gain the maximum benefit from this book.

Mathematical Level

This book is aimed at students who have a background in a quantitative discipline such as pure data science, physics, electrical engineering, or econometrics. The derivations generally assume at least two semesters of calculus and one semester of linear algebra as well as familiarity with the basics of probability and statistics. More advanced courses such as multi-variate calculus, Fourier analysis, and mathematical statistics are useful but not necessary.

Chapter 2 presents a refresher on several mathematical and statistical topics necessary for later chapters. Similarly, Chapter 5 covers an introduction to Fourier analysis. For many students it may be sufficient to skim the material to ensure familiarity. If you are not familiar with the topics the intention is that these chapters (with the accompanying worked problems) will serve as a sufficient introduction to master ensuing time series specific applications.

License

This book is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike license. In short, this means you are free to use this book, either in part or its entirety, for any non-commercial purposes provided you provide clear attribution. This work should be referenced as

Time Series Analysis for Data Scientists by Charles Forgy, PhD.

I would ask that any instructor or lecturer using substantial portions of the book please send me an email at ccforgy@umd.edu to help understand the contexts in which the book finds use and how it can be improved.

Contributing

This book is maintained at <https://github.com/charlescforgy/time-series-book.git>. If you find any errors or

simply areas you'd like to see expanded upon I would be grateful if you open a pull request. If you are not comfortable with GitHub, you can also email me directly at ccforgy@umd.edu.

Anyone interested in getting more involved in this open source project to bring rigorous but applicable time series analysis tools to the public is invited to contribute. There are three areas in particular for which we are looking for contributors:

1. Copy editing for grammar and syntax
2. Translation into languages other than English
3. Graphic and website design to improve layout and readability

If you have other ideas for ways to improve the book feel free to drop me a line.

Best of luck in your time series journey!

References

1. Shumway, R. H., & Stoffer, D. S. (2025). Time Series Analysis and Its Applications: With R Examples. In *Springer Texts in Statistics*. Springer Nature Switzerland. [10.1007/978-3-031-70584-7](#)
2. Brockwell, P. J., & Davis, R. A. (1991). Time Series: Theory and Methods. In *Springer Series in Statistics*. Springer New York. [10.1007/978-1-4419-0320-4](#)

Time Series Analysis for Data Scientists (Forgy)

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (<https://LibreTexts.org>) and like the thousands of other texts available within this powerful platform, it is freely available for reading, printing, and "consuming."

The LibreTexts mission is to bring together students, faculty, and scholars in a collaborative effort to provide an accessible, and comprehensive platform that empowers our community to develop, curate, adapt, and adopt openly licensed resources and technologies; through these efforts we can reduce the financial burden borne from traditional educational resource costs, ensuring education is more accessible for students and communities worldwide.

Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects. Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



LibreTexts is the adaptable, user-friendly non-profit open education resource platform that educators trust for creating, customizing, and sharing accessible, interactive textbooks, adaptive homework, and ancillary materials. We collaborate with individuals and organizations to champion open education initiatives, support institutional publishing programs, drive curriculum development projects, and more.

The LibreTexts libraries are Powered by [NICE CXone Expert](#) and were supported by the Department of Education Open Textbook Pilot Project, the California Education Learning Lab, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptations contact info@LibreTexts.org or visit our main website at <https://LibreTexts.org>.

The LibreTexts name, logo, and book covers are trademarked by LibreTexts, Inc. (a 501(c)(3) not-for-profit organization) and protected against unauthorized use.

This text was compiled on 10/10/2026

TABLE OF CONTENTS

Preface

Licensing

1: Introduction

- 1.1: What is a Time Series?
- 1.2: What Makes Time Series Analysis Unique?
- 1.3: Time Series Analysis vs. Forecasting
- 1.4: Basic Time Series Models
- 1.5: Additional Resources

2: Background Math

- 2.1: Introduction
- 2.2: Geometric Series
- 2.3: Complex Numbers
- 2.4: Properties of Variance and Covariance
- 2.5: Information Criteria

3: Autocovariance and Autocorrelation

- 3.1: Introduction
- 3.2: Autocovariance
- 3.3: Stationarity
- 3.4: Autocorrelation
- 3.5: Cross-Covariance and Cross-Correlation
- 3.6: Estimation of Autocovariance, Autocorrelation, and Cross-Correlation

4: Methods to Enforce Stationarity

- 4.1: Introduction
- 4.2: Time Series with Trends
- 4.3: Differencing and the Backshift Operator
- 4.4: Time Series with Seasonality
- 4.5: Increasing Variance and the Log Transform

5: Time Series Decomposition

- 5.1: Introduction
- 5.2: Classical Decomposition
- 5.3: Local Regression
- 5.4: STL Decomposition

6: ARMA Models

- 6.1: Introduction
- 6.2: Autoregressive Models
- 6.3: Moving Average Models

7: Appendices**Index****Glossary****Detailed Licensing**

This page titled [Time Series Analysis for Data Scientists \(Forgy\)](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

Licensing

A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.

This page titled [Licensing](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

1: INTRODUCTION

1.1: What is a Time Series?

1.2: What Makes Time Series Analysis Unique?

1.3: Time Series Analysis vs. Forecasting

1.4: Basic Time Series Models

1.5: Additional Resources

This page titled 1: [Introduction](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

1.1: What is a Time Series?

Data Type

By the strict mathematical definition, a time series is defined as a series of observations ordered by time. Note that we have not defined the observations as *numeric* values, a time series can be any sort of observation, including not only numeric data such as stock prices or infection rates, but also observations such as:

- Images indexed by time (i.e. videos)
- System health (healthy, warning, etc.) every hour
- User sentiment on social media every day
- The geographic location of IP addresses visiting our website every second

While the above are all examples of time series, this text will focus on purely numeric data. Non-numeric data will be assumed to be encoded in a numeric form such as Healthy=0, Warning=1, etc.

Serial Correlation

Consider the following thought experiment:

Thought Exercise 1.1.1

Every day for a year, you flip a coin and record the date and either -1 (tails) or 1 (heads).

By the strict definition given above, because this is indexed by time it qualifies as a time series. However, in the context of business analysis and data science, we generally mean something more when we refer to a set of observations as a "time series." We will only consider a series to be a time series if the past has predictive value, often (but not always) due to causation. Knowing a full year of coin flips for a fair coin provides no additional ability to predict the next flip—though a strong preponderance of one result may force us to re-evaluate our priors regarding the fairness of the coin.

Thought Exercise 1.1.2

You start a running tally at 0. Every day for a year you flip a coin and either subtract 1 (tails) or add 1 (heads) and record the date and new tally.

In this case, knowing the previous day's value *will* help you predict the following value. For example, given that the current value is 8, we know that the next value must either be 7 or 9; put slightly differently, the series exhibits *serial correlation* meaning that the observation at time t is highly correlated with the observation at time $t-1$, slightly less correlated with the observation at $t-2$, and so on. We will see in later chapters that this is an example of a *random walk*.

Other cases may not be as clear-cut. Imagine we are working at a medical school analyzing patient statistics such as blood pressure, heart rate, and temperature. Let's imagine we analyze these records for a week; is this a time series? What about a year? What about 50 years? It seems unlikely that a single week of data will contain any useful trends or predictive value stemming from its ordering. On the other hand, 50 years of data will almost certainly contain valuable data on health trends and the direction we can expect vitals to take in the future.

So where do we draw the line as to what is or is not a time series? As with so much of data science, this question involves not only statistics, but *domain expertise*. There is no general rule, as a practicing data scientist you need to work with experts in the field to make an informed decision for your use case and goals.

Go to a newspaper or news site and find a series that could be considered a time series. Do they treat it as such? Are the variables

numeric, categorical, or something else?

Exercise 1.1.1

Go to a newspaper or news site and find a series that could be considered a time series. Do they treat it as such? Are the variables numeric, categorical, or something else

This page titled [1.1: What is a Time Series?](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

1.2: What Makes Time Series Analysis Unique?

Time series analysis has multiple important differences from most of data science that necessitate unique methodologies. This section covers a few of the most important.

Temporal Ordering

If you take nothing else away from this book, please take away this warning:

Temporal Ordering

The temporal ordering of a time series must always be maintained, and predictions must always be from past observations to future ones.

One important ramification of the above is that standard data science resampling techniques such as bootstrapping are not valid. Bootstrapping is clearly invalid as shuffling the observations will destroy any trends present in the data. Standard cross-validation techniques are also problematic as they do not guarantee we will not predict the past based on the future. One variation that can be used is the *block bootstrap*, in which we train our model on a given number of training and target observations, say years 1–51–5 for training to predict year 6, years 2–62–6 to predict 7, etc. Variations include the *expanding window bootstrap* in which we train on a given block, again say 1–51–5 and use year 6 as the target observation, then train on 1–61–6 and use 7 as the target, and so on.



Figure 1.2.1

Serial Correlation

As mentioned in [the previous section](#), an important aspect—arguably the *central* aspect—of time series is the serial correlation. Beyond the [temporal ordering mentioned above](#), this also results in the assumption of *independent and identically distributed* (*iid*) being invalid. Time series will usually violate the first "i" (independence), later chapters will discuss models for time series that also violate the "id" (identical distribution) as well.

The lack of *iid* means that we cannot rely on many of the hypothesis tests we're used to in broader data science such as ANOVA and t-tests (at least not on the raw data). However, after covering autocovariance and autocorrelation, we will see that the same lack of independence gives us powerful tools for time series analysis.

Non-Necessity of Features

Another aspect differentiating time series analysis from broader data science is the ability to generate predictions without features, or as they tend to be known in time series analysis *exogenous variables*. Given a single univariate time series, say historical stock prices or infection rates, it is perfectly reasonable to ask what we expect future values to be.

Indeed, we will spend most of the first half of this book dealing exclusively with univariate time series in which the same variable is both the feature and the target. Later chapters will introduce models such as ARIMAX that incorporate additional features.

Time Series as Processes

Many data science use cases involve target variables that are inherently discrete from each other. For example, the set of patients for whom we're predicting the risk of a disease given their blood work is inherently discrete by individual. On the other hand, the target variables in time series analysis are usually understood to be samples of an underlying continuous *process* taken at discrete intervals.

For example, it is not reasonable to ask "What is the disease risk of patient 1.5?". In contrast, given the temperature at time $t=1t=1$ and $t=2t=2$, it is perfectly reasonable to interpolate the temperature at time $t=1.5t=1.5$. Using discrete samples to try to understand continuous processes can be the source of great confusion and consternation among data scientists, electrical engineers, and any other discipline concerned with *signal processing*.

A particular concern is *aliasing*, in which an excessively low sampling rate can introduce spurious patterns into the data.

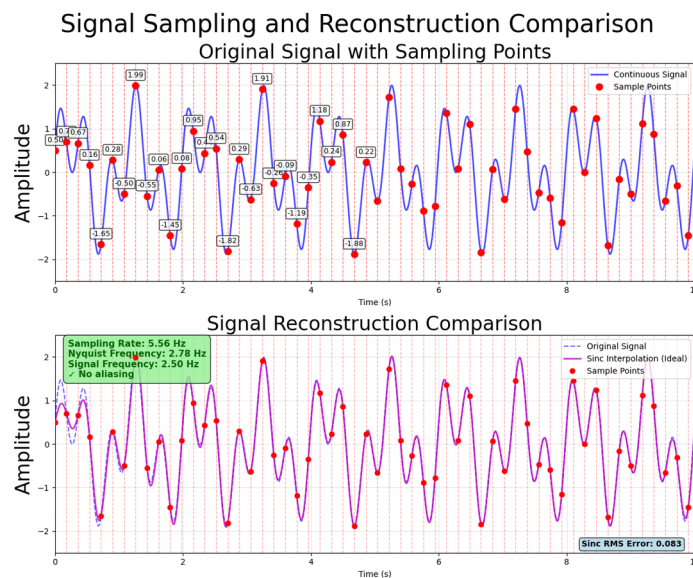


Figure 1.2.2: Sampling correctly reproducing the original signal.

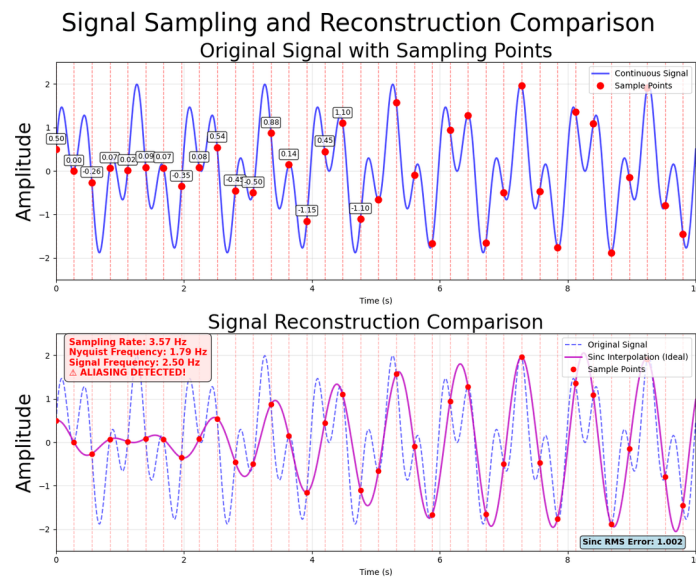


Figure 1.2.3: Sampling exhibiting aliasing distorting the reconstructed signal.

Exercise 1.2.1

Music is often sampled at around 44.1 kHz, despite the fact that the maximum frequency perceptible by humans is around 15–20 kHz. Why is this high sampling rate necessary? You may want to revisit this question after reading the chapter on Fourier analysis

Interestingly, the exact same phenomenon can occur in images, where it is responsible for the *Moiré effect*. The Moiré effect often arises due to jpeg compression of highly textured images. Ultimately, the cause is the same as with time series; sampling a continuous scene with too few discrete pixels introduces spurious patterns.



1.2.4: Original image (left) and image after jpeg compression (right) exhibiting Moiré patterns (images from [Wikipedia](#) used under CC BY-SA 3.0 license).

This page titled [1.2: What Makes Time Series Analysis Unique?](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

1.3: Time Series Analysis vs. Forecasting

One aspect that time series share with broader data science is that we are interested both in understanding historical patterns and forecasting or predicting future values^[1]. Let's take a look at some of the methods and goals of each.

Analysis

Time series *analysis* is concerned with the underlying structure of the time series and how values relate to one another. Typical questions asked in analysis include

- Does a higher stock market return today increase the probability of a low return tomorrow?
- What time of day sees the most visitors to our website?
- How does colder weather affect flu transmission?
- What factors are most closely associated with our supply chain failing to deliver goods on time? Are there warning signs that our supply chain is close to being overloaded?
- Is there any correlation between the number of visitors to our park and wildlife migration?

We will explore multiple methods throughout this book focused on such questions. The workhorse of this analysis will be the ARIMA family of models, but other techniques such as STL Decomposition and cross-correlation analysis will also factor in heavily.

Forecasting

Time series *forecasting* is primarily concerned with predicting future values. Forecasting is often an outgrowth of analysis, for example once we've constructed an ARIMA model of our series we can also use this model to provide predictions. Pure forecasting techniques can also rely on methods such as deep learning or boosted forests, which can be extremely accurate but tend to offer little insight into the underlying behavior of the time series.

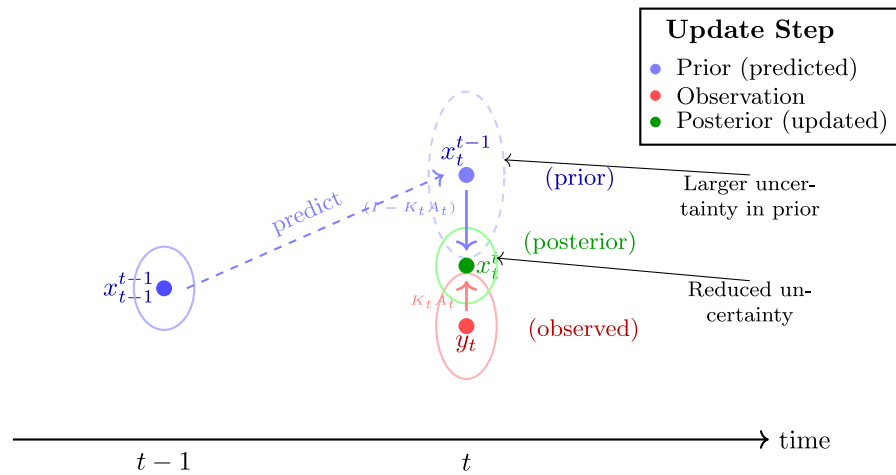
Of course, it is not uncommon in data science to be interested in both analysis and forecasting. In other cases, clients may be interested in forecasting but only willing to accept forecasts that have clear motivations and interpretations. In such scenarios, use of a single model such as ARIMA will often serve both as our analysis tool and (interpretable) prediction method.

Filtering and Smoothing: Predicting the Present

A special set of sub-cases that does not fall neatly into either analysis or forecasting is the problem of *filtering* and *smoothing*. We will delve into the difference between the two terms in future chapters. For now, it is sufficient to think of them as “predicting the present.”

You may well be asking why we would ever need to predict the present—aren't we observing it right now? Filtering problems arise whenever we have less than perfect fidelity in our models and sensors, especially when we wish to combine data from multiple noisy sensors in *sensor fusion*. A common example might be tracking an aircraft on radar. Our analysis of its trajectory suggests it should be at one location, while our radar tracking places it elsewhere. To add to the confusion, perhaps the aircraft itself is transmitting a location agreeing with neither. Filtering and smoothing deal with averaging together disparate and noisy observations to obtain a better estimate than any single sensor or model could provide.

For those interested, we provide a sneak peek at a widely-used type of filter known as a *Kalman filter* below.



1.3.1: Kalman filter demonstrating combining prior forecasts with current noisy sensor data.

Distinguishing Analysis from Forecasting

It's relatively easy to build competitions and leaderboards for forecasting. In contrast, it's difficult to objectively rank insight into historical patterns. For this reason, new data scientists often focus nearly exclusively on the forecasting portion of time series, with their interest in analysis confined to its use to create forecasting models.

While entirely understandable, this mindset misses the broader question of “What is the best way to help my client or employer?”. There are instances in which your goal as a data scientist is confined to improving forecasts to, say, ensure adequate cloud compute power is available. However, it is at least as common to be tasked with time series analysis as part of a broader project that requires insights into trends or causative factors. In such cases, highly interpretable pure analysis methods will generally provide far more value than exquisite but opaque forecasting methods. As always in data science, a thorough understanding of the business goals and use cases must precede any decisions of model choice.

Exercise 1.3.1

Go to a time series model leaderboard or benchmark and look at the data set(s) used. Are they primarily focused on forecasting or analysis? How might the competition or benchmark change if the focus changed?

Footnotes

Some texts define prediction and forecasting as being subtly different. In particular, “Prediction” is sometimes used to mean the general task of predicting an unknown target variable from known features, while “forecasting” is reserved for time series analysis in which we are forecasting future values of the same series we are observing. This book will not make such distinctions and will use the terms interchangeably.

1.4: Basic Time Series Models

White Noise

A central model we will use throughout this textbook is **white noise**.

Whiteness of Noise

The reason for the description “white” will be analyzed in future chapters.

In the context of time series, white noise is defined as a sequence of random variables w_t with mean 0 and (finite) variance σ_w^2 , denoted as:¹

$$w_t \sim wn(0, \sigma_w^2) \quad (1.4.1)$$

In cases in which the mean is not zero, we can always subtract the mean to treat the remaining noise as white. We will often require the noise to be independent and identically distributed (iid). In particular, we will frequently use Gaussian white noise:

$$w_t \sim \mathcal{N}(0, \sigma_w^2) \quad (1.4.2)$$

Noise vs. Shocks

Disciplines such as econometrics use the term shocks instead of noise, often denoting shocks as ϵ_t . The underlying models and statistics remain unchanged.

Visualizing White Noise

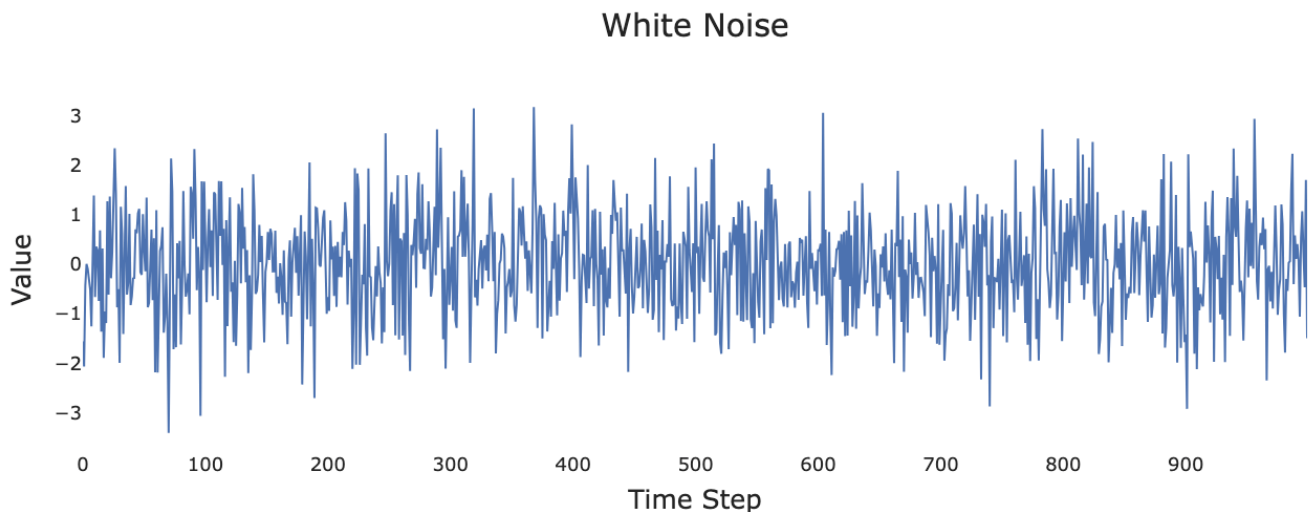
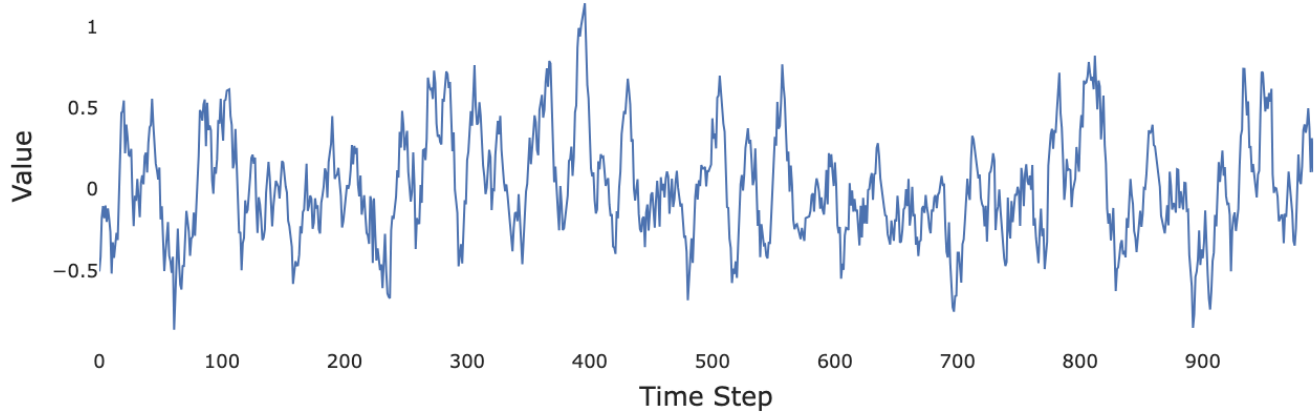


Figure 1.4.1: Simulated white noise process.

Smoothed White Noise with Kernal Size 10



Smoothed version of white noise.

Figure 1.4.2: Smoothed version of white noise.

The following code lets you play around with different values for the noise and distributions. Note that if you are reading this in GitHub pages you will not be able to run the cell, you'll either need to download the book or simply copy and paste the code into a Jupyter notebook.

```
import numpy as np
import matplotlib.pyplot as plt

standard_deviation = 1.0 # Play around with this parameter!
white_noise = np.random.normal(loc=0.0, scale=standard_deviation, size=1000)
# We can also look at alternative distributions. Uncomment the following to use a Cauchy d
# white_noise = np.random.standard_cauchy(size=1000)
plt.plot(np.arange(0, 1000, 1), white_noise)
plt.show()
```

Moving Average

Figure 1.4.1 and Figure 1.4.2:suggest one model for introducing temporal correlation in a series.

In a **moving average** (also known as a rolling average), each value is replaced by the average of that value and a selection of its neighbors.

Simple Three-Value Moving Average

For example, a simple centered three-value moving average would define v_t as:

$$v_t \triangleq \frac{1}{3} (w_{t-1} + w_t + w_{t+1}) \quad (1.4.3)$$

where $\triangleq$ is read as "is defined as."

Even if the w_t 's are iid, the v_t 's will be serially correlated.

Example 1.4.1

Consider:

$$v_2 = \frac{1}{3}(w_1 + w_2 + w_3)$$
$$v_3 = \frac{1}{3}(w_2 + w_3 + w_4)$$

Solution

Both v_2 and v_3 depend on the values w_2 and w_3 , creating correlation between consecutive observations despite w_t being iid.

Weighted Moving Average

$$v_t \triangleq \frac{1}{10}w_{t-2} + \frac{2}{10}w_{t-1} + \frac{4}{10}w_t + \frac{2}{10}w_{t+1} + \frac{1}{10}w_{t+2} \quad (1.4.4)$$

This gives more weight to the central observation while still incorporating information from neighbors.

Random Walk

Applying a moving average to uncorrelated noise is one plausible method for introducing temporal correlation. Another scenario is when each time step starts from the value of the previous point (e.g., a stock price that opens trading at yesterday's closing price).

Basic Random Walk

This suggests a model of the form:

$$x_t = x_{t-1} + w_t \quad (1.4.5)$$

where x_t is the value of the series x at time t and the w_t 's are iid white noise.

Random Walk with Drift

If there is also an underlying trend (e.g., a stock's long-term price going up), we can add a constant drift term δ :

$$x_t = \delta + x_{t-1} + w_t \quad (1.4.6)$$

Apparent Trends

Spurious Trends

Random walks can very convincingly appear to have an underlying trend, even when none exists!

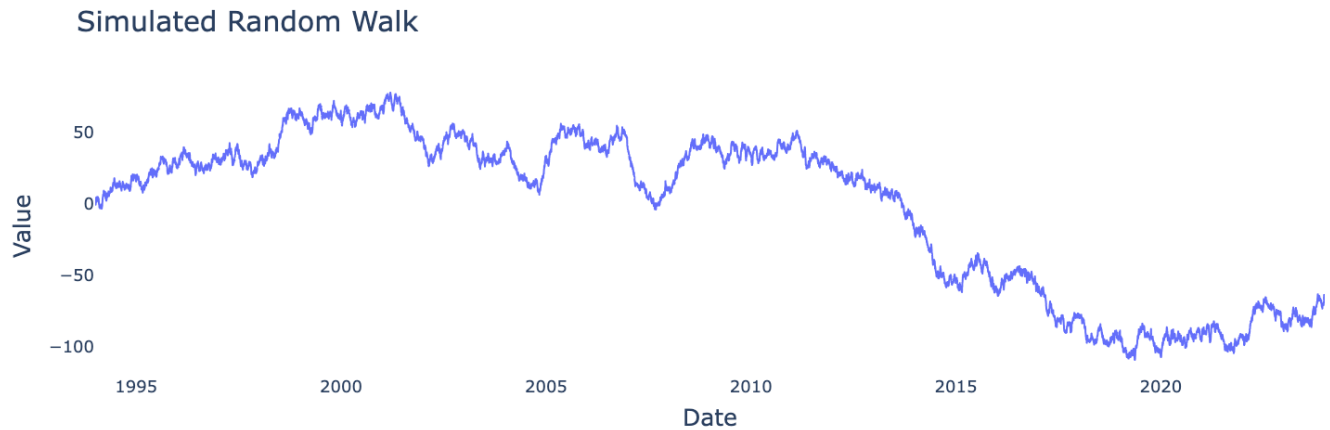


Figure 1.4.3: Simulated random walk with $w_t \sim N(0,12)$ showing apparent trends.

We can simulate a random walk starting with our white noise and using the `np.cumsum` function—if you don't understand why yet don't worry, we'll address it in later chapters. For now, this cell lets you get an idea of what random walks might look like. As before, you may need to copy and paste into a Jupyter notebook.

```
import numpy as np
import matplotlib.pyplot as plt

standard_deviation = 1.0 # Play around with this parameter!
white_noise = np.random.normal(loc=0.0, scale=standard_deviation, size=1000)
# We can also look at alternative distributions. Uncomment the following to use a Cauchy d
# white_noise = np.random.standard_cauchy(size=1000)
plt.plot(np.arange(0, 1000, 1), np.cumsum(white_noise))
plt.show()
```

We will return to random walks and their cousins extensively throughout the book.

Autoregression

The concept of a random walk can be generalized to **autoregressive processes**.

Second Order Autoregression Example

Below we present a simulated autoregressive process generated by:

$$x_t = 1.5x_{t-1} - 0.9x_{t-2} + w_t \quad (1.4.7)$$

Simulated AR(2) Process with Complex Roots

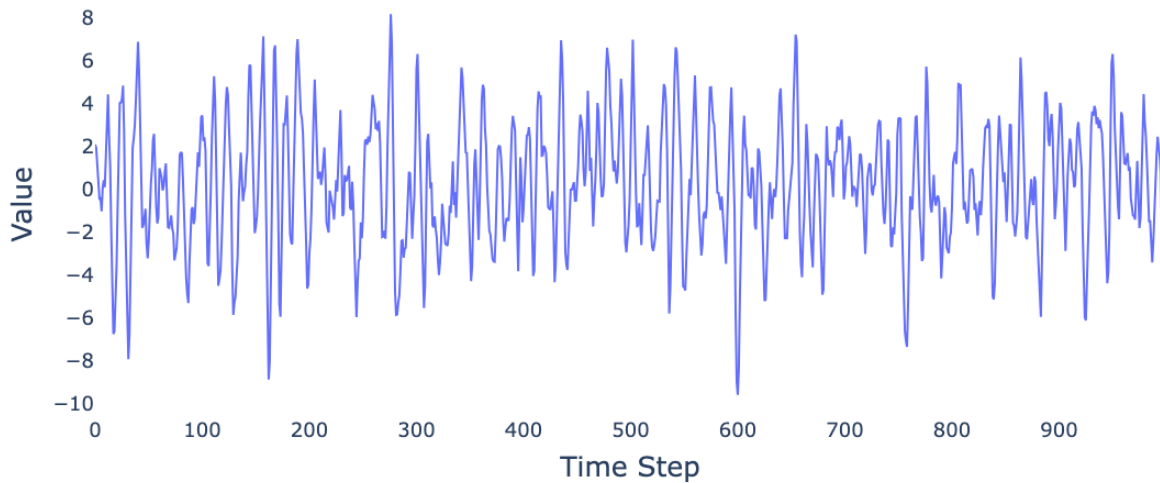


Figure 1.4.4:

Simulated AR(2) process.

Note the long-range patterns, despite the fact that the model can only “remember” two time steps into the past. Autoregressive models have surprising ability to capture complex dynamics.

Signal and Noise

In many real-life scenarios, we consider our signal to be formed by the addition of white noise to an underlying process.

Separating signal from noise is a central focus in signal processing, with applications to:

- Radar systems
- Telecommunications
- Sound engineering
- Image processing
- And many other fields

Example: Sinusoidal Signal with Noise

Signal $2\sin\left(\frac{2\pi}{20}t + \frac{3}{5}\pi\right) + w_t$ for $w_t \sim \mathcal{N}(0, 0^2)$, $\mathcal{N}(0, 1^2)$, and $\mathcal{N}(0, 5^2)$.

Figure 1.4.5: Signal $2\sin\left(\frac{2\pi}{20}t + \frac{3}{5}\pi\right) + w_t$ for $w_t \sim \mathcal{N}(0, 0^2)$, $\mathcal{N}(0, 1^2)$, and $\mathcal{N}(0, 5^2)$.

As the variance of the noise increases, it becomes progressively more difficult to recover the underlying signal. Much of time series analysis is concerned with developing methods to extract meaningful patterns from noisy observations.

Example 1.4.2

One of the best ways to become comfortable with time series and avoid being misled by apparent patterns is to simulate random series and plot the output. Use the code above to plot 10 different time series. Do they appear to exhibit a pattern? What happens if you use a Cauchy distribution instead of a Gaussian?

Footnotes

1. The notation used in this book is explained in the [appendix](#).

1.5: Additional Resources

This online textbook is designed to be as self-contained as possible. Nevertheless, it is neither possible nor desirable to have a single work cover all aspects at all levels. The resources below contain additional background material and detailed derivations.

Recommended Textbook

For those looking to deepen their understanding of the mathematical and statistical underpinnings of time series analysis, I highly recommend *Time Series Analysis and Its Applications* by Robert Shumway and David Stoffer [Shumway & Stoffer \(2025\)](#).¹ However, data scientists new to time series analysis may find Shumway and Stoffer to be too abstract and lacking in real world applications. For this reason, you may consider Shumway and Stoffer to be the next level beyond the material in this textbook.

Other Resources

There are many other valuable resources for learning time series analysis. I have found the following to be particularly useful:

1. Time Series Forecasting in Python

Author: Marco Peixeiro

In many ways the polar opposite of Shumway and Stoffer, [Time Series Forecasting in Python](#) contains multiple valuable guides to implementing time series analysis in Python but minimal statistical background.

2. Forecasting: Principles and Practice, 3rd ed.

Authors: Rob Hyndman and George Athanasopoulos

Available online: <https://otexts.com/fpp3/>, with a version incorporating Python recently published at <https://otexts.com/fpppy/>.

While the math and statistics are more basic than those covered in this textbook, it is still a valuable introduction to multiple topics, particularly exponential smoothing. The original versions used the language R, with a recent version using Python also available.

3. Prof. Steve Brunton's YouTube Lectures

Topic: Fourier Analysis

Link: <https://www.youtube.com/watch?v=jNC0jxb0OxE>

An approachable but rigorous introduction to Fourier analysis.

4. Prof. Aric LaBarr's YouTube Lessons

Topic: Time Series (5-minute lessons)

Link: <https://www.youtube.com/@AricLaBarr/playlists>

Easily understood short lessons on multiple topics in time series analysis and econometrics.

5. tsa4-python GitHub Repository

Link: <https://github.com/borisgarbuzov/tsa4-python/tree/master>

Ports the first three chapters of code from *Time Series Analysis and Its Applications, 4th ed.* from R to Python.

6. Kalman and Bayesian Filters in Python

Link: <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>

An online textbook with interactive notebooks providing intuitive explanations and demonstrations of Kalman filters.

References

1. Shumway, R. H., & Stoffer, D. S. (2025). *Time Series Analysis and Its Applications: With R Examples*. In *Springer Texts in Statistics*. Springer Nature Switzerland. [10.1007/978-3-031-70584-7](#)

2: BACKGROUND MATH

2.1: Introduction

2.2: Geometric Series

2.3: Complex Numbers

2.4: Properties of Variance and Covariance

2.5: Information Criteria

This page titled [2: Background Math](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

2.1: Introduction

This chapter contains background on geometric series, complex numbers, the properties of variance and covariance, and information criteria that will be necessary for later chapters. Feel free to skip this chapter if you're already familiar with the material and/or return when you encounter the concepts in the text.

This page titled [2.1: Introduction](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

2.2: Geometric Series

Geometric series appear frequently in time series analysis, particularly when working with ARMA models and spectral analysis. Understanding their properties is essential for many derivations in this course.

Finite Geometric Series

Let $S = x + x^2 + x^3 + \dots + x^N = \sum_{n=1}^N x^n$. This is known as a **geometric progression**.

Now consider xS :

$$xS = x^2 + x^3 + x^4 + \dots + x^{N+1} = \sum_{n=2}^{N+1} x^n \quad (2.2.1)$$

Subtracting these:

$$(1 - x)S = S - xS = (x + x^2 + x^3 + \dots + x^N) - (x^2 + x^3 + x^4 + \dots + x^{N+1}) = x - x^{N+1} = x(1 - x^N) \quad (2.2.2)$$

Therefore:

$$S = x \frac{1 - x^N}{1 - x} \quad \text{for } x \neq 1. \quad (2.2.3)$$

Summing from 0

When we instead sum from $n = 0$, the formula becomes $\frac{1 - x^{N+1}}{1 - x}$.

Exercise

Prove that $\sum_{n=0}^N x^n = \frac{1 - x^{N+1}}{1 - x}$ for $x \neq 0, 1$.

Answer available at the end of this section.

Infinite Geometric Series

Eq. 2.2.3 is exact for any $x \neq 1$. We can divide its applications into two cases:

- **Case 1:** If $|x| > 1$, the series will diverge as N approaches infinity.
- **Case 2:** For $|x| < 1$, we have $\lim_{N \rightarrow \infty} x(1 - x^N) = x$. Thus we arrive at:

$$\lim_{N \rightarrow \infty} \sum_{n=1}^N x^n = \frac{x}{1 - x} \quad \text{for } |x| < 1 \quad (2.2.4)$$

And starting from $n = 0$, $\lim_{N \rightarrow \infty} (1 - x^N) = 1$, yielding:

$$\lim_{N \rightarrow \infty} \sum_{n=0}^N x^n = \frac{1}{1 - x} \quad \text{for } |x| < 1, x \neq 0 \quad (2.2.5)$$

The value of Eq. 2.2.5 for $x = 0$ depends on our definition of 0^0 , which in different contexts may be either undefined or defined as 1.

Related Infinite Sums

We can derive several related infinite sums from the basic formula:

Alternating Series

$$\sum_{n=1}^{\infty} (-1)^n x^n = \sum_{n=1}^{\infty} (-x)^n = \frac{-x}{1 - (-x)} = \frac{-x}{1+x} \quad (2.2.6)$$

Even Powers

$$\sum_{n=1}^{\infty} x^{2n} = \sum_{n=1}^{\infty} (x^2)^n = \frac{x^2}{1-x^2} \quad (2.2.7)$$

We will come back to Eqs. 2.2.4 and 2.2.5 throughout the course of this book.

Exercise

Find the values of $\sum_{n=0}^{\infty} \frac{1}{4^n}$ and $\sum_{n=1}^{\infty} \frac{1}{4^n}$.

Answer available at the end of this section.

Answers

Exercise

This follows immediately from the derivation in Eq. 2.2.2. Let $T = 1 + x + x^2 + \dots + x^N = \sum_{n=0}^N x^n$.

$$(1-x)T = T - xT = (1 + x + x^2 + \dots + x^N) - (x + x^2 + x^3 + \dots + x^{N+1}) = 1 - x^{N+1} \quad (2.2.8)$$

Thus we arrive at

$$T = \frac{1 - x^{N+1}}{1 - x} \quad (2.2.9)$$

Exercise

$$\sum_{n=0}^{\infty} \frac{1}{4^n} = \frac{1}{1 - \frac{1}{4}} = \frac{1}{\frac{3}{4}} = \frac{4}{3} \quad (2.2.10)$$

$$\sum_{n=1}^{\infty} \frac{1}{4^n} = \frac{\frac{1}{4}}{1 - \frac{1}{4}} = \frac{\frac{1}{4}}{\frac{3}{4}} = \frac{1}{3} \quad (2.2.11)$$

Note that summing from 1 instead of 0 reduces the sum by exactly 1 as it removes the term $x^0 = 1$.

2.3: Complex Numbers

Definition of Complex Numbers

Imaginary Components

Complex numbers expand the real numbers by adding the *imaginary* square root of -1

$$i \triangleq \sqrt{-1}, \quad (2.3.1)$$

thus $\sqrt{-9} = 3i$, $\sqrt{-2} = \sqrt{2}i$, etc. ...{note}

i vs. *j*

In some fields, in particular electrical engineering, the square root of -1 is denoted by j instead of i . This is also the convention used in NumPy.

Complex: Real plus Imaginary

Speaking more broadly, a *complex* number has both real and imaginary components. A complex number z can be expressed as

$$z = a + bi \quad (2.3.2)$$

with a being the real component and b the imaginary. Of course, we can treat any purely real or imaginary number as a complex number by setting y or x to zero, respectively.

Complex Conjugate

A key property of complex number is the complex conjugate z^* defined as

$$z^* = (a + bi)^* = a - bi \quad (2.3.3)$$

i.e. the complex conjugate leaves the real components unchanged and flips the sign of the imaginary components. From this it immediately follows that any purely real number is its own complex conjugate.

Magnitude of Complex Numbers

The magnitude of a complex number, denoted as $|z|$, is given as

$$|z| = |a + bi| = \sqrt{(a + bi)(a + bi)^*} = \sqrt{(a + bi)(a - bi)} = \sqrt{a^2 - abi + abi + b^2} = \sqrt{a^2 + b^2} \quad (2.3.4)$$

where we have used the fact that $-b^2i^2 = b^2$.

Eq. 2.3.4 suggests that complex numbers may be viewed as vectors in the real plane with length $|z|$.

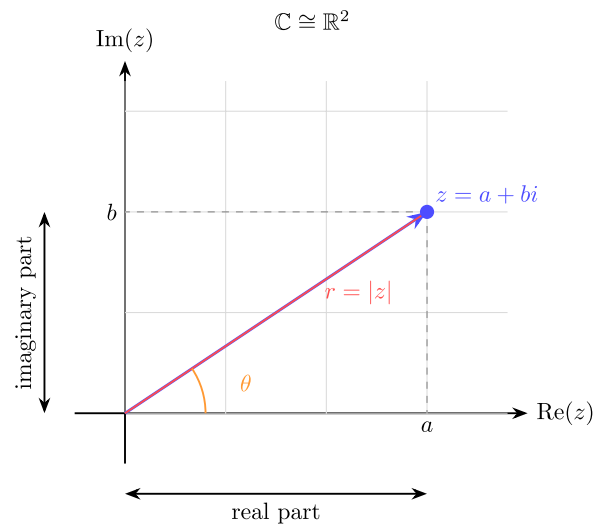


Figure 2.3.1: Conversion between polar and Cartesian forms of complex numbers.

You may recognize Figure 2.3.1 as an example of an *isomorphism* between $\mathbb{C}$ and $\mathbb{R}^2$. If you're not familiar with the term isomorphism, you can consider it a fancy way of saying "mapping."

Polar Form

Figure 2.3.1 suggests the *polar form* of complex with magnitude $r = |z|$ and angle $\theta = \arctan(\frac{y}{x})$, allowing the substitution

$$z = r \cos(\theta) + ir \sin(\theta). \quad (2.3.5)$$

While Eq. 2.3.5 may be suggestive as a heuristic, we will see in the next section that it actually touches upon a deep concept in mathematics.

Euler's Formula: The "Jewel" of Mathematics

Euler's formula, called "our jewel" by Richard Feynman, establishes a profound connection between trigonometry and complex numbers.

Euler's Formula:

$$e^{i\theta} = \cos(\theta) + i \sin(\theta) \quad (2.3.6)$$

Or, multiplying through by r for magnitudes different than 1:

$$re^{i\theta} = r \cos(\theta) + ir \sin(\theta) \quad (2.3.7)$$

Euler's Identity

When $\theta = \pi$, we have:

$$e^{i\pi} = -1 + i \cdot 0 \quad (2.3.8)$$

or

$$e^{i\pi} + 1 = 0 \quad (2.3.9)$$

This is known as **Euler's identity**, which relates five fundamental mathematical constants: e , i , π , 1, and 0.

Proof of Euler's Formula

Let $f(\theta) = \frac{\cos(\theta) + i \sin(\theta)}{e^{i\theta}}$

$$\begin{aligned}
 \frac{df(\theta)}{d\theta} &= \frac{d}{d\theta} \frac{\cos(\theta) + i \sin(\theta)}{e^{i\theta}} \\
 &= \frac{d}{d\theta} e^{-i\theta} (\cos(\theta) + i \sin(\theta)) \\
 &= -ie^{-i\theta} (\cos(\theta) + i \sin(\theta)) + e^{-i\theta} (-\sin(\theta) + i \cos(\theta)) \\
 &= e^{-i\theta} (-i \cos(\theta) + \sin(\theta) - \sin(\theta) + i \cos(\theta)) \\
 &= e^{-i\theta} (0) \\
 &= 0
 \end{aligned}
 \tag{2.3.10}$$

Since $\frac{df}{d\theta} = 0$, $f(\theta)$ is a constant, and we need only establish its value at a single point.

At $\theta = 0$:

$$f(0) = \frac{1 + 0i}{1} = 1 \tag{2.3.11}$$

Therefore $f(\theta) = 1$ for all θ , or:

$$e^{i\theta} = \cos(\theta) + i \sin(\theta) \tag{2.3.12}$$

Note

Proof via Taylor Series This formula can also be proven by replacing the exponential and trigonometric functions with their Taylor series expansions and grouping the real and complex terms.

Exercise

Express $\cos(\theta)$ purely in terms of complex exponentials.

Answer available at the end of this section.

Applications of Euler's Formula

Solving Problems Involving i

Euler's formula can help solve otherwise extremely difficult problems involving i . To see this, note that given that $e^{i\theta} = \cos(\theta) + i \sin(\theta)$, at $\theta = \frac{\pi}{2}$ we have

$$e^{\frac{i\pi}{2}} = \cos\left(\frac{\pi}{2}\right) + i \sin\left(\frac{\pi}{2}\right) = 0 + 1i = i \tag{2.3.13}$$

Solving i^i

Eq. 2.3.13 allows us to solve problems that at first may seem intractable. For example, let us calculate i^i :

$$i^i = \left(e^{\frac{i\pi}{2}}\right)^i = e^{\frac{i^2\pi}{2}} = e^{-\frac{\pi}{2}}. \quad (2.3.14)$$

Somewhat surprisingly, it turns out that i^i is a real number slightly greater than $\frac{1}{5}$.

Exercise

We have seen that generating the square root of -1 requires expanding beyond the real numbers. Does calculating $\sqrt{i}$ require expansion beyond the complex numbers, or is contained in the complex numbers?

Answer available at the end of this section.

Sine and Cosine Addition Formulae

Use of Euler's formula can simplify trigonometric derivations by transforming geometric problems into algebraic ones. A pair of trigonometric identities that we will make use of later in the book are the *sine and cosine addition formulae*, given as

$$\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \quad (2.3.15)$$

and

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta). \quad (2.3.16)$$

While these can be proven purely geometrically, Euler's formula gives us an arguably far simpler algebraic method to prove them instead. Using $\Re(z)$ to denote the real component of z and $\Im(z)$ to denote the imaginary component, we have

$$\cos(\alpha + \beta) = \Re(e^{i(\alpha+\beta)}) \quad (2.3.17)$$

and

$$\sin(\alpha + \beta) = \Im(e^{i(\alpha+\beta)}). \quad (2.3.18)$$

Note that

$$\begin{aligned} e^{i(\alpha+\beta)} &= e^{i\alpha}e^{i\beta} \\ &= (\cos(\alpha) + i\sin(\alpha))(\cos(\beta) + i\sin(\beta)) \\ &= \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ &\quad + i\sin(\alpha)\cos(\beta) + i\cos(\alpha)\sin(\beta) \\ &= [\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)] \\ &\quad + i[\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)]. \end{aligned} \quad (2.3.19)$$

Combining Eq. 2.3.22 with Eq. 2.3.23 we arrive at the sine addition formula

$$\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta). \quad (2.3.20)$$

Similarly, combining Eq. 2.3.21 with Eq. 2.3.23 gives us the cosine addition formula

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta). \quad (2.3.21)$$

Answers

Exercise

First note that sine is an odd function while cosine is even. Thus,

$$\cos(-\theta) + i \sin(-\theta) = \cos(\theta) - i \sin(\theta). \quad (2.3.22)$$

Adding Eqs 2.3.6 and 2.3.13 gives us:

$$e^{i\theta} + e^{-i\theta} = [\cos(\theta) + i \sin(\theta)] + [\cos(\theta) - i \sin(\theta)] = 2 \cos(\theta). \quad (2.3.23)$$

We conclude that

$$\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta}). \quad (2.3.24)$$

Exercise

We can solve this using Eq. 2.3.13:

$$\sqrt{i} = \sqrt{e^{\frac{i\pi}{2}}} = (e^{\frac{i\pi}{2}})^{\frac{1}{2}} = e^{\frac{i\pi}{4}} = \cos\left(\frac{\pi}{4}\right) + i \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}}(1 + i). \quad (2.3.25)$$

In more advanced math texts this is explained as stemming from the *algebraic closure* of the complex numbers. Put simply, any meaningful formula involving complex numbers has a solution that is also a complex number.

2.4: Properties of Variance and Covariance

We will make heavy use of both variance and covariance (in particular the *autocovariance*) throughout the book. This chapter presents a refresher on these topics laying the foundation for the forms used in time series analysis discussed in the next chapter.

Operators

There are several *operators* we will encounter in this book. Values such as mean, variance, and covariance can all be cast in the operator formalism. In later chapters we will introduce the backshift and Fourier operators. So what is an operator?

An operator $\mathbb{O}$ is defined as a rule that maps a member of a set to another member. Thus, an operator could just be a function such as 5, defined as multiplying by 5 (i.e. $f(x) = 5, x)^{[1]}$. However, the most common operators we will use map one function to another function. Two such operators are differentiation $\frac{d}{dx}$ and integration $\int dx$.

1: In some disciplines such as quantum mechanics, operators are often denoted by a "hat" as $\hat{\mathbb{O}}$, not to be confused with the hat used to denote a statistical estimator.

Linear operators are of particular interest. An operator is a linear operator if it fulfills the following two conditions:

1. $\mathbb{O}, aF(x) = a, \mathbb{O}, F(x)$ for any constant a
2. $\mathbb{O}, (F(x) + G(y)) = \mathbb{O}, F(x) + \mathbb{O}, G(y)$

We may write the two conditions more succinctly as

$$\mathbb{O}, (aF(x) + bG(y)) = a, \mathbb{O}, F(x) + b, \mathbb{O}, G(y) \quad (2.4.1)$$

for any constants a and b .

Going forward, we will assume that all operators discussed are linear unless otherwise noted.

Exercise

Demonstrate that square root is *not* a linear operator.

Answer available at the end of this section.

Exercise

Demonstrate that differentiation is a linear operator.

Answer available at the end of this section.

Expectation

The most important operator in statistics and data science is the *expectation* operator $\mathbb{E}[F(x)]$, usually first encountered in the context of the arithmetic mean.

Expectation Notation

We will use the notation $\mathbb{E}$ to distinguish the expectation operator from a function..

Expectation is defined as:

$$\mathbb{E}[F(x)] \triangleq \begin{cases} \sum_x F(x)P(x) & \text{discrete } x \\ \int F(x)P(x), dx & \text{continuous } x \end{cases} \quad (2.4.2)$$

where $P(x)$ is either the probability function of the random variable x such that

$$\mathbb{P}(a < X < b) = \sum_{a < x_i < b} P(x_i) \quad (2.4.3)$$

(discrete x) or the probability density function such that

$$\mathbb{P}(a < X < b) = \int_a^b P(x), dx \quad (2.4.4)$$

(continuous x).

In general, we will not explicitly reference both the discrete and continuous cases in this book. Instead, we will use the notation $\mathbb{E}$ or one of the two methods in Eq. 2.4.2 with an understanding that a reference to either one implicitly refers to both unless specified otherwise.

Linearity of Expectation

An important property of expectation is that it is a linear operator. We can demonstrate this fact by proving that $\mathbb{E}[a, F(x) + b, G(y)] = a, \mathbb{E}[F(x)] + b, \mathbb{E}[G(y)]$ as follows:

$$\begin{aligned} \mathbb{E}[aF(x) + bG(y)] &= \iint (aF(x) + bG(y))P(x, y)dx dy \\ &= \iint aF(x)P(x, y)dx dy + \iint bG(y)P(x, y)dx dy \\ &= a \iint F(x)P(x, y)dx dy + b \iint G(y)P(x, y)dx dy \\ &= a \int F(x)P(x)dx + b \int G(y)P(y)dy \\ &= a\mathbb{E}[F(x)] + b\mathbb{E}[G(y)] \end{aligned} \quad (2.4.5)$$

where in the fourth line we have used the definition of marginal probabilities: $P(x) = \int P(x, y), dy$ and $P(y) = \int P(x, y), dx$.

Functions of Different Variable

In most scenarios both F and G will be functions of the same variable. We have presented them as functions of different variables for the sake of proving linearity for the more general case of different variables.

Exercise

Why can we simplify $\int \int F(x), P(x, y), dx, dy$ to $\int F(x), P(x), dx$? What property of marginal probabilities justifies this step?

Answer available at the end of this section.

Moments

We have not yet specified the identity of $F(x)$. By far, the most commonly used functions are powers of x , i.e.

$$F(x) = x^n, \quad n = 0, 1, 2, \dots \quad (2.4.6)$$

The expectation value of Eq. 2.4.6 is the n th *moment* of $P(x)$:

$$\mathbb{E}[x^n] = \int x^n P(x), dx, \quad (2.4.7)$$

(moment-def) where we are usually interested in cases of $n = 1 - 4$ [2]:

1. The first moment is the arithmetic mean denoted as μ_x .
2. The second moment relates to the variance, or how widely spread the distribution is.
3. The third moment relates to the skew, or how symmetric the function is.
4. The fourth moment relates to the kurtosis, or the "fatness" of the tails.

Exercise

Why do we not consider the case of $n = 0$?

Answer available at the end of this section.

Finite Moments

An important theorem states that if the k th moment $\mathbb{E}[x^k]$ is finite, then all moments $j < k$ must also be finite. As a corollary, if $\mathbb{E}[x^k]$ is infinite, all moments $m > k$ must also be infinite.

Proof: Let $\mathbb{E}[x^k]$ be finite and $j < k$ ($\forall$ is read as "for all")

$$\begin{aligned} \mathbb{E}[x^j] &= \int_{-\infty}^{\infty} x^j P(x) dx \\ &= \int_{-\infty}^{-1} x^j P(x) dx + \int_{-1}^1 x^j P(x) dx + \int_1^{\infty} x^j P(x) dx \\ &\text{note that } |x^j| \leq |x^k| \forall |x| \geq 1 \text{ and } |x^j| \leq 1 \forall |x| \leq 1 \\ &\leq \left| \int_{-\infty}^{-1} x^k P(x) dx \right| + \int_{-1}^1 1 P(x) dx + \int_1^{\infty} x^k P(x) dx \\ &\leq \left| \int_{-\infty}^{-1} x^k P(x) dx \right| + 1 + \int_1^{\infty} x^k P(x) dx \\ &< \infty \end{aligned}$$

where we have used the fact that $\int_{-1}^1 1 P(x), dx \leq 1$, with equality only occurring if the entire probability mass is contained in the interval $[-1, 1]$.

Exercise

Consider the standard Cauchy distribution (also known as the Cauchy-Lorentz or Lorentz distribution) defined as

$$P(x) = \frac{1}{\pi} \cdot \frac{1}{1 + x^2} \quad (2.4.8)$$

- a. Prove that this is a valid probability distribution.
- b. What is the first moment of the Cauchy distribution?

c. What does part b tell you about higher moments?

Answer available at the end of this section.

Variance

The variance of a random variable X is defined as

$$\mathbb{V}(X) \triangleq \mathbb{E}[(X - \mu_x)^2], \quad (2.4.9)$$

and is often denoted as σ_x^2 . The variance gives us a measure of how widely the distribution is spread about the mean. In practice, we more commonly make use of the *standard deviation* σ_x , which is simply the square root of the variance.

As written, the variance is slightly different than our definition of the second moment.

Central Moment

The quantity $\mathbb{E}[(X - \mu_x)^n]$ is referred to as the *central moment*

By exploiting the linearity of expectation we can express Eq. 2.4.9 using the first and second moments exclusively:

$$\begin{aligned} \mathbb{E}[(X - \mu_x)^2] &= \mathbb{E}[X^2 - 2X\mu_x + \mu_x^2] \\ &= \mathbb{E}[X^2] - \mathbb{E}[2X\mu_x] + \mathbb{E}[\mu_x^2] \\ &= \mathbb{E}[X^2] - 2\mu_x\mathbb{E}[X] + \mu_x^2\mathbb{E}[1] \\ &= \mathbb{E}[X^2] - 2\mu_x^2 + \mu_x^2 \\ &= \mathbb{E}[X^2] - \mu_x^2 \\ &= \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \end{aligned} \quad (18)$$

Covariance

The covariance of two random variables X, Y is defined as

$$\text{Cov}(X, Y) \triangleq \mathbb{E}[(X - \mu_x)(Y - \mu_y)]. \quad (2.4.11)$$

Covariance is also written as $\sigma_{x,y}$. Note that $\text{Cov}(X, X) = \mathbb{V}(X)$.

Unlike variance, which is never negative, covariance can be negative, zero, or positive. Following the same logic used in Eq. 2.4.18, we can also express the covariance as

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \quad (2.4.12)$$

Exercise

Why can variance never be negative?

Answer available at the end of this section.

Covariance gives us a measure of how much random variables change in tandem with one another.

Variance of Sums of Random Variables

One of the most fundamental aspects of time series analysis is understanding the variance of the sum of random variables. Let us begin with the variance of a sum of two random variables, X and Y .

Variance of Multiple Random Variables

$$\begin{aligned}
 \mathbb{V}(X + Y) &= \mathbb{E}[(X + Y - \mu_x - \mu_y)^2] \\
 &= \mathbb{E}[X^2] + \mathbb{E}[Y^2] + 2\mathbb{E}[XY] - 2\mathbb{E}[X\mu_y] \\
 &\quad - 2\mathbb{E}[X\mu_y] - 2\mathbb{E}[Y\mu_x] - 2\mathbb{E}[Y\mu_y] + 2\mathbb{E}[\mu_x\mu_y] \\
 &\quad + \mathbb{E}[\mu_x^2] + \mathbb{E}[\mu_y^2] \\
 &= \mathbb{E}[X^2] + \mathbb{E}[Y^2] + 2\mathbb{E}[XY] - \mu_x^2 - \mu_y^2 - 2\mu_x\mu_y \\
 &= (\mathbb{E}[X^2] - \mu_x^2) + (\mathbb{E}[Y^2] - \mu_y^2) + 2(\mathbb{E}[XY] - \mu_x\mu_y) \\
 &= \mathbb{V}(X) + \mathbb{V}(Y) + 2\text{Cov}(X, Y)
 \end{aligned} \tag{2.4.13}$$

The last line of Eq. 2.4.21 can be rewritten as

$$\text{Cov}(X, X) + \text{Cov}(Y, Y) + \text{Cov}(X, Y) + \text{Cov}(Y, X), \tag{2.4.14}$$

or, renaming the variables as X_0 and X_1

$$\sum_{i,j=0}^1 \text{Cov}(X_i, X_j). \tag{2.4.15}$$

The above suggests (though does not prove) that the variance of a sum of variables is the sum of all covariance combinations. For random variables $X_0, X_1, X_2, \dots, X_{n-1}$:

$$\mathbb{V}\left(\sum_{i=0}^{n-1} X_i\right) = \sum_{i,j=0}^{n-1} \text{Cov}(X_i, X_j) = \sum_{i=0}^{n-1} \mathbb{V}(X_i) + 2 \sum_{i=0, j>i}^{n-1} \text{Cov}(X_i, X_j) \tag{2.4.16}$$

where the last term sums i to $n-2$ and j to $n-1$. Eq. 2.4.16 can be proven in the same manner as Eq. 2.4.24, though the algebra gets rather intricate. We present a more direct proof in the following problem.

Exercise

Prove Eq. 2.4.16 using the definition of variance and covariance. Hint, express $\mathbb{V}(X)$ as $\text{Cov}(X, X)$.

Answer available at the end of this section.

Variance of Independent Variables

From Eq. 2.4.16 we can see that *if all variables have zero covariance* (most commonly due to independence), the variance of a sum of variables is the sum of the variances of each variable

$$\mathbb{V}\left(\sum_{i=0}^{n-1} X_i\right) = \sum_{i=0}^{n-1} \mathbb{V}(X_i) \quad \text{for zero covariance.} \quad (2.4.17)$$

While it is very tempting to simply assume that Eq. 2.4.17 holds, in real life we must justify its use, either from theoretical analysis and/or empirical evidence.

It should also be recalled that knowing that random variables have zero covariance does not inherently prove independence. As a simple counterexample, consider a random variable X with zero mean and zero third moment and let $Y = X^2$. An example might be $X \sim N(0, 1)$ and $Y = X^2 \sim \chi_1^2$. Clearly, X and Y are highly dependent; for example, knowing that $Y > 4$ tells us $|X| > 2$. Nevertheless, they still have zero covariance:

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = \mathbb{E}[XY] - 0 \cdot \mathbb{E}[Y] = \mathbb{E}[X^3] = 0$$

Variance-Covariance Inequality

Warning: The math in this section can get rather heavy. Feel free to skip this section if you're having difficulty. While the material below does add to the overall understanding of future material, it is not absolutely necessary.

Arithmetic Inequality

From Eq. 2.4.21 we see that

$$\mathbb{V}(X + Y) = \mathbb{V}(X) + \mathbb{V}(Y) + 2, \text{Cov}(X, Y). \quad (2.4.18)$$

By substituting in $-Y$, we arrive at

$$\mathbb{V}(X - Y) = \mathbb{V}(X) + \mathbb{V}(Y) - 2, \text{Cov}(X, Y). \quad (2.4.19)$$

As variances are by definition non-negative, by combining the above two equations we arrive at the inequality $2, |\text{Cov}(X, Y)| \leq \mathbb{V}(X) + \mathbb{V}(Y)$. We can express this in terms of the arithmetic mean of the variances:

$$|\text{Cov}(X, Y)| \leq \frac{1}{2}(\mathbb{V}(X) + \mathbb{V}(Y)) \quad (2.4.20)$$

While Eq. 2.4.20 is true, we will soon see that we can get an even tighter bound on the inequality.

Cauchy-Schwarz Inequality

In order to better understand the relation between variance and covariance, we must first define the Cauchy-Schwarz inequality, a valuable inequality from linear algebra. In words, it states that the square of the inner product of two vectors must always be less than or equal to the norm of the first vector squared times the second vector's norm squared.

$$(\mathbf{u} \cdot \mathbf{v})^2 \leq |\mathbf{u}|^2 |\mathbf{v}|^2 \quad (2.4.21)$$

or, recognizing that vector norms are always non-negative

$$|\mathbf{u} \cdot \mathbf{v}| \leq |\mathbf{u}| |\mathbf{v}| \quad (2.4.22)$$

Eqs 2.4.29 and 2.4.30 will be equalities if and only if $\mathbf{u}$ and $\mathbf{v}$ can be expressed as scalar multiples of one another (i.e. lie on the same line).

If $\mathbf{u}$ and/or $\mathbf{v}$ is the zero vector, the inequality is trivially true, let us prove the inequality when neither is:

$$\text{Let } \mathbf{w} \triangleq \frac{\mathbf{u}}{|\mathbf{u}|} \pm \frac{\mathbf{v}}{|\mathbf{v}|}$$

$$\begin{aligned}
 \text{Let } \mathbf{w} &\triangleq \frac{\mathbf{u}}{\|\mathbf{u}\|} \pm \frac{\mathbf{v}}{\|\mathbf{v}\|} \\
 \mathbf{w} \cdot \mathbf{w} &= \left(\frac{\mathbf{u}}{\|\mathbf{u}\|} \pm \frac{\mathbf{v}}{\|\mathbf{v}\|} \right) \cdot \left(\frac{\mathbf{u}}{\|\mathbf{u}\|} \pm \frac{\mathbf{v}}{\|\mathbf{v}\|} \right) \\
 &= \frac{\mathbf{u} \cdot \mathbf{u}}{\|\mathbf{u}\|^2} \pm 2 \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} + \frac{\mathbf{v} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \\
 &= 1 \pm 2 \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} + 1 \\
 &= 2 \pm 2 \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}
 \end{aligned}$$

as with any inner product $\mathbf{w} \cdot \mathbf{w} \geq 0$

$$\begin{aligned}
 0 &\leq 2 \pm 2 \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \\
 1 &\geq \frac{|\mathbf{u} \cdot \mathbf{v}|}{\|\mathbf{u}\| \|\mathbf{v}\|} \\
 \|\mathbf{u}\| \|\mathbf{v}\| &\geq |\mathbf{u} \cdot \mathbf{v}|
 \end{aligned}$$

Functions as Infinite Dimensional Vectors

The Cauchy-Schwarz inequality may be extended to integrals by viewing functions as infinite dimensional vectors living in Hilbert space. Let us imagine we have two continuous functions, $f(x)$ and $g(x)$ that are square-integrable on the interval $[a, b]$. Let us create n -dimensional vectors by sampling the value of the function at n evenly spaced points along the interval, producing the vectors

$$(f(x_1), f(x_2), \dots, f(x_n) = f(b)), \quad (2.4.23)$$

and

$$(g(x_1), g(x_2), \dots, g(x_n) = g(b)) \quad (2.4.24)$$

where $f(x_0) = f(a)$ and $g(x_0) = g(a)$ are omitted since we are using a right Riemann sum.

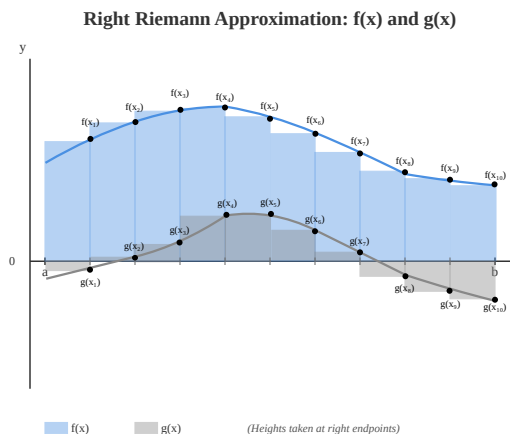


Figure 2.4.1 : Right Riemann approximation to $f(x)$ and $g(x)$ for $n = 10$.

By the Cauchy-Schwarz inequality:

$$\left(\sum_{i=1}^n f(x_i)g(x_i) \right)^2 \leq \left(\sum_{i=1}^n f(x_i)^2 \right) \left(\sum_{i=1}^n g(x_i)^2 \right) \quad (2.4.25)$$

As the term $\left[\frac{b-a}{n} \right]^2$ is a positive constant for any fixed a, b , and n , we may freely multiply both sides by it:

$$\left[\frac{b-a}{n} \right]^2 \left(\sum_{i=1}^n f(x_i) g(x_i) \right)^2 \leq \left[\frac{b-a}{n} \right]^2 \left(\sum_{i=1}^n f(x_i)^2 \right) \left(\sum_{i=1}^n g(x_i)^2 \right)$$

$$\left(\sum_{i=1}^n f(x_i) g(x_i) \left[\frac{b-a}{n} \right] \right)^2 \leq \left(\sum_{i=1}^n f(x_i)^2 \left[\frac{b-a}{n} \right] \right) \left(\sum_{i=1}^n g(x_i)^2 \left[\frac{b-a}{n} \right] \right).$$

Defining Δx as $\frac{b-a}{n}$:

$$\left(\sum_{i=1}^n f(x_i) g(x_i) \Delta x \right)^2 \leq \left(\sum_{i=1}^n f(x_i)^2 \Delta x \right) \left(\sum_{i=1}^n g(x_i)^2 \Delta x \right).$$

Finally, by taking the limit $\lim_{n \rightarrow \infty} \Delta x$ we arrive at the Cauchy-Schwarz inequality in integral form:

$$\left(\int_a^b f(x) g(x) dx \right)^2 \leq \int_a^b f(x)^2 dx \int_a^b g(x)^2 dx \quad (2.4.26)$$

Geometric Inequality

We can use the Cauchy-Schwarz inequality to derive a tighter upper bound on the covariance of two variables than that found in Eq. 2.4.20. We will only explicitly prove the bound for the discrete case of sample covariance, but by Eq. 2.4.26 the bound will also hold for the continuous case. Let

$$\mathbf{x}' \triangleq \frac{1}{\sqrt{n-1}} (\mathbf{x} - \mu_x) \quad (2.4.27)$$

for random variable realizations $\mathbf{x} = (x_1, x_2, \dots, x_n)$, and let $\mathbf{y}'$ be defined analogously. By the definitions of (sample) variance and covariance,

$$\mathbb{V}(X) = |\mathbf{x}'|^2 = \mathbf{x}' \cdot \mathbf{x}', \quad (2.4.28)$$

$$\mathbb{V}(Y) = |\mathbf{y}'|^2 = \mathbf{y}' \cdot \mathbf{y}', \quad (2.4.29)$$

and

$$(\text{Cov}(X, Y))^2 = (\mathbf{x}' \cdot \mathbf{y}')^2. \quad (2.4.30)$$

Substituting these definitions into Eq. 2.4.21, we conclude that

$$(\mathbf{x}' \cdot \mathbf{y}')^2 \leq |\mathbf{x}'|^2 |\mathbf{y}'|^2, \quad (2.4.31)$$

or

$$(\text{Cov}(X, Y))^2 \leq \mathbb{V}(X) \mathbb{V}(Y) \quad (2.4.32)$$

Equivalently, by taking the square roots of both sides and recognizing that covariance could be negative, we arrive at

$$|\text{Cov}(X, Y)| \leq \sqrt{\mathbb{V}(X) \mathbb{V}(Y)}, \quad (2.4.33)$$

or more concisely

$$|\sigma_{x,y}| \leq \sigma_x \sigma_y. \quad (2.4.34)$$

We thus arrive at the tighter bound that the absolute value of the covariance must always be less than or equal to the *geometric* mean of the variances.

There is actually an even stricter requirement for covariance, namely that the covariance matrix be *positive semidefinite*. We will defer discussion of positive semidefiniteness until we encounter it in the specific [time series application of covariance](#).

Correlation

Drawbacks to Covariance

While generally useful, the covariance does have potential drawbacks. In particular, $\text{Cov}(X, Y)$ has units of $X \times Y$, causing two major difficulties:

1. The exact value obtained will depend on the units used, for example the covariance of height and weight will be different in imperial vs. metric.
2. It can be very difficult to assess if the value of the covariance is practically significant, e.g. is a covariance of 3 kg cm high or low?

Correlation Definition

To overcome these difficulties with covariance, it is often convenient to instead normalize covariance to a unitless quantity bounded by $[-1, 1]$:

$$\text{Cor}(X, Y) \triangleq \frac{\sigma_{x,y}}{\sigma_x \sigma_y} \quad (2.4.35)$$

This quantity is referred to as the *correlation*. Note that the Cauchy-Schwarz inequality as applied in Eq. 2.4.34 guarantees that Eq. 2.4.35 will always lie in $[-1, 1]$. Of course, even when using correlation determining if a value such as 0.7 should be considered a high correlation or not will be context and situation dependent.

Footnotes

[1]: In some disciplines such as quantum mechanics, operators are often denoted by a "hat" as $\hat{O}$, not to be confused with the hat used to denote a statistical estimator

[2]: Some texts define moments as $\int (x - c)^n P(x), dx$ for some constant c . We will assume $c = 0$ unless otherwise specified.

Answers

Exercise

Square root violates both of the conditions listed above, i.e. in general $\sqrt{a \cdot F(x)} \neq a \cdot \sqrt{F(x)}$ and $\sqrt{F(x) + G(y)} \neq \sqrt{F(x)} + \sqrt{G(y)}$.

Exercise

We can demonstrate this by showing that Eq. 2.4.1 holds. If x and y are different variables, then $\frac{d}{dx} b, G(y) = 0$ and linearity is trivial. If both F and G are functions of x , we have:

$$\frac{d}{dx}(a \cdot F(x) + b \cdot G(x)) = \frac{d}{dx} a \cdot F(x) + \frac{d}{dx} b \cdot G(x) = a \cdot \frac{d}{dx} F(x) + b \cdot \frac{d}{dx} G(x) \quad (2.4.36)$$

Exercise

Since $F(x)$ doesn't depend on y , we can factor it out of the inner integral:

$$\int \int F(x), P(x, y), dx, dy = \int F(x) \left(\int P(x, y), dy \right) dx \quad (2.4.37)$$

The inner integral $\int P(x, y), dy$ is the definition of the marginal probability $P(x)$ in that it sums the joint probability over all possible values of y (which must sum to 1) to obtain the probability of x alone. Therefore:

$$\int F(x) \left(\int P(x, y), dy \right) dx = \int F(x), P(x), dx \quad (2.4.38)$$

Exercise

The zeroth moment is simply $\int P(x), dx$, which must equal 1 for any correctly normalized probability distribution. The only scenario in which we'd need to calculate the zeroth moment is when we do not know the correct normalization constant for our probability distribution.

Exercise

a. In order to be a valid probability distribution, the Cauchy distribution must satisfy the three Kolmogorov axioms of probability:

1. Non-negativity, i.e. $\mathbb{P}(x) \geq 0, \forall x$: This is satisfied because $\frac{1}{1+x^2}$ is strictly positive for all real x .
2. σ -additivity, i.e. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$ for $A \cap B = \emptyset$: For two disjoint intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$, we have

$$\mathbb{P}(A \cup B) = \int_{a_1}^{a_2} P(x), dx + \int_{b_1}^{b_2} P(x), dx = \mathbb{P}(A) + \mathbb{P}(B). \quad (2.4.39)$$

In texts on mathematical statistics you will find that the above point actually has additional nuance stemming from the definition of Lebesgue integrals and other measure-theoretic arguments. That said, the proof used here is fully sufficient for our purposes.

3. Summation to unity, i.e. $\int_{-\infty}^{\infty} P(x), dx = 1$.

$$\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{1}{1+x^2}, dx = \frac{1}{\pi} \arctan(x) \Big|_{x=-\infty}^{\infty} = \frac{1}{\pi} \left(\lim_{x \rightarrow \infty} \arctan(x) - \lim_{x \rightarrow -\infty} \arctan(x) \right) = \frac{1}{\pi} \left(\frac{\pi}{2} - \left(-\frac{\pi}{2}\right) \right) = \frac{1}{\pi} \pi = 1. \quad (2.4.40)$$

b. We must evaluate the integral $\mathbb{E}[x]$ for the Cauchy distribution:

$$\mathbb{E}[x] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x}{1+x^2}, dx \quad (2.4.41)$$

Letting $u = 1 + x^2$ so that $du = 2x, dx$, we have

$$\frac{1}{\pi} \int \frac{x}{1+x^2}, dx = \frac{1}{2\pi} \int \frac{du}{u} = \frac{1}{2\pi} \ln(u). \quad (2.4.42)$$

Substituting the definition of u and limits of integration

$$\frac{1}{2\pi} \ln(1+x^2) \Big|_{x=-\infty}^{\infty} = \frac{1}{2\pi} \left(\lim_{x \rightarrow \infty} \ln(1+x^2) - \lim_{x \rightarrow -\infty} \ln(1+x^2) \right) = \frac{1}{2\pi} (\infty - \infty) = \infty. \quad (2.4.43)$$

Note that strictly $\infty - \infty$ is an indeterminate form. For our purposes the relevant point is that it is not finite.

c. The fact that the first moment (the arithmetic mean) is not finite guarantees that all higher moments will also not be finite. Thus, the distribution cannot have a defined variance, skew, etc. We will see later in the book that this causes real world problems as many extreme financial events roughly obey a Cauchy distribution.

Exercise

Variance is defined as $\mathbb{E}[(X - \mu_x)^2] = \int (X - \mu_x)^2, P(x), dx$. By definition, $P(x)$ can never be negative for any x , while $(X - \mu_x)^2$ is the square of a real number and also cannot be negative. Thus, all terms in the integral are non-negative.

Exercise

Click to reveal solution **Solution:**

$$\begin{aligned} \mathbb{V} \left(\sum_{i=0}^{n-1} X_i \right) &= \text{Cov} \left(\sum_{i=0}^{n-1} X_i, \sum_{j=0}^{n-1} X_j \right) \\ &= \mathbb{E}[(X_0 - \mu_0)(X_0 - \mu_0) + (X_0 - \mu_0)(X_1 - \mu_1) \\ &\quad + (X_1 - \mu_1)(X_0 - \mu_0) + \dots + (X_{n-1} - \mu_{n-1})(X_{n-1} - \mu_{n-1})] \\ &= \mathbb{E}[(X_0 - \mu_0)(X_0 - \mu_0)] + \mathbb{E}[(X_0 - \mu_0)(X_1 - \mu_1)] \\ &\quad + \mathbb{E}[(X_1 - \mu_1)(X_0 - \mu_0)] + \dots + \mathbb{E}[(X_{n-1} - \mu_{n-1})(X_{n-1} - \mu_{n-1})] \\ &= \text{Cov}(X_0, X_0) + \text{Cov}(X_0, X_1) + \text{Cov}(X_1, X_0) + \dots + \text{Cov}(X_{n-1}, X_{n-1}) \\ &= \sum_{i,j=0}^{n-1} \text{Cov}(X_i, X_j) \\ &= \sum_{i=0}^{n-1} \mathbb{V}(X_i) + 2 \sum_{i=0, j>i}^{n-1} \text{Cov}(X_i, X_j) \end{aligned}$$

2.5: Information Criteria

Model Evaluation

In data science, we are often presented with the necessity to compare models to find the best for our particular use case. Quite often, this boils down to a tradeoff between accuracy and interpretability. Different scenarios call for striking a different balance. Competing on a leaderboard may call for a purely *predictive* "black box" model that prioritizes accuracy above all other metrics. In contrast, much of the day-to-day work of a practicing data scientist may align more closely with a *prescriptive* framework that prioritizes usable business intelligence stemming from interpretable models.

Predictive vs. Prescriptive Data Science

Predictive vs. Prescriptive Data Science As a data scientist, you should always be asking yourself where your client's needs fall on the continuum of predictive vs. prescriptive data science. Real life data science projects more frequently fail due to a poor understanding of the client's needs than due to a lack of technical skills.

Cross-Validation and Train/Test Splits

As a data scientist, you are probably well acquainted with the use of cross-validation and train/test splits. Recall that standard cross-validation and resampling techniques [cannot be used in time series analysis](#). While methods such as the block bootstrap can still be used, they are less powerful than the standard bootstrap.

Use of a train/test split can be less valuable in time series analysis for the simple reason that we are often presented with very small datasets. It is not at all unusual to be presented with 104 observations covering two years of weekly data, or even 60 observations covering five years of monthly data^[1]. In such cases, holding out 20% of the data for a test split substantially degrades the predictive power of an already small dataset while still only providing a small sample upon which to test accuracy.

[1]: Actually, there's a good chance you'll have even fewer as at least a couple of observations are likely to be missing. In such scenarios, it is usually necessary to come up with a realistic interpolation scheme first.

For these reasons and others, we would like to find a way to supplement or even replace cross-validation and train/test splits by creating a metric to objectively evaluate models trained on an entire dataset. The methods we will employ consist of *information criteria*.

Information Criteria

Accuracy vs. Complexity

Information criteria are built around the premise that we wish to find ways to balance rewarding models for greater accuracy with penalizing them for greater complexity. Put differently, information criteria ask the question "Does the additional complexity added to a model by a new parameter justify its inclusion by a **sufficient** increase in accuracy?"

In essence information criteria consist of two terms:

1. A penalty for greater complexity as measured by parameter count.
2. A reward for greater accuracy as measured by likelihood.

Likelihood

Likelihood can be thought of as a cousin of Bayesian posterior probability. Namely, the likelihood function $\mathcal{L}$ for n observations with p parameter(s) $\theta \triangleq (\theta_0, \theta_1, \dots, \theta_{p-1})$ is defined as

$$\mathcal{L}_n(\theta) = \prod_{i=1}^n \mathbb{P}(x_i \mid x_1, x_2, \dots, x_{i-1}, \theta) \quad (2.5.1)$$

i.e. the product of probabilities that we would observe x_i given parameter(s) θ and all prior observations through x_{i-1} .

Note that likelihood is not a probability density function itself and has no requirement to sum to unity.

In general we use the log-likelihood denoted as $\ell_n(\theta)$ and defined as

$$\ell_n(\theta) \triangleq \ln(\mathcal{L}_n(\theta)) \quad (2.5.2)$$

Since $\mathcal{L}$ is a strictly positive function, the same parameters that maximize $\mathcal{L}$ will also maximize ℓ .

Exercise 2.5.1

Why are we not concerned that taking the logarithm of $\mathcal{L}$ will result in a logarithm of zero error or complex numbers stemming from the logarithm of negative numbers?

Answer available at the end of this section.

Akaike Information Criterion

The Akaike information criterion (AIC), formulated by Hirotugu Akaike and introduced in the early 1970s (Akaike (1998)) is designed to approximate the Kullback-Leibler (KL) divergence between a suggested model and the true (unknown) process model. The AIC of a model S with p parameters can be defined as

$$-2\ell_S + 2p \quad (2.5.3)$$

AIC Definition

There are multiple competing definitions of AIC in use, the definition in Eq. 2.5.3 follows the usage in statsmodels, in which case lower AIC indicates better model performance. Other sources such as Wasserman (2004) differ by a factor of 2 or even -2. While dividing by positive 2 will not change the ordering of models, dividing by -2 will reverse the order resulting in higher AIC being better.

Always check whether your program considers higher or lower AIC to be better, and never compare AIC values across programs without first verifying that they use the same definition!

Note that as written ℓ , and consequently AIC, will depend on the number of observations in our sample. This is usually not an issue as we generally compare models trained on the same dataset. Nevertheless, you should keep this factor in mind anytime you are tempted to compare AIC across different projects with different datasets.

Bayesian Information Criterion

The Bayesian information criterion is, on the surface, extremely similar to AIC. BIC is defined as

$$-2\ell_S + p \ln(n) \quad (2.5.4)$$

for model S with p parameters and n observations in the training data.

BIC Definition

As with AIC above, our definition in Eq. 2.5.4 follows the definition used in statsmodels in which lower BIC corresponds to better model performance. Other sources differ in their definitions by a factor of ± 2 .

At first glance, it might seem strange that BIC penalizes the number of observations; isn't having a larger training dataset a good thing? The rationale behind this design is that we should be more certain about the likelihood if we have more observations with which to calculate it, thus the same ℓ at different n values should be treated differently.

Significance of AIC or BIC Values

When comparing models using AIC or BIC, the raw numbers have almost no significance. Instead, you should focus on the *relative* values of how far models fall from each other. I.e., 20 vs. 10 should be treated the same as 20,010 vs. 20,000, as in both cases $\Delta = 10$. So how large a Δ do we require to indicate significance? gives a nice guideline for AIC that I would also consider valid for BIC:

- $|\Delta| \leq 2$: Very little evidence to prefer either model.
- $4 \leq |\Delta| \leq 7$: Some evidence to prefer better scoring model.
- $|\Delta| > 10$: Very strong evidence to prefer better scoring model.

The values above should be divided by 2 if your program differs from our definitions of AIC and BIC (Eqs. 2.5.3 2.5.4 by a factor of ± 2).

AIC vs. BIC

While similar in form to AIC, and often used in the same role, it is worth noting that from a *theoretical* standpoint, AIC and BIC are in fact very different. As mentioned, AIC was derived from an information-theoretic standpoint with the goal of approximating the KL divergence between a proposed model and the true generating model. Crucially, AIC does not assume that the true model is in our set of models—it merely asks how similar a given model is to the true model.

While BIC can also be motivated from an information-theoretic standpoint, this is not its standard interpretation. Instead, BIC is motivated from a Bayesian perspective[2]. BIC starts with the assumption that the true model is lurking somewhere in our set of possible models and seeks to discover which model is most probable to be the true one. Assuming a uniform prior distribution over models, it can be shown that the best BIC exactly maximizes the posterior probability of a model. Even without a uniform prior, BIC is generally understood to select the model with the highest posterior probability.

[^2]: Strictly, AIC could also be motivated by a Bayesian argument by starting with a different prior.

Differences in philosophy notwithstanding, in practice AIC and BIC tend to give very similar results. The main difference is that BIC weights the number of parameters more heavily than AIC (2 for AIC vs. $\ln(n)$ for BIC). As a result, BIC often tends to favor slightly more parsimonious but less accurate models than AIC. Put differently, AIC is more susceptible to picking overfit models, while BIC is more susceptible to picking underfit models. Some studies indicate that AIC outperforms BIC for small sample sizes, while BIC excels for larger sample sizes. Studies have also demonstrated that BIC may perform better for models with small or large effect sizes, while AIC may perform better for moderate effect sizes (Vrieze (2012)).

Ultimately, my advice is to use both AIC and BIC. Chances are they'll agree anyway in more than half the cases you're likely to encounter. When they disagree, ask why they are likely to be disagreeing. Is AIC picking an overfit model, or is BIC picking an underfit one? How large are the effect sizes? Finally, what's most valuable for your client, parsimony and interpretability or predictive accuracy? There is no definitive best in the AIC vs. BIC debate, but these questions can help you identify the best one for a given scenario.

Other Information Criteria

AIC and BIC are by far the most widely used information criteria. There are a handful of other ones in use that are worth being aware of that we will touch on briefly.

1. Akaike information criteria, corrected (AICc): As noted above, AIC can pick overfit models, especially for small sample sizes. AICc addresses this by introducing an additional penalty term to more harshly penalize the number of parameters for models trained on small datasets.

$$\begin{aligned} AICc &\triangleq AIC + \frac{2p^2 + 2p}{n - p - 1} \\ &= -2\ell_S + \frac{2np}{n - p - 1} \end{aligned} \quad (2.5.5)$$

2. Hannan-Quinn information criteria (HQIC): Frequently cited but rarely used in practice, HQIC in effect strikes a balance between AIC and BIC by adding an additional level of logarithm and a factor of 2 to the second term in BIC.

$$HQIC \triangleq -2\ell_S + 2p \ln(\ln(n)) \quad (2.5.6)$$

3. Mallows's C_p : Very similar to AIC but limited to use in linear regression. Mallows's C_p balances lack of fit measured by the residual sum of squares (RSS) with model complexity measured by parameter count and estimated population variance.

$$C_p \triangleq \frac{1}{n} (RSS + 2p\hat{\sigma}^2) \quad (2.5.7)$$

Exercise 2.5.1

Can information criteria such as AIC be used for random forests? If so, how would you go about it? If not, why not?

Answer available at the end of this section.

Footnotes

1. Actually, there's a good chance you'll have even fewer as at least a couple of observations are likely to be missing. In such scenarios, it is usually necessary to come up with a realistic interpolation scheme first.
2. Strictly, AIC could also be motivated by a Bayesian argument by starting with a different prior.

References

1. Akaike, H. (1998). Information Theory and an Extension of the Maximum Likelihood Principle. In Selected Papers of Hirotugu Akaike (pp. 199-213). Springer New York. [10.1007/978-1-4612-1694-0_15](#)
2. Wasserman, L. (2004). All of Statistics: A Concise Course in Statistical Inference. In Springer Texts in Statistics. Springer New York. [10.1007/978-0-387-21736-9](#)
3. Burnham, K. P., & Anderson, D. R. (2004). Multimodel Inference. Sociological Methods & Research, 33(2), 261-304. [10.1177/0049124104268644](#)
4. Vrieze, S. I. (2012). Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). Psychological Methods, 17(2), 228-243. [10.1037/a0027127](#)

Answers

Exercise 2.5.1

$\mathbb{P}(x_i | \theta)$ is a probability density and as such can never be negative. While in theory one could construct a scenario in which $\mathbb{P}(x_i | \theta)$ is 0 (e.g. the probability of observing a 7 on a single die roll), in practice this will only arise if our model is mis-specified and/or there's an error in our data. A correctly specified model should never "argue with the data" by predicting zero probability density at the point of an actual observation.

Exercise 2.5.1

No, information criteria cannot be used for random forests (at least not without a great many assumptions). Information criteria use the number of parameters of a model and the likelihood, both of which are problematic for random forests:

1. There is no clear way to count the number of parameters in a random forest. Do we use the number of trees, the depth of the trees, a combination of the two, or something else entirely?
2. Even if we define a way to count the number of parameters in a random forest, there is no direct way to calculate the likelihood. Linear models can be calculated using maximum likelihood estimation (MLE). Indeed, the standard solution to ordinary least squares is equivalent to MLE. In contrast, random forests are not trained with MLE, and do not have any likelihood function defined.

3: AUTOCOVARIANCE AND AUTOCORRELATION

3.1: Introduction

3.2: Autocovariance

3.3: Stationarity

3.4: Autocorrelation

3.5: Cross-Covariance and Cross-Correlation

3.6: Estimation of Autocovariance, Autocorrelation, and Cross-Correlation

This page titled [3: Autocovariance and Autocorrelation](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

3.1: Introduction

This chapter launches into time series in earnest by introducing one of the most fundamental and powerful tools in our time series analysis arsenal, *autocovariance*. We will also discuss autocorrelation, cross-correlation, and examine correlation of both synthetic and actual time series.

This page titled [3.1: Introduction](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

3.2: Autocovariance

Autocovariance Definition

As discussed in [the first chapter](#), serial correlation of time series is both one the most challenging and most valuable aspects of time series. *Autocovariance* allows us to exploit serial correlation to address questions such as:

- Is a high market close on one day indicative of a lower close the following day?
- How long does a heat wave continue to skew temperatures above the norm?
- Given higher sunspot activity this year, what year(s) in the future should we expect a repeat of this activity?

To give a concrete example, if we wish to know how strongly today's highest temperature influences tomorrow's, we may take the covariance of the series $\mathbf{T} = (T_0, T_1, \dots, T_{n-1})$ and its one-step-ahead counterpart $\mathbf{T}' = (T_1, T_2, \dots, T_n)$:

$$\text{Cov}(\mathbf{T}, \mathbf{T}'), \quad (3.2.1)$$

A high positive value for Eq. 3.2.1 tells us that higher (lower) temperatures today are indicative of higher (lower) temperatures tomorrow. While implausible for weather, in general many time series exhibit negative correlation, in which case a higher (lower) value at one time step is indicative of a lower (higher) value in the next.

We can naturally extend Eq. 3.2.1 to larger time lags to ask how long an anomalous temperature continues to skew the daily high. We will see in subsequent chapters that it is very common for time series to obey an exponentially decaying autocovariance.

Differing Definitions

The value that we have defined as autocovariance is often referred to instead as *autocorrelation* in the fields of electrical engineering and signal processing. Our definition is consistent with the common usage in statistics.

Notation

Following the notation used in sources such as [Shumway and Stoffer, 2025](#) and [Brockwell & Davis \(2016\)](#) we will use the notation $\gamma_x(s, t)$ to denote the autocovariance at time lags s and t for time series x :

$$\gamma_x(s, t) \triangleq \mathbb{E}[(x_s - \mu_s)(x_t - \mu_t)] = \mathbb{E}[x_s x_t] - \mu_s \mu_t \quad (3.2.2)$$

Note that $\gamma_x(s, t) = \gamma_x(t, s)$.

In general, we will drop the x subscript when it is obvious which time series we are referring to and only write $\gamma(s, t)$. For an arbitrarily large (or infinite) observation window, $\mu_s = \mu_t$ as both series share almost all the same observations, leading to the simplification

$$\gamma(s, t) = \mathbb{E}[x_s x_t] - \mu^2 \quad (3.2.3)$$

where we have dropped the subscript on μ . In the [next section](#), we will cover stronger justification for this simplification stemming from the principle of *stationarity*.

Mean and Autocovariance of Moving Average

Recall from [the first chapter](#) that a white noise process is defined as a mean zero process with finite variance

$$w_t \sim \text{wn}(0, \sigma_w^2). \quad (3.2.4)$$

A summation of multiple values of w_t will still have a mean of zero

$$\mathbb{E}[\dots + w_{t-1} + w_t + w_{t+1} + \dots] = \dots + \mathbb{E}[w_{t-1}] + \mathbb{E}[w_t] + \mathbb{E}[w_{t+1}] + \dots = 0. \quad (3.2.5)$$

Consider the moving average

$$v_t = \frac{1}{3}(w_{t-1} + w_t + w_{t+1}). \quad (3.2.6)$$

v_t has a mean of zero, but the autocovariance will depend on the difference between s and t :

$$\gamma_v(s, t) = \begin{cases} \frac{3}{9}\sigma_w^2, & s = t \\ \frac{2}{9}\sigma_w^2, & |s - t| = 1 \\ \frac{1}{9}\sigma_w^2, & |s - t| = 2 \\ 0, & \text{otherwise} \end{cases} \quad (3.2.7)$$

Note that **even if all instances of the initial time series are iid**, averaging will still introduce serial correlation.

Exercise 3.2.1

Prove Eq. 3.2.7. Hint, since the w_t 's are iid, $\text{Cov}(w_i, w_j) = \delta_{ij}\sigma_w^2$ where δ_{ij} is defined as 1 if $i = j$ and 0 otherwise.

Answer available at the end of this section.

Mean and Autocovariance of Random Walk

Mean for Zero Drift

Recall that a random walk (without drift) is defined as:

$$x_t = x_{t-1} + w_t \quad (3.2.8)$$

Iterating backwards, we have

$$\begin{aligned} x_t &= x_{t-1} + w_t \\ &= (x_{t-2} + w_{t-1}) + w_t \\ &= x_{t-2} + w_{t-1} + w_t \\ &= x_{t-3} + w_{t-2} + w_{t-1} + w_t \\ &\dots \\ &= \sum_{h=0}^{\infty} w_{t-h} \end{aligned} \quad (3.2.9)$$

Of course, in real life we cannot have an infinite number of observations. Calling the first observation w_0 , this gives

$$\begin{aligned} x_t &= w_0 + w_1 + \dots + w_t \\ &= \sum_{h=0}^t w_{t-h} \end{aligned} \quad (3.2.10)$$

Whether we use Eq. 3.2.12 or 3.2.13, the expectation value for the mean will always be 0, i.e.

$$\begin{aligned} \mathbb{E}[x_t] &= \mathbb{E}\left[\sum w_{t-h}\right] \\ &= \sum \mathbb{E}[w_{t-h}] \\ &= 0. \end{aligned} \quad (3.2.11)$$

Mean with Drift

A random walk with drift is defined as

$$x_t = \delta + x_{t-1} + w_t, \quad \delta \neq 0. \quad (3.2.12)$$

Following the same reasoning used above in Eqs 3.2.12 and 3.2.13, we can iterate backwards to obtain

$$\begin{aligned} x_t &= \delta + x_{t-1} + w_t \\ &= \delta + (\delta + x_{t-2} + w_{t-1}) + w_t \\ &= 2\delta + x_{t-2} + w_{t-1} + w_t \\ &= 3\delta + x_{t-3} + w_{t-2} + w_{t-1} + w_t \\ &\dots \\ &= t\delta + \sum_{h=0}^t w_{t-h} \end{aligned} \quad (3.2.13)$$

where we have only used the finite case to avoid $t\delta$ going to $\pm\infty$.

The expectation value for Eq.3.2.16 is

$$\begin{aligned} \mathbb{E}[x_t] &= \mathbb{E}\left[t\delta + \sum_{h=0}^t w_{t-h}\right] \\ &= \mathbb{E}[t\delta] + \sum_{h=0}^t \mathbb{E}[w_{t-h}] \\ &= t\delta \end{aligned} \quad (3.2.14)$$

Random Walk Variance

Recall that while adding constants to random variables adjusts the means, it does not affect the variance or covariance, i.e.

$$\mathbb{V}(a + X) = \mathbb{V}(X), \quad (3.2.15)$$

and

$$\text{Cov}(a + X, b + Y) = \text{Cov}(X, Y), \quad (3.2.16)$$

thus the autocovariance of a random walk is independent of drift terms. To simplify notation we will therefore only explicitly address a random walk without drift.

Exercise 3.2.1

Prove Eqs 3.2.18 and 3.2.19.

Answer available at the end of this section.

The autocovariance at lags s and t is

$$\text{Cov}\left(\sum_{i=0}^s w_i, \sum_{j=0}^t w_j\right) = \text{Cov}(w_0, w_0) + \text{Cov}(w_0, w_1) + \dots + \text{Cov}(w_s, w_t).$$

By the assumption of independent w_t 's, the only non-zero terms will be when $i = j$ (i.e. the variances). Assuming without loss of

generality that $t \leq s$, we have

$$\begin{aligned}\text{Cov}\left(\sum_{i=0}^s w_i, \sum_{j=0}^t w_j\right) &= \sum_{i=0}^t \mathbb{V}(w_i), \quad t \leq s \\ &= (t+1)\sigma_w^2\end{aligned}$$

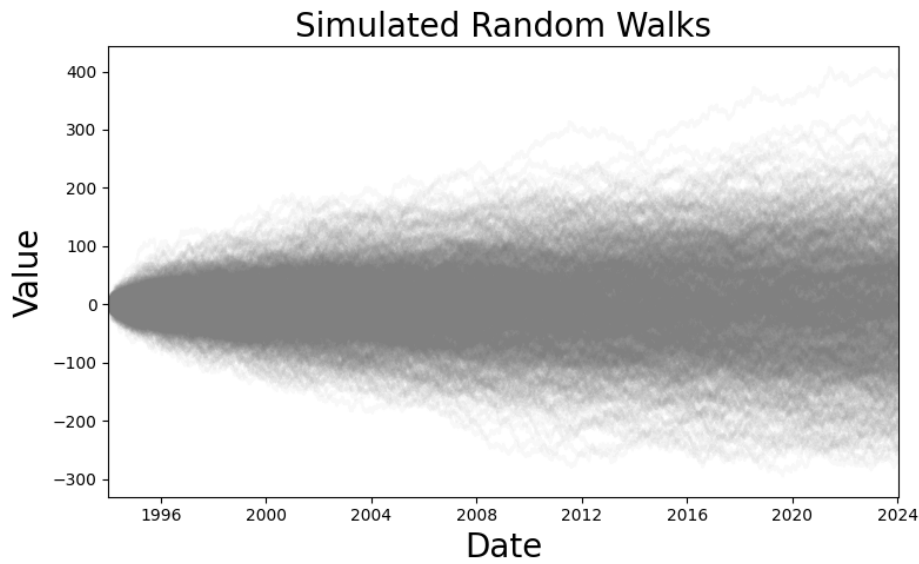
From Eq. 3.2.23 we see that the variance of a random walk $\gamma(t, t)$ increases linearly with respect to t , i.e.

$$\gamma(t, t) = (t+1)\sigma_w^2 \quad (3.2.17)$$

equivalently, the standard deviation of a random walk increases with respect to $\sqrt{t+1}$

$$\text{s.d.} = \sqrt{t+1}\sigma_w \quad (3.2.18)$$

Figure 3.2.1 displays how random walks spread out with respect to time. Note how the random walks more closely follow a square root rather than a linear spread, which provides a concrete example of why standard deviation is often favored over variance in analysis.



3.2.1: 1,000 simulated random walks with $\sigma_w^2 = 1$ and $\delta = 0$.

References

1. Shumway, R. H., & Stoffer, D. S. (2025). Time Series Analysis and Its Applications: With R Examples. In *Springer Texts in Statistics*. Springer Nature Switzerland. [10.1007/978-3-031-70584-7](#)
2. Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting. In *Springer Texts in Statistics*. Springer International Publishing. [10.1007/978-3-319-29854-2](#)

Answers

Exercise 3.2.1

Let's go through the four cases in order:

$$\begin{aligned}\gamma_v(t, t) &= \text{Cov}\left(\frac{1}{3}(w_{t-1} + w_t + w_{t+1}), \frac{1}{3}(w_{t-1} + w_t + w_{t+1})\right) \\ &= \frac{1}{9} [\text{Cov}(w_{t-1}, w_{t-1}) + \text{Cov}(w_{t-1}, w_t) + \dots + \text{Cov}(w_{t+1}, w_{t+1})] \\ &= \frac{1}{9} [\text{Cov}(w_{t-1}, w_{t-1}) + \text{Cov}(w_t, w_t) + \text{Cov}(w_{t+1}, w_{t+1})] \\ &= \frac{1}{9} [\mathbb{V}(w_{t-1}) + \mathbb{V}(w_t) + \mathbb{V}(w_{t+1})] \\ &= \frac{3}{9}\sigma_w^2\end{aligned} \quad (3.2.19)$$

where we have exploited the fact that the covariance between different instances of w_t is 0 to move between the second and third lines.

$$\begin{aligned}
 \gamma_w(t, t+1) &= \text{Cov} \left(\frac{1}{3} (w_{t-1} + w_t + w_{t+1}), \frac{1}{3} (w_t + w_{t+1} + w_{t+2}) \right) \\
 &= \frac{1}{9} [\text{Cov}(w_t, w_t) + \text{Cov}(w_{t+1}, w_{t+1})] \\
 &= \frac{1}{9} [\mathbb{V}(w_t) + \mathbb{V}(w_{t+1})] \\
 &= \frac{2}{9} \sigma_w^2
 \end{aligned} \tag{3.2.20}$$

$$\begin{aligned}
 \gamma_w(t, t+2) &= \text{Cov} \left(\frac{1}{3} (w_{t-1} + w_t + w_{t+1}), \frac{1}{3} (w_{t+1} + w_{t+2} + w_{t+3}) \right) \\
 &= \frac{1}{9} \text{Cov}(w_{t+1}, w_{t+1}) \\
 &= \frac{1}{9} \mathbb{V}(w_{t+1}) \\
 &= \frac{1}{9} \sigma_w^2
 \end{aligned} \tag{3.2.21}$$

Finally, note that following the above derivations, if $|s - t| \geq 3$, any two instances of v_t and v_s will have zero terms in common, leading to 0 autocovariance.

Exercise 3.2.1

We'll first prove Eq. 3.2.18

Solution: We'll first prove Eq. (18):

$$\begin{aligned}
 \mathbb{V}[a + X] &= \mathbb{E}[(a + X - \mathbb{E}[a + X])^2] \\
 &= \mathbb{E}[(a + X - \mathbb{E}[a] - \mathbb{E}[X])^2] && \text{Expectation is a linear operator} \\
 &= \mathbb{E}[(a + X - a - \mathbb{E}[X])^2] && \text{Expectation value of a constant is the constant} \\
 &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\
 &= \mathbb{V}[X]
 \end{aligned} \tag{3.2.22}$$

Intuitively, this should make sense. Imagine I gather a group of people and measure their mean height as 175 cm with a standard deviation of 5 cm (variance of 25 cm²). I then have them stand on a 25 cm box and measure their heights again. The mean will now be 200 cm, but the variance about mean will remain unchanged.

We can prove Eq. 3.2.19 as follows:

$$\begin{aligned}
 \text{Cov}[a + X, b + Y] &= \mathbb{E}[(a + X - \mathbb{E}[a + X]) \cdot (b + Y - \mathbb{E}[b + Y])] \\
 &= \mathbb{E}[(a + X - \mathbb{E}[a] - \mathbb{E}[X]) \cdot (b + Y - \mathbb{E}[b] - \mathbb{E}[Y])] \\
 &= \mathbb{E}[(a + X - a - \mathbb{E}[X]) \cdot (b + Y - b - \mathbb{E}[Y])] \\
 &= \mathbb{E}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])] \\
 &= \text{Cov}[X, Y]
 \end{aligned} \tag{3.2.23}$$

3.3: Stationarity

Strict vs. Weak Stationarity

Now that we are familiar with the autocovariance, we can address *stationarity*, one of the most central concepts in time series analysis. Loosely, a stationary time series is one that does not demonstrate any trend or seasonal variation. More concretely, there are two distinct definitions of stationarity:

1. *Strict stationarity* is defined as a time series for which the probabilistic behavior of any subset of observations is identical, i.e.

$$\mathbb{P}_x(x_{t_1} < c_1, x_{t_2} < c_2, \dots, x_{t_k} < c_k) = \mathbb{P}_x(x_{t_1+h} < c_1, x_{t_2+h} < c_2, \dots, x_{t_k+h} < c_k). \quad (3.3.1)$$

2. *Weak stationarity* is defined as a time series for which the following hold:

- i. The mean $E[x_t]$ is finite and constant across all times t .
- ii. The variance $\mathbb{V}(x_t) = \gamma(t, t)$ is finite and constant across all times t .
- iii. The autocovariance is a function of lag alone and not of absolute time, $\gamma(s, t) = \gamma(s + v, t + v) = \gamma(|s - t|)$.

An example of a strictly stationary series would be pure (iid) white noise. Stock returns (*not* values) are also reasonably approximated as being strictly stationary. Weakly stationary processes are far more prevalent; in coming chapters we will encounter numerous examples from fields such as medicine and econometrics.

Strict to Weak Stationarity

It should be obvious that demonstrating weak stationarity does not demonstrate strict stationarity^[^1]. Interestingly, the converse also does not necessarily hold. Consider the white noise process defined by the standard Cauchy distribution:

[^1]: It can be proved that if the process that generates x_t is a Gaussian process, then weak stationarity does demonstrate strict stationarity. This is not overly useful in practice as it requires us to know x_t 's hidden generating process.

$$x_t \sim \text{Cauchy}(0, 1) \quad (3.3.2)$$

x_t is strictly stationary as every point is drawn from the same probability distribution as every other point. However, x_t is *not* weakly stationary as the Cauchy distribution lacks a finite variance. That said, any strictly stationary finite variance process is also weakly stationary.

Advantages of Weak Stationarity

In practice, we generally do not use strict stationarity for two major reasons:

1. Establishing strict stationarity requires knowing the underlying probability distribution (or at least having a confident guess to it). In contrast, weak stationarity is established solely via observables in the form of the mean, variance, and autocovariance.
2. Strict stationarity is generally too strict for most purposes. We will see in coming chapters that methods requiring stationarity can be used with weakly stationary time series.

Stationarity

Stationarity Following sources such as and , we will use the term "stationary" to refer to weak stationarity and specify "strict stationarity" when necessary.

Exercise

Consider the process v_t from [Chapter 1](#) defined as

$$v_t \triangleq \frac{1}{3}(w_{t-1} + w_t + w_{t+1}) \quad (3.3.3)$$

where $w_t \sim \text{iid}(0, \sigma_w^2)$. Is this process weakly stationary? Is it strictly stationary?

Answer available at the end of this section.

Autocovariance of a Stationary Process

As discussed [above](#), the autocovariance $\gamma(s, t)$ of a stationary process depends exclusively on the lag $h = |s - t|$. Consequently, when dealing with stationary processes we will simply use the notation

$$\gamma(h), \quad h = |s - t| \quad (3.3.4)$$

where $\gamma(0)$ is the variance, $\gamma(1)$ is the autocovariance at lag 1, and so on. Note that

$$\gamma(-h) = \gamma(|t - s|) = \gamma(|s - t|) = \gamma(h), \quad (3.3.5)$$

consequently, we will only calculate $\gamma(h)$ for $h \geq 0$.

Exercise

What can we say about the relationship between $\gamma(h)$ and $\gamma(0)$ given that the relevant time series is stationary?

Answer available at the end of this section.

Autocovariance of a Stationary Process is Positive Semidefinite

We are now in a position to prove a property of autocovariance that will be useful in future chapters. The autocovariance matrix Γ is defined as

$$\Gamma = \begin{bmatrix} \gamma(0) & \gamma(1) & \dots & \gamma(h) \\ \gamma(1) & \gamma(0) & \dots & \gamma(h-1) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(h) & \gamma(h-1) & \dots & \gamma(0) \end{bmatrix}. \quad (3.3.6)$$

Γ for a stationary process, like standard covariance matrices, is *positive semidefinite*. A positive semidefinite matrix is defined as a symmetric matrix [\[2\]](#) $\mathbf{X} \in \mathbb{R}^{n \times n}$ with the property that for any vector $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{v}^T \mathbf{X} \mathbf{v} \geq 0. \quad (3.3.7)$$

To demonstrate that Γ is positive semidefinite, let

$$\boldsymbol{\ell} \triangleq (\ell_0, \ell_1, \dots, \ell_h) \quad (3.3.8)$$

and

$$\mathbf{x}_t \triangleq (x_t, x_{t-1}, \dots, x_{t-h}) \quad (3.3.9)$$

so that

$$\boldsymbol{\ell}^T \mathbf{x}_t = \ell_0 x_t + \ell_1 x_{t-1} + \dots + \ell_h x_{t-h}. \quad (3.3.10)$$

As variance cannot be negative, we have

$$\mathbb{V}(\ell^T \mathbf{x}_t) = \mathbb{V}(\ell_0 x_t + \ell_1 x_{t-1} + \dots + \ell_h x_{t-h}) \geq 0, \quad (3.3.11)$$

or using the [variance of a sum](#)

$$\begin{aligned} \mathbb{V}(\ell) &= \sum_{i=0}^h \sum_{j=0}^h \ell_i \ell_j \text{Cov}(x_{t-i}, x_{t-j}) \\ &= \sum_{i=0}^h \sum_{j=0}^h \ell_i \ell_j \gamma(|i-j|) \\ &= \begin{bmatrix} \ell_0 & \ell_1 & \dots & \ell_h \end{bmatrix} \begin{bmatrix} \gamma(0) & \gamma(1) & \dots & \gamma(h) \\ \gamma(1) & \gamma(0) & \dots & \gamma(h-1) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(h) & \gamma(h-1) & \dots & \gamma(0) \end{bmatrix} \begin{bmatrix} \ell_0 \\ \ell_1 \\ \vdots \\ \ell_h \end{bmatrix} \\ &= \ell^T \mathbf{\Gamma} \ell \\ &\geq 0. \end{aligned} \quad (3.3.12)$$

As the above holds for any possible vector $\ell \in \mathbb{R}^{h+1}$ we conclude that $\mathbf{\Gamma}$ must be positive semidefinite.

Eigenvalues

It can be demonstrated that an equivalent definition for "positive semidefinite" is the requirement that all eigenvalues be non-negative.

Footnotes

1. It can be proved that if the process that generates x_t is a Gaussian process, then weak stationarity does demonstrate strict stationarity. This is not overly useful in practice as it requires us to know x_t 's hidden generating process.
2. The definition of positive semidefiniteness can be expanded to include complex vectors and matrices, but that will not be necessary for this book.

References

1. Shumway, R. H., & Stoffer, D. S. (2025). Time Series Analysis and Its Applications: With R Examples. In *Springer Texts in Statistics*. Springer Nature Switzerland. [10.1007/978-3-031-70584-7](#)
2. Brockwell, P. J., & Davis, R. A. (1991). Time Series: Theory and Methods. In *Springer Series in Statistics*. Springer New York. [10.1007/978-1-4419-0320-4](#)

Answers

Exercise

The expectation value (mean) of v_t is always 0. As demonstrated in [the previous section](#), the autocovariance is exclusively a function of $|s-t|$. Thus the process is weakly stationary.

v_t is also strictly stationary as every instance of v_t is drawn from same distribution as every other instance. Note that demonstrating strict stationarity for a finite variance process also proves weak stationarity.

Exercise

Click to reveal solution **Solution:** As demonstrated in [Chapter 2](#), for any finite variance random variables X and Y

$$|\text{Cov}(X, Y)| \leq \sqrt{\mathbb{V}(X)\mathbb{V}(Y)}. \quad (3.3.13)$$

As $\gamma(s, s) = \gamma(t, t) = \gamma(0)$ and $\gamma(s, t) = \gamma(|s-t|) = \gamma(h)$, we conclude

$$|\gamma(h)| \leq \gamma(0). \quad (3.3.14)$$

This in turn guarantees that the value $\frac{\gamma(h)}{\gamma(0)}$ lies in the interval $[-1, 1]$.

3.4: Autocorrelation

Autocorrelation Definition

As discussed in [Chapter 2](#), covariance, and by extension autocovariance, has some undesirable characteristics. For this reason, in time series analysis we often prefer to work with the *autocorrelation*, which is a unitless quantity bounded by $[-1, 1]$.

Differing Definitions

As with autocovariance in the [second section of this chapter](#), the definition of autocorrelation we use is the one from statistics. This definition differs slightly from the one used in disciplines such as electrical engineering.

For any finite variance process, the autocorrelation is simply the autocovariance normalized by the variances.

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}. \quad (3.4.1)$$

When considering stationary time series, we can employ the simplifications

$$\gamma(s, t) = \gamma(|s - t|) = \gamma(h), \quad (3.4.2)$$

and

$$\gamma(s, s) = \gamma(t, t) = \gamma(0). \quad (3.4.3)$$

As such, a stationary time series has autocorrelation given by

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}. \quad (3.4.4)$$

Note that $\rho(h) = \rho(-h)$ and $\rho(0) = 1$.

Exercise

Find $\rho(h)$ for $h = 0, 1, 2, \dots$ for the moving average v_t from [Chapter 1](#).

Answer available at the end of this section.

Autocorrelation is Positive Semidefinite

The autocorrelation matrix $\mathbf{P}$ is related to the autocovariance matrix $\mathbf{\Gamma}$ by

$$\mathbf{P} = \frac{1}{\gamma(0)}\mathbf{\Gamma}. \quad (3.4.5)$$

$\gamma(0)$ is the variance, and hence cannot be negative, so $\mathbf{v}^T\mathbf{P}\mathbf{v}$ will only differ from $\mathbf{v}^T\mathbf{\Gamma}\mathbf{v}$ by a positive constant term given by $\frac{1}{\gamma(0)}$. As a result, since $\mathbf{\Gamma}$ is [positive semidefinite](#), $\mathbf{P}$ must also be as well.

Autocorrelation Plots

It is often instructive to plot the autocorrelation of a time series. [statsmodels](#) has a nice built-in function to create plots:

```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(my_time_series, lags=20)
```

where the parameter `lags` controls the maximum value of h . `lags=20` is generally a good starting choice for most scenarios, though for longer patterns such as hourly data with a 24-hour period you may want to set lags to something like 72 or 96.

`plot\_acf` has other arguments that we will discuss [once we've covered more theory](#). For now, let's take a look at a couple of different autocorrelation plots to get a feeling for how they behave.

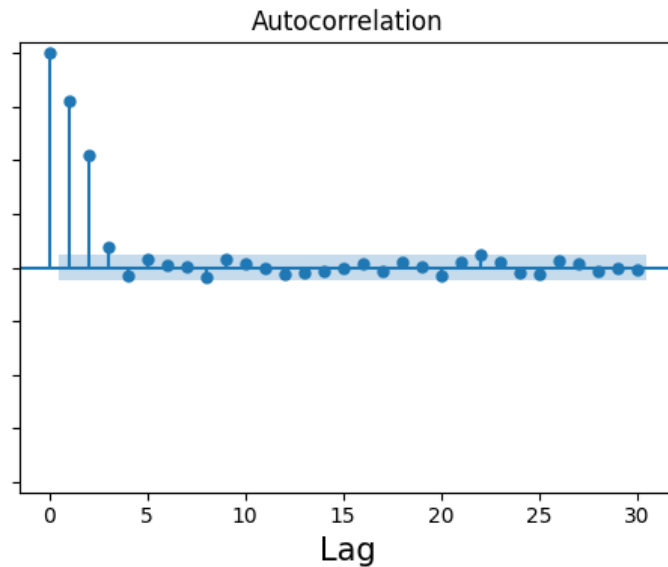


Figure 3.4.1: Sample autocorrelation plot from `statsmodels` with sudden cutoff point. Note that lag 0 is always equal to 1.

Figure 3.4.1 shows two significant lags (plus the value of 1 at $h = 0$), and falls to statistical insignificance thereafter as denoted by the blue shading. Later on we will see that this is a hallmark characteristic of a *moving average* process.

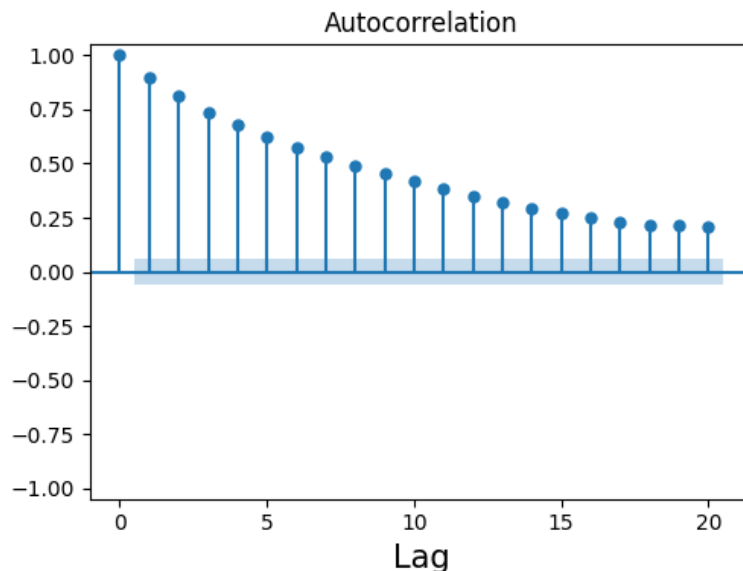


Figure 3.4.1: Sample autocorrelation plot from `statsmodels` with exponential decay.

Figure 3.4.2 demonstrates an exponentially decaying autocorrelation that remains statistically significant even at $h = 20$. In later chapters we will see that this behavior corresponds to an *autoregressive* process.

Exercise 3.4.1

One of the best ways to learn time series analysis is to experiment with time series. For this problem, download the non-seasonally adjusted US unemployment rate from the St. Louis Federal Reserve bank and examine the raw data as follows:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf

df = pd.read_csv("https://fred.stlouisfed.org/graph/fredgraph.csv?bgcolor=%23ebf3fb&char
                index_col=0,
                parse_dates=True,
                )
# statsmodels is very particular regarding missing values, we will fill in missing value
df.ffill(inplace=True)
# examine a quick plot
plt.plot(df)

-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[1], line 3
      1 import pandas as pd
      2 import matplotlib.pyplot as plt
----> 3 from statsmodels.graphics.tsaplots import plot_acf
      5 df = pd.read_csv("https://fred.stlouisfed.org/graph/fredgraph.csv?bgcolor=%23ebf
      6                 index_col=0,
      7                 parse_dates=True,
      8                 )
      9 # statsmodels is very particular regarding missing values, we will fill in missi

ModuleNotFoundError: No module named 'statsmodels'
```

What factors do you observe in the plot? Can you pick out signs of the pandemic?

Next, let's plot the autocorrelation function:

```
# Assigning the plot to "fig" prevents Jupyter from printing the plot twice.
fig = plot_acf(df, lags=48) # four years of autocorrelation
fig.show()
```

What do you observe with regard to the autocorrelation? Does it resemble either Figure 3.4.1 or Figure 3.4.2?

Answers

Exercise

In the [previous section](#) we demonstrated that v_t is both weakly and strictly stationary. Consequently, $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$. Drawing on the autocovariance from the [second section](#), we have:

$$\rho(h) = \begin{cases} \frac{\frac{3}{9}\sigma_w^2}{\frac{3}{9}\sigma_w^2} = 1, & h = 0 \\ \frac{\frac{2}{9}\sigma_w^2}{\frac{3}{9}\sigma_w^2} = \frac{2}{3}, & h = 1 \\ \frac{\frac{1}{9}\sigma_w^2}{\frac{3}{9}\sigma_w^2} = \frac{1}{3}, & h = 2 \\ \frac{\frac{0}{9}\sigma_w^2}{\frac{3}{9}\sigma_w^2} = 0, & \text{otherwise.} \\ 0, & \text{otherwit.} \end{cases} \quad (5)$$

Eq. 3.4.5 demonstrates an additional advantage of the autocorrelation in that it does not require us to know anything about the variance σ_w^2 beyond the requirement that it be finite.

This page titled [3.4: Autocorrelation](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgry](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

3.5: Cross-Covariance and Cross-Correlation

Cross-Covariance Definition

When examining the relationship between two time series, it is often useful to see how they move in tandem. One powerful tool for this is the *cross-covariance function* defined as

$$\gamma_{x,y}(s,t) \triangleq \mathbb{E}[(x_s - \mu_{x,s})(y_t - \mu_{y,t})] \quad (3.5.1)$$

for time series x_s and y_t . Analogous to stationarity for autocovariance, the two series are said to be *jointly stationary* if both:

1. x_s and y_t are individually stationary.
2. The cross-covariance is only a function of the lag $h = s - t$.

In such a case, Eq. 3.5.1 can be simplified to

$$\gamma_{x,y}(h) = \mathbb{E}[(x_{t+h} - \mu_x)(y_t - \mu_y)]. \quad (3.5.2)$$

Since we are dealing with two distinct time series, it is generally the case that $\gamma_{x,y}(h) \neq \gamma_{x,y}(-h)$ for $h \neq 0$. This is due to the fact that $\gamma_{x,y}(h)$ represents the covariance between y and x at h lags in the *future*, whereas $\gamma_{x,y}(-h)$ represents the covariance between y and x at h lags in the *past*. Put differently, there is no reason to assume the covariance between employment and the following month's inflation is the same as the covariance between employment and the preceding month's inflation.

Exercise

Why have we dropped the subscripts s and t from μ in Eq. 3.5.2?

Answer available at the end of this section.

Cross-Correlation Definition

The cross-correlation function is defined as

$$\rho_{x,y}(s,t) \triangleq \frac{\gamma_{x,y}(s,t)}{\sqrt{\gamma_x(s,s)\gamma_y(t,t)}}. \quad (3.5.3)$$

For jointly stationary time series, Eq. 3.5.3 simplifies to

$$\rho_{x,y}(h) = \frac{\gamma_{x,y}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}}. \quad (3.5.4)$$

Exercise

We have seen that for any finite variance process x_t , $\rho_x(0) = 1$. Is $\rho_{x,y}(0)$ also always 1? Why or why not?

Answer available at the end of this section.

Leading vs. Lagging

For $\gamma_{x,y}(h)$ or $\rho_{x,y}(h)$, when $h > 0$ we say y_t is a *leading indicator* for x_t (or "leads" x_t) because y_t allows us to predict what will happen to x_t in the future. Equivalently, especially in scenarios in which y_t is difficult to directly observe, we can say that x_t is a *lagging indicator* for y_t (or "lags" y_t) because it allows us to infer past values of y_t . Similarly, when $h < 0$ we say x_t is a leading indicator (or y_t is a lagging indicator).

Cross-Covariance and Cross-Correlation in statsmodels

In `statsmodels`, the cross-covariance and cross-correlation are accessed by `statsmodels.tsa.stattools.covf` and `statsmodels.tsa.stattools.ccf`, respectively. The only arguments we need to worry about at this point are `x` and `y`, the two time series. Note that these functions only calculate positive h values, to get both positive and negative values (i.e. both `x` and `y` leading) you must run both `ccf(x, y)` and `ccf(y, x)`.

Stationarity of Series

`statsmodels` uses the forms from Eqs 3.5.2 and 3.5.4, which rely on the assumptions of stationarity. If you attempt to run them on non-stationary time series you may get results such as correlations with an absolute value greater than 1. We will cover tools for making a non-stationary time series stationary in the next chapter.

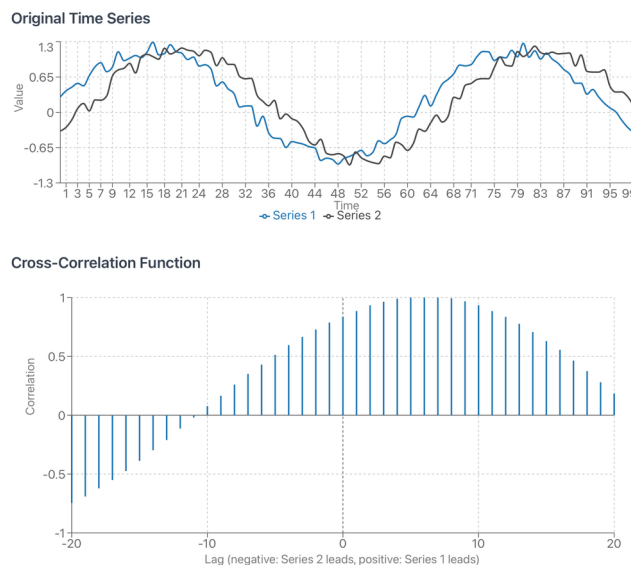


Figure 3.5.1: Notional plot of cross-correlation function for two arbitrary time series consisting of phase shifted sine curves with noise. Note that in `statsmodels` it is necessary to calculate positive and negative h values separately.

Answers

Exercise

By assumption, both x_s and y_t are individually stationary. As such, the means μ_x and μ_y are independent of time, making the subscript for time superfluous.

Exercise

No, it need not always be 1. $\rho_x(0)$ represents a pure *variance* divided by itself, resulting in 1. In contrast, even at lag 0, $\rho_{x,y}(0)$ is still the *covariance* of two random variables divided by their standard deviations. Consequently, $\rho_{x,y}(0)$ can fall anywhere in $[-1, 1]$ (by the [Cauchy-Schwarz inequality](#)).

3.6: Estimation of Autocovariance, Autocorrelation, and Cross-Correlation

Thus far, we have dealt with the theoretical values for various forms of variance, covariance, and correlation based on the assumption that we know the underlying process from which our observations have been drawn. In real life, we will usually be given the time series and then attempt to infer aspects such as the underlying process from statistical estimators. In this section we will develop techniques analogous to how sample covariance and sample correlation are used to estimate the true covariance and correlation in data science and statistics.

Estimate Definitions

Sample Autocovariance

The standard formula used in sources such as [Shumway & Stoffer \(2025\)](#) and [Brockwell & Davis \(1991\)](#) for sample autocovariance is

$$\hat{\gamma}(h) \triangleq \frac{1}{n} \sum_{t=0}^{n-h-1} (x_{t+h} - \bar{x})(x_t - \bar{x}) \quad (3.6.1)$$

where $\bar{x}$ is the sample mean and the "hat" notation indicates that $\hat{\gamma}$ is an estimate of the true population value. Note that we assume the same mean for x_t and x_{t+h} .

Exercise

Prove that the sample mean $\bar{x}$ is an unbiased estimator of true population mean μ_x .

Answer available at the end of this section.

Sample Autocorrelation

Autocorrelation is estimated by defining the sample autocorrelation

$$\hat{\rho}(h) \triangleq \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}. \quad (3.6.2)$$

Sample Cross-Covariance

$$\hat{\gamma}_{x,y}(h) \triangleq \frac{1}{n} \sum_{t=0}^{n-h-1} (x_{t+h} - \bar{x})(y_t - \bar{y}) \quad (3.6.3)$$

and the sample cross-correlation as

$$\hat{\rho}_{x,y}(h) \triangleq \frac{\hat{\gamma}_{x,y}(h)}{\sqrt{\hat{\gamma}_x(0)\hat{\gamma}_y(0)}}. \quad (3.6.4)$$

Use of Hat Notation

Going forward, we will generally drop the "hat" notation from γ and ρ when it is clear from context whether we are referring to sample or population values.

Properties of Estimators

Positive Semidefiniteness

You may have noticed something strange about Eq. 3.6.1. For n observations, we only have $n - h$ pairs to sum over; for example when $n = 100$ and $h = 2$ there are 98 pairs consisting of $(x_0, x_2), (x_1, x_3), \dots, (x_{97}, x_{99})$. Why then, do we divide Eq. 3.6.1 by n rather than $n - h$?[1]

The answer relates to the necessity that the autocovariance matrix (and by extension autocorrelation matrix) be **positive semidefinite** in order to avoid the possibility of generating negative variances. To see that the form used in Eq. 3.6.1 guarantees this property, let us sketch the proof from **Brockwell & Davis (2016)** [2]. Let us define the matrix $\mathbf{X}$ using the demeaned time series such that

$$\left(\frac{1}{\sqrt{n}}\mathbf{X}\right)^2 = \Gamma, \quad (3.6.5)$$

where the elements of $\mathbf{X}$ are the vectors $[\dots, x_0 - \bar{x}, x_1 - \bar{x}, x_2 - \bar{x}, \dots]$ appropriately padded with zeros. More precisely, we have now factored Γ such that

$$\Gamma = \frac{1}{n}\mathbf{X}^T\mathbf{X}. \quad (3.6.6)$$

Since we have expressed Γ as the square of $\mathbf{X}$, for any vector $\mathbf{v}$ we have

$$\mathbf{v}^T\Gamma\mathbf{v} = \frac{1}{n}\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} = \frac{1}{n}(\mathbf{X}\mathbf{v})^T\mathbf{X}\mathbf{v} = \frac{1}{n}\|\mathbf{X}\mathbf{v}\|^2 \geq 0. \quad (3.6.7)$$

The central point to understand is that **we could only factor Γ because we could pull out $\frac{1}{n}$ from every entry**. If we separately weighted each $\gamma(h)$ in Γ by $n - h$, we would no longer be able to express the autocovariance matrix Γ as the product of two other matrices. Consequently, we would have no guarantee that Γ was positive semidefinite, and could end up with negative variances.

Bias and Consistency

Note that neither dividing by n nor $n - h$ creates an unbiased estimator, i.e. in neither case is it true that

$$\mathbb{E}[\hat{\gamma}(h)] = \gamma(h) \quad (3.6.8)$$

though when dividing by $n - h$ the bias will be smaller.

The estimates are both, however, *consistent* in that in as $n \rightarrow \infty$

$$\hat{\gamma}(h) \xrightarrow{P} \gamma(h). \quad (3.6.9)$$

As observed in **Wasserman (2004)**, most modern statistics texts consider consistency to be sufficient and are less concerned with estimators being unbiased.

Standard Error of Autocorrelation and Cross-Correlation

A natural question to ask is how we might determine if an autocorrelation value is statistically significant. We **previously touched on** the fact that programs such as **statsmodels** estimate statistical significance automatically. But we have not yet explained *how* significance is estimated.

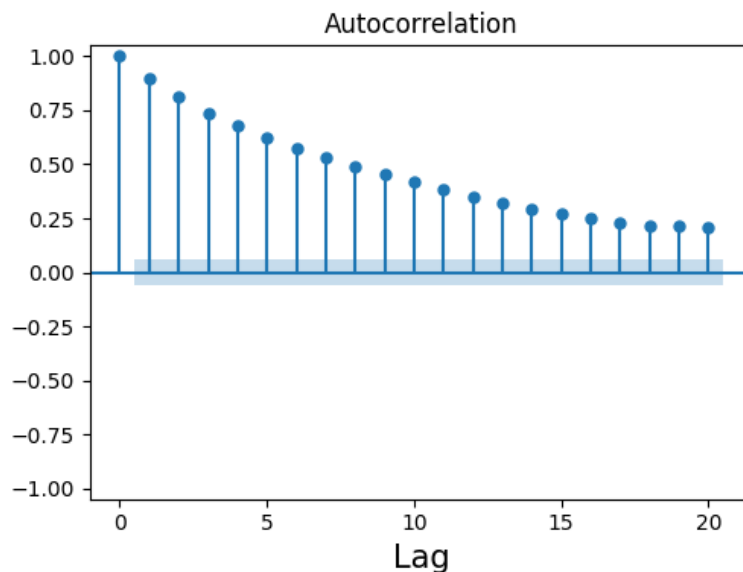


Figure 3.6.1: Sample autocorrelation plot from `statsmodels` with blue shading indicating area of statistical insignificance.

It turns out that in order to definitively calculate statistical significance, we would need to know the underlying process that generated our data—exactly the question we are trying to answer in the first place. Instead, let's ask a slightly different question: What values should we expect for the autocorrelation if our time series is pure white noise?

Under the assumption of white noise[3], we can expect the sample autocorrelation values to be normally distributed with mean 0 and variance $\frac{1}{n}$

$$\hat{\rho}(h) \sim \mathcal{N}\left(0, \frac{1}{n}\right) \quad (3.6.10)$$

where n is the total number of observations (**Box et al. (2008)**). Equivalently, the standard error of $\hat{\rho}(h)$ is given by

$$\text{se}(\hat{\rho}(h)) = \frac{1}{\sqrt{n}}. \quad (3.6.11)$$

For a white noise process, we expect around 95% of all sample autocorrelation values to fall within $\pm \frac{2}{\sqrt{n}}$. By default, `statsmodels` usually uses Eq. 3.6.11 at a 95% confidence interval to estimate statistical significance to create figures such as `{ref}ar1-fig[4]`. It can be shown that this formula also holds for estimating the sample cross-correlation $\hat{\rho}_{x,y}(h)$.

Approximate Nature of Standard Error

Approximate Nature of Standard Error Eq. (3.6.11) is only a rough approximation of the standard error of the autocorrelation and cross-correlation functions, and is not strictly valid for time series that are not simple white noise. Use it as a general guide, but don't try to read too closely into exact confidence levels.

Footnotes

- While Eq. (1) is the default method for calculating sample autocovariance in `statsmodels`, the autocovariance and autocorrelation (found in `statsmodels.tsa.stattools.acovf` and `statsmodels.tsa.stattools.acf`, respectively) have an argument `adjusted` that can be set to `True` in order to divide by $n - h$ instead. By default, both functions set `adjusted=False` (i.e. dividing by n by default). The cross-covariance and cross-correlation functions in `statsmodels` default to `adjusted=True` as they do not share the same requirement to be positive semidefinite. Our definition in Eq. (6) follows sources such as **Box et al. (2008)** and **Shumway & Stoffer (2025)**.
- The full proof is unnecessary for our purposes, but can be found in Sec. 2.4.2 of the referenced text for those who are interested.
- Strictly, this formula is only valid for noise with a finite fourth moment such as Gaussian white noise (**Shumway & Stoffer**

(2025)). As the formula is in any event an approximation for any non-white noise time series we will not be too concerned about this point.

4. `statsmodels` can be forced to always use this method by setting `bartlett_confint=False`.

References

1. Shumway, R. H., & Stoffer, D. S. (2025). Time Series Analysis and Its Applications: With R Examples. In *Springer Texts in Statistics*. Springer Nature Switzerland. [10.1007/978-3-031-70584-7](#)
2. Brockwell, P. J., & Davis, R. A. (1991). Time Series: Theory and Methods. In *Springer Series in Statistics*. Springer New York. [10.1007/978-1-4419-0320-4](#)
3. Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting. In *Springer Texts in Statistics*. Springer International Publishing. [10.1007/978-3-319-29854-2](#)
4. Wasserman, L. (2004). All of Statistics: A Concise Course in Statistical Inference. In *Springer Texts in Statistics*. Springer New York. [10.1007/978-0-387-21736-9](#)
5. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). Time Series Analysis. In *Wiley Series in Probability and Statistics*. Wiley. [10.1002/9781118619193](#)

Answers

Exercise

By the definition of bias, in order to prove that $\bar{x}$ is unbiased we must demonstrate that

$$\mathbb{E}[\bar{x}] = \mu_x. \quad (3.6.12)$$

We can demonstrate this as follows:

$$\begin{aligned} \mathbb{E}[\bar{x}] &= \mathbb{E}\left[\frac{1}{n} \sum_{i=0}^{n-1} x_i\right] \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{E}[x_i] \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \mu_x \\ &= \frac{1}{n} n \mu_x \\ &= \mu_x \end{aligned} \quad (3.6.13)$$

where we have exploited the linearity of expectation to go between the first and second lines.

4: METHODS TO ENFORCE STATIONARITY

4.1: Introduction

4.2: Time Series with Trends

4.3: Differencing and the Backshift Operator

4.4: Time Series with Seasonality

4.5: Increasing Variance and the Log Transform

This page titled [4: Methods to Enforce Stationarity](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

4.1: Introduction

In this chapter we will discuss reasons that a time series may be non-stationary and methods to convert it into a stationary series. We will cover three primary reasons a time series may be non-stationary:

1. Presence of a trend
2. Presence of a unit root
3. Presence of seasonal fluctuations

We will also cover three methods to convert a non-stationary time series into a stationary one, loosely paralleling the three causes above:

1. Detrending
2. Differencing
3. Seasonal adjustment

In the context of differencing we will also introduce a useful convention in the form of the *backshift operator*. We will conclude the chapter with a brief discussion of non-stationarity stemming from increasing variance and methods to correct it.

4.2: Time Series with Trends

Definition of Trend

The first primary cause of non-stationarity we will discuss is the presence of a *trend*. At its most basic level, a trend is simply a long-term increase or decrease in values.

Recall from [the first chapter](#) that time series can exhibit highly convincing spurious trends. We will cover some statistical methods to determine if a trend is real in later chapters, but none of these are fool-proof. As a data scientist, you must always combine your prior domain knowledge to assist in determining if it is reasonable to assume the existence of a trend.

Trends and Detrending

S&P 500 Trend

The following plot of the S&P 500 values appears to exhibit a stable overall trend.



Figure 4.2.1: Raw values of S&P 500 index for the 10-year period from January 2016 through January 2026 from [Federal Reserve Bank of St. Louis](#).

You can experiment with the values from Figure 4.2.1 by downloading the data using the following cell:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("https://fred.stlouisfed.org/graph/fredgraph.csv?
                 bgcolor=%23ebf3fb&chart_type=line&drp=0&fo=open%20sans&graph_bgcolor=%23ff
                 =false&line_style=solid&mark_type=none&mw=3&lw=3&ost=-99999&oet=99999&mma
                 index_col=0,
                 parse_dates=True,
                 )
# statsmodels is very particular regarding missing values, we will fill in missing values
df.ffill(inplace=True)
# examine a quick plot
plt.plot(df)
```

Trend Stationary Process

Given the apparent upward trend, it is reasonable to explore the possibility that the S&P 500 is *trend stationary*, i.e. the time series takes the form

$$x_t = \mu_t + y_t \quad (4.2.1)$$

where x_t is the observed time series, μ_t is a trend, and y_t is an underlying stationary time series. Eq. 4.2.1 suggests recovering y_t via the process of *detrending*

$$\hat{y}_t = x_t - \hat{\mu}_t. \quad (4.2.2)$$

What form should our estimate $\hat{\mu}_t$ take? Let's start with the most basic model, ordinary least squares (OLS)

$$\hat{\mu}_t = \beta_0 + \beta_1 t \quad (4.2.3)$$

where β_0 is the intercept and β_1 is the slope.

Regressing onto Time

Regressing onto Time In Eq. 4.2.3 we are regressing *onto time*. While this may be a natural assumption for many new data scientists, in reality regressing onto time is the exception in time series analysis, not the rule.

Detrending S&P 500

We will fit an OLS regression of the form in Eq. 4.2.3 using the following code:

```
import statsmodels.api as sm
import numpy as np

y = sp_500_df['SP500']
X = np.arange(len(sp_500_df))
# By default, statsmodels does not include an intercept term.
# We will add an intercept manually using add_constant.
X = sm.add_constant(X)

model = sm.OLS(y, X)
results = model.fit()

print(results.summary())
```

You should now see something like the following:

OLS Regression Results

```
=====
Dep. Variable: SP500 R-squared: 0.907
Model: OLS Adj. R-squared: 0.907
Method: Least Squares F-statistic: 2.543e+04
Date: Sun, 25 Jan 2026 Prob (F-statistic): 0.00
Time: 18:33:02 Log-Likelihood: -19337.
No. Observations: 2610 AIC: 3.868e+04
Df Residuals: 2608 BIC: 3.869e+04
Df Model: 1
Covariance Type: nonrobust
```

```
=====
coef std err t P>|t| [0.025 0.975]
-----
const 1644.6932 15.634 105.199 0.000 1614.037 1675.350
x1 1.6550 0.010 159.469 0.000 1.635 1.675
=====
Omnibus: 10.728 Durbin-Watson: 0.011
Prob(Omnibus): 0.005 Jarque-Bera (JB): 10.076
Skew: -0.117 Prob(JB): 0.00649
Kurtosis: 2.806 Cond. No. 3.01e+03
=====
```

Note that the R^2 is quite high, and both the intercept and slope are statistically significant.

Based on the results of our OLS, we will detrend the S&P 500 results using the formula

$$SP500_{detrended} = SP500 - 1645 - 1.66, t \quad (4.2.4)$$

where t consists of $[0, 1, \dots, n - 1]$ for n observations. We can apply this formula using the code

```
detrend_component = results.params["const"] + results.params["x1"] * np.arange(len(sp_500_df))
sp_500_df_detrended = sp_500_df["SP500"] - pd.Series(detrend_component, index=sp_500_df.index)
```

Detrended S&P 500 Index



Figure 4.2.2: Detrended values of S&P 500 index for the 10-year period from January 2016 through January 2026 from [Federal Reserve Bank of St. Louis](#) detrended using $SP500_{detrended} = SP500 - 1645 - 1.66, t$.

Exercise

While linear trends may be the most intuitive, we could use quadratic (or, rarely, even higher) order polynomials as well. Fit a trend line of the form

$$SP500 = \beta_0 + \beta_1, t + \beta_2, t^2 \quad (4.2.5)$$

using the code

```
import statsmodels.api as sm

y = sp_500_df['SP500']
X = np.arange(len(sp_500_df))
```

```
# Add quadratic term
X_quad = np.column_stack([X, X**2])

# Add intercept
X_quad = sm.add_constant(X_quad)

model = sm.OLS(y, X_quad)
results = model.fit()

print(results.summary())
```

Are the β 's significant? What does this tell you about use of a linear vs. a quadratic trend for the S&P 500? What other variations on a linear trend might you try as well?

Answer available at the end of this section.

Answers

Exercise

4.3: Differencing and the Backshift Operator

Random Walk

As discussed, a random walk [has a non-constant variance](#). Since $\gamma(0)$ is not constant, a random walk (even without drift) is not stationary. How might we go about isolating a stationary time series from a random walk? Detrending will remove any drift thereby stabilizing the mean, but it will not help with the non-constant variance. Instead, we rely on the process of *differencing*.

Differencing is defined as taking the difference between values. For example, the first difference of a random walk without drift is given by

$$z_t = x_t - x_{t-1}. \quad (4.3.1)$$

Note that (starting our time series with x_0 as some arbitrary constant)

$$\begin{aligned} z_1 &= x_1 - x_0 \\ &= (x_0 + w_1) - x_0 \\ &= w_1 \\ z_2 &= x_2 - x_1 \\ &= (x_1 + w_2) - x_1 \\ &= w_2 \\ &\dots \end{aligned} \quad (4.3.2)$$

Thus, the first difference of a random walk without drift is white noise and hence stationary. What about a random walk with drift? Starting again with an arbitrary x_0

$$\begin{aligned} z_1 &= x_1 - x_0 \\ &= (\delta + x_0 + w_1) - x_0 \\ &= \delta + w_1 \\ z_2 &= x_2 - x_1 \\ &= (\delta + x_1 + w_2) - x_1 \\ &= \delta + w_2 \end{aligned} \quad (4.3.3)$$

Since δ is constant, $\mathbb{E}[z_t] = \delta$ is also constant. Similarly, the addition of δ [does not change the covariance \$\gamma\(h\)\$](#) , so the first difference of a random walk with drift is also stationary.

Forecasting Random Walk

For a pure random walk, it is impossible to make a better statistical prediction than that the next value will be the same as the current value (plus any drift when applicable).

Difference Stationary Process

For pure random walk processes, Eqs 4.3.2 and 4.3.3 represent the limit of our analysis. However, very often we will encounter a *difference stationary process* defined again as

$$x_t = \mu_t + y_t \quad (4.3.4)$$

where in this case μ_t is a random walk process^[1]. Taking the first difference, we have

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= (\delta + \mu_{t-1} + w_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + y_t - y_{t-1} + w_t \end{aligned} \quad (4.3.5)$$

If y_t is stationary, the first difference $v_t = y_t - y_{t-1}$ must be as well:

$$\mathbb{E}[v_t] = \mathbb{E}[y_t - y_{t-1}] = \mathbb{E}[y_t] - \mathbb{E}[y_{t-1}] = \mu_y - \mu_y = 0 \quad (4.3.6)$$

and

$$\begin{aligned} \gamma_v(h) &= \text{Cov}(v_{t+h}, v_t) \\ &= \text{Cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\ &= 2\gamma_y(h) - \gamma_y(h-1) - \gamma_y(h+1) \end{aligned} \quad (4.3.7)$$

Exercise

Prove that the fact that v_t is stationary demonstrates that $z_t = x_t - x_{t-1}$ must be stationary as well.

Answer available at the end of this section.

Differencing to Differentiation

Finite Differences

Differencing is closely related to the discrete analog of differentiation. Assuming our time series p_t originates from the continuous process $p(t)$ with first derivative $\frac{d}{dt}p(t)$, the backward finite difference approximation to the derivative is

$$\frac{d}{dt}p(t) \approx \frac{p_t - p_{t-1}}{\Delta t}. \quad (4.3.8)$$

As we are assuming a constant sampling rate, we can treat Δt as 1 in the unit of our time steps (seconds, days, years, etc.). By the same logic, we can approximate the second derivative using the backward second-order difference

$$\begin{aligned} \frac{d^2}{dt^2}p(t) &\approx \frac{\frac{p_t - p_{t-1}}{\Delta t} - \frac{p_{t-1} - p_{t-2}}{\Delta t}}{\Delta t} \\ &= \frac{p_t - 2p_{t-1} + p_{t-2}}{(\Delta t)^2}. \end{aligned} \quad (4.3.9)$$

Where as before we can reduce to $p_t - 2p_{t-1} + p_{t-2}$ by treating Δt as 1[2].

Since the first derivative of a linear process is a constant, taking the first difference of a linear [trend stationary process](#) will result in a stationary process, though it may be very different than the series generated by detrending. By the same logic, taking the second difference will convert a quadratic trend into a stationary process.

Differencing Notation

Because differencing plays such a central role in time series analysis, there are specific notations designed to allow easier manipulation. The first difference is denoted by ∇

$$\nabla x_t \triangleq x_t - x_{t-1}. \quad (4.3.10)$$

Note the similarity to Eq. 4.3.10 . The second difference is denoted by ∇^2 and resembles Eq.4.3.11

$$\begin{aligned} \nabla^2 x_t &\triangleq \nabla x_t - \nabla x_{t-1} \\ &= (x_t - x_{t-1}) - (x_{t-1} - x_{t-2}) \\ &= x_t - 2x_{t-1} + x_{t-2}. \end{aligned} \quad (4.3.11)$$

Higher order differences can be defined analogously, but in practice we will almost never need to go beyond the second difference.

Backshift Operator

The *backshift operator*, $\mathbb{B}$, is a valuable tool in time series analysis. While at first it may seem like we are introducing notation for its own sake, over the course of the book we will see that the backshift operator is an elegant and powerful way to manipulate time series.

Lag vx. Backshift Terminology

Some texts use the term *lag operator* instead of backshift operator. `statsmodels` uses this term instead, which is denoted by L . The underlying principle is identical.

The backshift operator changes a member of a series to the preceding value, i.e.:

$$\mathbb{B}x_t = x_{t-1} \quad (4.3.12)$$

Similarly, $\mathbb{B}^2x_t = \mathbb{B}(\mathbb{B}x_t) = \mathbb{B}(x_{t-1}) = x_{t-2}$, and so on.

For completeness, we also define $\mathbb{B}$'s inverse $\mathbb{B}^{-1}$ as the forward-shift operator such that $\mathbb{B}\mathbb{B}^{-1} = \mathbb{B}^{-1}\mathbb{B} = 1$.

Differencing in Backshift Operator Notation

Combining Eq. 4.3.12 and Eq. 4.3.14, we can rewrite the first difference with unit time as

$$\begin{aligned} \nabla x_t &= x_t - x_{t-1} \\ &= x_t - \mathbb{B}x_t \\ &= (1 - \mathbb{B})x_t. \end{aligned} \quad (4.3.13)$$

The second difference can be expressed as

$$\begin{aligned} \nabla^2 x_t &= (1 - \mathbb{B})^2 x_t \\ &= (1 - 2\mathbb{B} + \mathbb{B}^2) x_t \\ &= x_t - 2x_{t-1} + x_{t-2}. \end{aligned} \quad (4.3.14)$$

Higher order differences d are defined as $(1 - \mathbb{B})^d$.

Exercise

Find $(1 - \mathbb{B}^{-1})^d$ for $d = 1, 2$.

Answer available at the end of this section.

Differencing vs. Detrending

While both detrending and differencing have their places, differencing is more commonly favored. Differencing has the major advantage that it is non-parametric, i.e. it does not rely on assuming any model (beyond a random walk) and parameters. In contrast, detrending assumes the existence of a linear (or higher-order) trend. Moreover, differencing a stationary series, while undesirable^[^3], will result in another stationary series, whereas detrending a stationary series will introduce the opposite trend.

^[^3]: When we cover ARMA processes we will see that unnecessary differencing adds an additional moving average (MA) term and frequently results in non-invertible models.

Differencing more naturally extends to higher derivatives, for example using the second difference for constant acceleration processes. Detrending via quadratic fit lines makes very strong assumptions about the underlying model and is prone to overfitting.

In contrast, detrending provides a readily interpretable model of the overall trend that can be communicated to clients or sponsors. Detrending allows a clean explanation along the lines of "After removing a steady three unit per month increase, we see that...", which is not possible with differencing. While differencing is more commonly the favored approach, ultimately the choice depends on your use case and intended audience.

Differencing S&P 500

Previously, we detrended the S&P 500 using a linear trend, resulting in Figure 4.3.1.

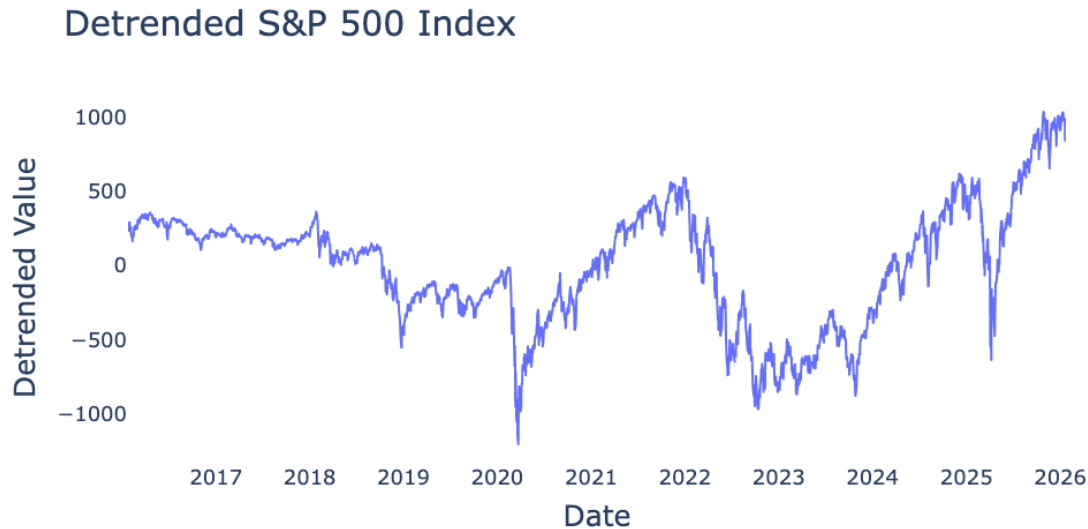


Figure 4.3.1: Detrended values of S&P 500 index for the 10-year period from January 2016 through January 2026 from [Federal Reserve Bank of St. Louis](#) detrended using $SP500_{detrended} = SP500 - 1645 - 1.66 \cdot t$.

What would happen if we instead take the first difference? `pandas` has a `diff` method accessed by `df.diff(periods=1)`. Running the code

```
# Pandas uses periods=1 by default.
sp_500_diff = sp_500_df.diff().dropna()

hello world
```

we can use the differenced time series to create {4.3.2.

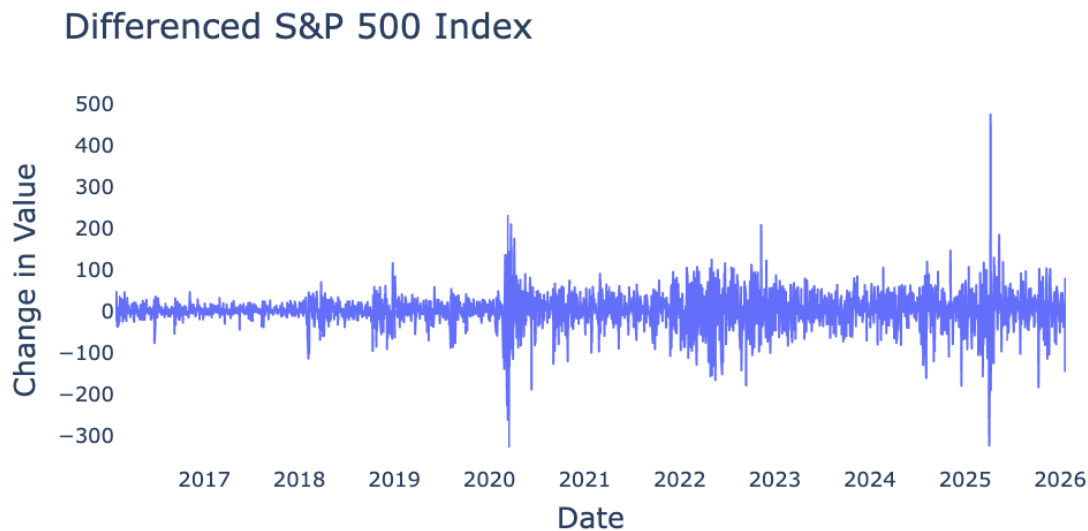


Figure 4.3.2: First difference of values of S&P 500 index for the 10-year period from January 2016 through January 2026 from [Federal Reserve Bank of St. Louis](#).

Figure 4.3.2 appears more likely to be stationary than Figure 4.3.1, though I would caution [against relying too heavily on pure visual inspection](#) for either trends or overall stationarity. The mean of the differenced S&P 500 is 1.93, fairly close to the linear regression slope of 1.66. Figure 4.3.2 does appear to exhibit *volatility clustering*, which we will learn in subsequent chapters can be understood via the ARCH family of models. Nevertheless, it is reasonable to conclude that the S&P 500 roughly follows a random walk with drift of $\delta \approx 1.9$, making its first difference stationary white noise.

Exercise

What would happen if you instead took the second difference? Run the code

```
sp_500_second_diff = sp_500_df.diff().diff().dropna()

-----
NameError                                Traceback (most recent call last)
Cell In[1], line 1
----> 1 sp_500_second_diff = sp_500_df.diff().diff().dropna()

NameError: name 'sp_500_df' is not defined
```

and plot the results. How does your figure compare to Figure 4.3.2

Autocorrelation of Random Walk

Before examining the autocorrelation, let's work out what we expect the autocovariance of a random walk to look like. We've [established that the autocovariance of a random walk](#) is given by

$$\begin{aligned}\gamma(s, t) &= \text{Cov} \left(\sum_{i=0}^s w_i, \sum_{j=0}^t w_j \right) \\ &= \sum_{i=0}^{\min(s, t)} \mathbb{V}(w_i) \\ &= (\min(s, t) + 1) \sigma_w^2\end{aligned}\tag{4.3.15}$$

Translating to autocorrelation is a bit tricky as [a random walk's variance is non-constant](#). Standard statistical packages still use the sample approximation for $\gamma(h)$ and $\rho(h)$

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}. \quad (4.3.16)$$

Even though Eq. 4.3.22 is only strictly valid for stationary time series, we can use it to approximate the autocorrelation for our case as well. Combining Eqs 4.3.21 and 4.3.22 for a time series of length t gives us

$$\begin{aligned} \hat{\rho}(h) &\approx \frac{\hat{\gamma}(s, t)}{\hat{\gamma}(t, t)} \\ &= \frac{(\min(s, t) + 1)\sigma_w^2}{(t + 1)\sigma_w^2} \\ &= \frac{(\min(s, t) + 1)}{t + 1}, \end{aligned} \quad (4.3.17)$$

which represents a linear decay as $h = t - s$ increases

Random Walk Autocorrelations

The sample autocorrelation of a random walk decays approximately linearly with respect to h .

Exercise

What would the autocovariance of the first difference of a random walk look like?

Answer available at the end of this section.

Do the autocorrelation functions agree with our assessment that the S&P 500 is a random walk? Let's look at them:

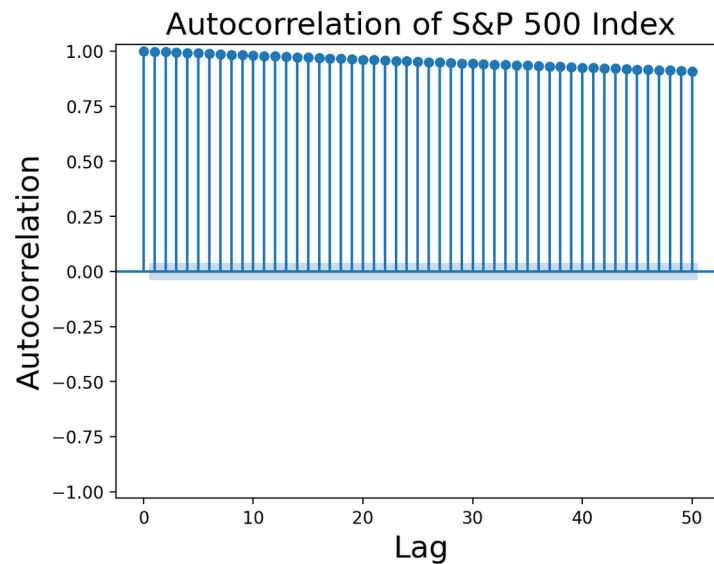


Figure 4.3.3: Autocorrelation function of S&P 500 values from [Federal Reserve Bank of St. Louis](#).

Figure 4.3.3 certainly looks like pure linear decay. What about the autocorrelation of the differenced series?

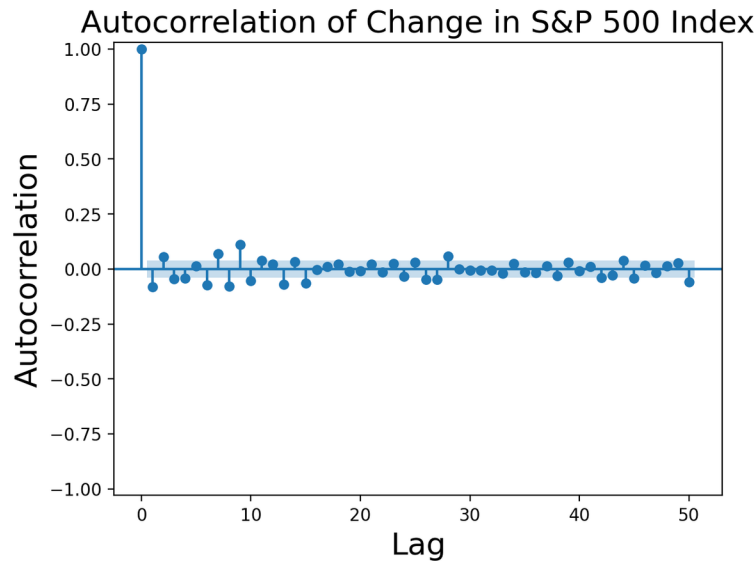


Figure 4.3.4: Autocorrelation function of first difference of S&P 500 values from [Federal Reserve Bank of St. Louis](#).

Multiple h values before $h \approx 15$ are significant beyond the level expected for pure white noise, indicating that our original process was probably not a pure random walk. Nevertheless, the autocorrelation values are small enough to say that a random walk is a reasonable first approximation to the S&P 500 returns.

Exercise

Plot the autocorrelation function of the detrended values from the previous section. Does the plot resemble Figure 4.3.3 or Figure 4.3.4 or is it distinct from both?

Footnotes

1. We will discover other scenarios of difference stationary process where $\mu\mu t$ is not a simple random walk, but is a process possessing a *unit root*. We will defer discussion of these cases until after we have covered ARMA processes. For the time being, you can think of the concept of unit root as referring to random walks.
2. Note that treating $\Delta t \Delta t$ as 1 will cause the values to agree, but the units will not be the same. Eq. (10) includes units of time–1time–1 and Eq. (11) includes units of time–2time–2.
3. When we cover ARMA processes we will see that unnecessary differencing adds an additional moving average (MA) term and frequently results in non-invertible models.

Answers

Exercise

Click to reveal solution **Solution:** We have already show that $\mathbb{E}[y_t - y_{t-1}] = 0$. Consequently,

$$\mathbb{E}[x_t - x_{t-1}] = \mathbb{E}[\delta + y_t - y_{t-1} + w_t] = \delta \quad (4.3.18)$$

which is clearly constant for all t . Similarly, (recognizing that the constant δ has no effect on the covariance) $\gamma_z(h)$ is given by

$$\begin{aligned} & \text{Cov}(y_{t+h} - y_{t+h-1} + w_{t+h}, y_t - y_{t-1} + w_t) \\ &= \text{Cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) + \text{Cov}(w_{t+h}, w_t) \\ &= 2\gamma_y(h) - \gamma_y(h-1) - \gamma_y(h+1) + \gamma_w(h) \\ &= 2\gamma_y(h) - \gamma_y(h-1) - \gamma_y(h+1) + \delta_{h,0}\sigma_w^2 \end{aligned} \quad (4.3.19)$$

where we have assumed iid w_t .

Exercise

Click to reveal solution **Solution:** This is most easily demonstrated by using a trial series x_t :

$$(1 - \mathbb{B}^{-1})x_t = x_t - x_{t+1} \quad (4.3.20)$$

For our purposes the forward and backward finite differences are identical because they only differ from one another by the index choice. Thus we can freely reindex with $s = t + 1$:

$$\begin{aligned}
 x_t - x_{t+1} &= x_{s-1} - x_s \\
 &= -(x_s - x_{s-1}) \\
 &= -(1 - \mathbb{B})x_s
 \end{aligned} \tag{4.3.21}$$

Thus $(1 - \mathbb{B}^{-1}) = -(1 - \mathbb{B})$.

While this suggests that $(1 - \mathbb{B}^{-1})^2 = (-(1 - \mathbb{B}))^2 = (1 - \mathbb{B})^2$, let's see if we can derive it explicitly. A similar manipulation provides:

$$\begin{aligned}
 (1 - \mathbb{B}^{-1})^2 x_t &= (1 - 2\mathbb{B}^{-1} + \mathbb{B}^{-2}) x_t \\
 &= x_t - 2x_{t+1} + x_{t+2} \\
 &= x_{t+2} - 2x_{t+1} + x_t
 \end{aligned} \tag{4.3.22}$$

reindexing with $v = t + 2$:

$$\begin{aligned}
 x_{t+2} - 2x_{t+1} + x_t &= x_v - 2x_{v-1} + x_{v-2} \\
 &= (1 - 2\mathbb{B} + \mathbb{B}^2) x_v \\
 &= (1 - \mathbb{B})^2 x_v
 \end{aligned} \tag{4.3.23}$$

Thus $(1 - \mathbb{B}^{-1})^2$ is effectively the same as $(1 - \mathbb{B})^2$.

Exercise

As demonstrated in Eqs 4.3.2 and 4.3.3, the first difference of a random walk is simply white noise with the addition of a drift term when applicable. iid white noise has an autocovariance of $\gamma(h) = \delta_{h,0}\sigma_w^2$. The autocovariance $\gamma(h)$, and consequently the autocorrelation $\rho(h)$, will be 0 for all $h > 0$.

4.4: Time Series with Seasonality

Seasonality Definition

One of the defining characteristics of time series is the presence of *seasonality*. Put simply, seasonality is a **regular and predictable** variation in a time series. Some examples of seasonal effects are:

- Higher temperatures during the day than the night
- Higher temperatures in July than January (in the northern hemisphere)
- Unemployment rates increasing in January after the holiday season
- Higher traffic density on weekdays than weekends

Note that multiple seasonal effects of different time scales may be layered together, for example higher temperatures during the day combining with higher temperatures during the summer.

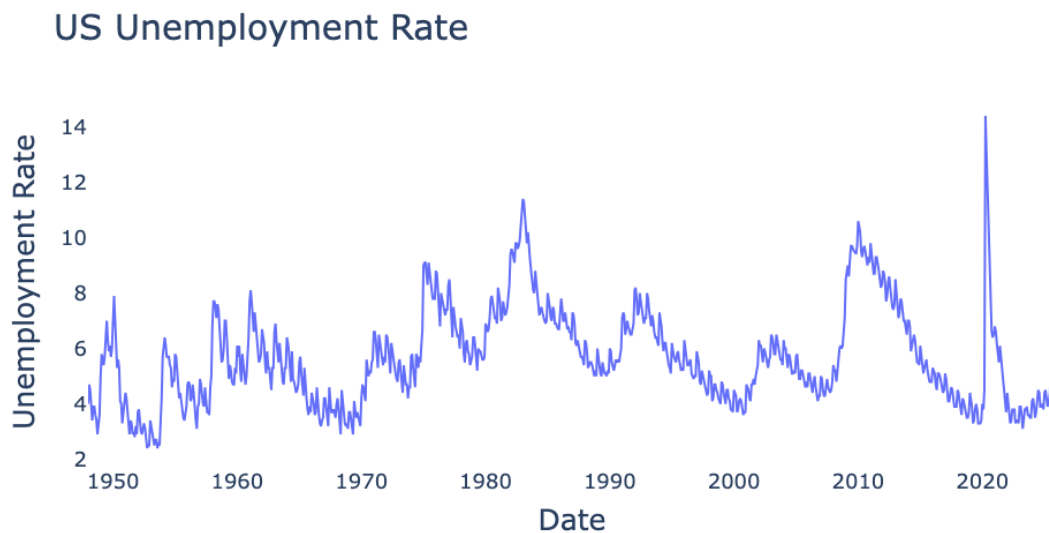


Figure 4.4.1: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) (not seasonally adjusted).

A time series with seasonal effects is not stationary because its mean is dependent on time. For example, Figure 4.4.1 is not stationary because the expectation value for unemployment is consistently higher in January.

Seasonal vs. Cyclic Changes

It is important to distinguish between *seasonal* effects and *cyclic* effects. They both describe periodic fluctuations in a time series, but are distinguished by regularity.

Cyclic Behavior and Stationarity

A time series exhibiting cycles can still be stationary as the irregular frequency of fluctuations means that no specific set of points has a higher expectation value than any other set.

Seasonal and cyclic effects are also sometimes distinguished by cycle length, in particular in econometrics. Behavior with a period of up to a year is often binned as seasonal, whereas behavior with periods longer than a year or two is binned as cyclic. It is important to understand that this is a common characteristic, not a definition. **Nothing inherently precludes the existence of seasonal effects with periods far longer than a year or cyclic effects with periods far shorter than a year.**

Simulated Time Series with Cyclic Behavior

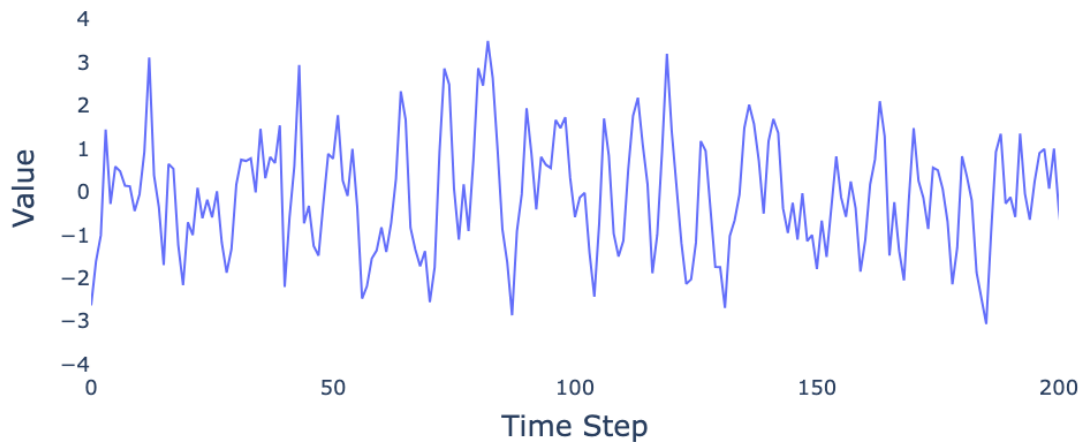


Figure 4.4.2: Simulated time series showing strong cyclical behavior with irregular repetition frequency.

Figure 4.4.2 shows a simulated time series with strong periodic behavior. However, since the periodicity is not constant, the underlying process is still stationary. In future chapters we will learn that this is a hallmark of autoregressive (AR) processes with complex roots.

Exercise 4.4.1

We have observed that Figure 4.4.1 exhibits seasonal effects with (potentially among others) yearly periodicity. Does it also exhibit cyclic behavior?

Answer available at the end of this section.

Distinguishing Seasonal and Cyclic Effects

So how do we know if a periodic effect is seasonal (and hence makes the time series non-stationary) or cyclic (and does not affect stationarity)? Some cycles such as Figure 4.4.2 or the multi-year cycles in Figure 4.4.1 are fairly easy to identify as not being seasonal. On the other hand, how do we verify that effects such as the yearly cycles in unemployment are seasonal and not cyclic? While there is no foolproof way to demonstrate that an effect is seasonal, there are a few tools we can use to help:

1. **High regularity:** When we have a long time series such as Figure 4.4.1 that covers over 75 years, the extreme regularity of January spikes is almost certainly a seasonal effect. However, if we only had 10 years or so to examine and/or had a much noisier time series, regularity would not be as strong an indicator.
2. **Domain expertise:** As with much of data science, we seek to incorporate priors from domain expertise. For example, it is quite reasonable to assume annual effects leading to an increase in unemployment in the same month every year. In contrast, had we seen an effect with, say, a seven month period, it would be less likely to have been a true seasonal effect.
3. **Autocorrelation:** The autocorrelation function for cyclic effects tends to display an exponential decay with sinusoidal behavior, such the autocorrelation function for Figure 4.4.2 shown in Figure 4.4.3:

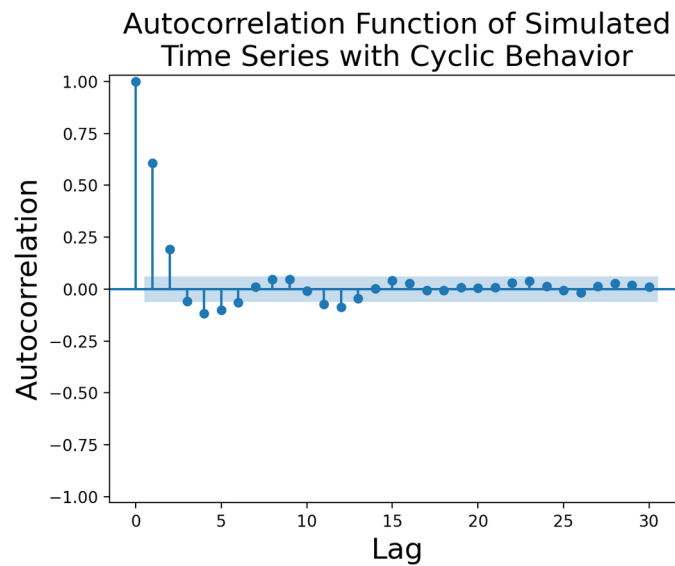


Figure 4.4.3: Autocorrelation function for simulated cyclic time series.

In contrast, the autocorrelation function for the first difference[1] of Figure 4.4.1 shows marked spikes with a twelve-month periodicity:

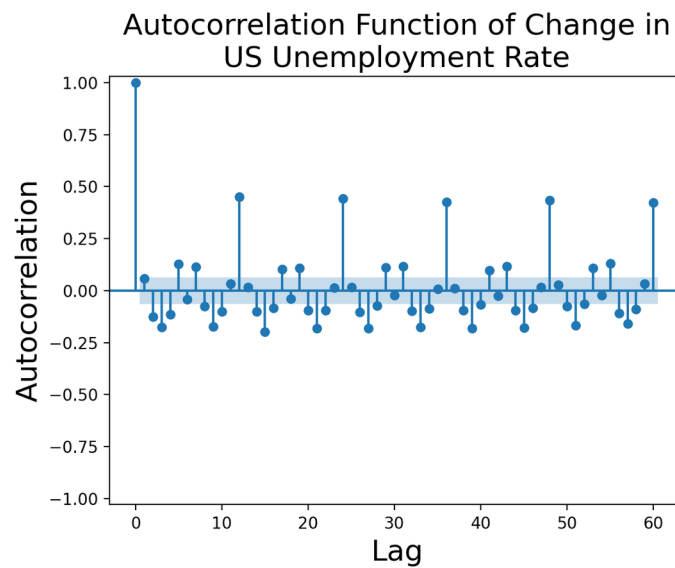


Figure 4.4.4: Autocorrelation function of change in US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) (not seasonally adjusted).

Be aware that for most time series you're likely to encounter in real life, the autocorrelation function is likely to be far more ambiguous than the examples above.

Exercise 4.4.1

In this problem you will examine an example of an autocorrelation function that may not be as easy to categorize as reflecting either seasonal or cyclic behavior. `statsmodels` has a preloaded time series dataset of historical sunspot activity, load it into a dataframe and plot the autocorrelation using the following code:

```
import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_acf

sunspots_df = sm.datasets.sunspots.load_pandas().data
sunspots_df.index = pd.Index(pd.date_range("1700", end="2009", freq="YE-DEC"))
sunspots_df.drop(columns="YEAR", inplace=True)
plot_acf(sunspots_df, lags=50)
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 import statsmodels.api as sm
      2 from statsmodels.graphics.tsaplots import plot_acf
      4 sunspots_df = sm.datasets.sunspots.load_pandas().data

ModuleNotFoundError: No module named 'statsmodels'
```

Does the plot of the autocorrelation function resemble either [ar2-sim-acf](#) or [unemployment-diff-acf](#)? What conclusions would you draw regarding the periodicity of sunspots?

Answer available at the end of this section.

In subsequent chapters, we cover additional methods for identifying seasonality using time series decomposition and Fourier analysis.

Methods for Addressing Seasonality

There are two major reasons we might want to remove seasonal effects from our time series:

1. Seasonal effects result in non-stationary time series, whereas methods such as the ARMA family require stationarity.
2. We may wish to identify long-term patterns and trends that can be obscured by normal seasonal fluctuations.

We will cover three major methods of removing seasonal effects.

Seasonal Moving Average

The first approach is arguably also the most intuitive, namely taking a *moving average* (also known as a *rolling average*) of our time series values with a window equal to the length of the "season." For example, if we are focused on daily temperatures but have hourly data, we might take a 24-hour moving average such that the first hourly value in our series is the mean of hours 0 – 23, the second value is the mean of hours 1 – 24, and so on.

Exercise 4.4.1

Would using a 48-hour window remove daily seasonality? What about a 36-hour window?

Answer available at the end of this section.

US Unemployment Rate with 12-Month Rolling Average

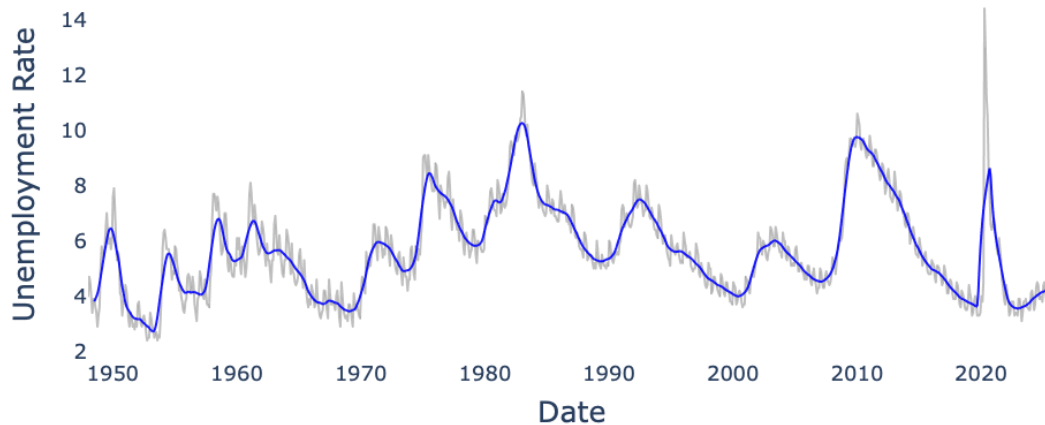


Figure 4.4.5: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) with raw data (gray) and a centered 12-month moving average (blue).

Seasonal Adjustment

Seasonal adjustment is a central technique in econometrics and is used heavily by organizations such as the Federal Reserve Bank. We will defer a complete discussion of seasonal adjustment until next chapter; for now we may think of it as defining a set seasonal effect S_t for each season and adjusting the time series to remove it. Seasonal adjustment is performed via one of two major methods:

1. **Additive:** By far the more common method, additive models assume our time series can be adjusted by subtracting S_t :

$$y_t = x_t - S_t, \quad (4.4.1)$$

where for example we might calculate that January is 20 degrees colder than average and July is 25 degrees hotter, giving

$$S_{January} = -20 \quad \dots \quad S_{July} = 25 \quad \dots \quad (4.4.2)$$

2. **Multiplicative:** Less commonly, we might instead assume our time series can be seasonally adjusted by dividing by S_t :

$$y_t = \frac{x_t}{S_t}. \quad (4.4.3)$$

Multiplicative seasonal adjustment most commonly applies in scenarios showing exponential growth such as stock or commodity prices (without adjustment for inflation). Relative prices for gasoline might consistently be a few percentage points higher in the summer than the winter, but the absolute price difference in 2026 is likely to be substantially greater than the absolute difference in 1976.

US Unemployment Rate (Seasonally Adjusted)

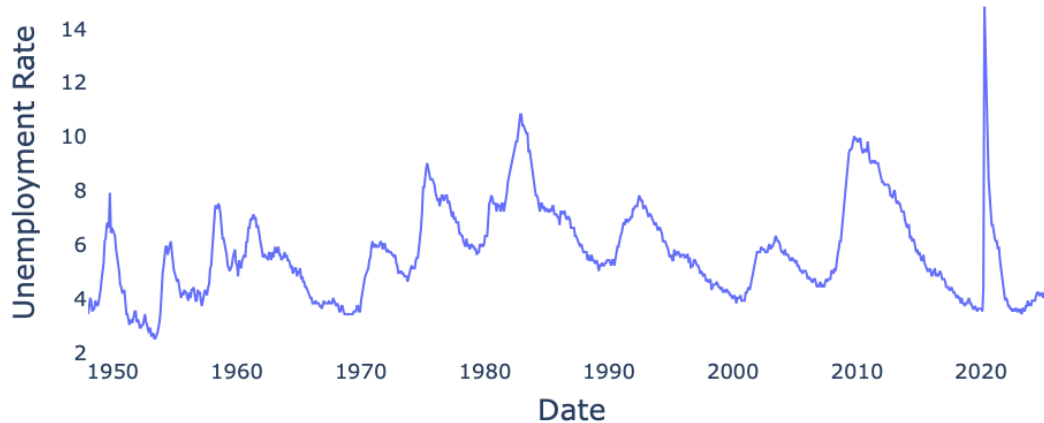


Figure 4.4.6: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) using Federal Reserve Bank's seasonal adjustment method.

Exercise

Several important seasonal adjustment algorithms only use additive adjustments. How might we apply such methods to a time series that exhibits exponential growth in seasonal fluctuations?

Answer available at the end of this section.

Seasonal Differencing

The final method we will use is *seasonal differencing*. We have seen how [the first difference removes random walk characteristics](#). By the same token, we may build a model consisting of a *seasonal* random walk in which the current season's value is determined by last season's value plus some noise w_t . For example, we might propose a monthly model

$$\begin{aligned} x_{\text{January},t} &= x_{\text{January},t-1} + w_{\text{January},t} \\ x_{\text{February},t} &= x_{\text{February},t-1} + w_{\text{February},t} \\ &\vdots \end{aligned} \quad (4.4.4)$$

where t refers to the current year and $t - 1$ to the previous year. Allowing that Eq. 4.4.6 is a valid representation of the underlying process generating our time series, the natural method to compensate for its effects is to take the *seasonal difference* ∇_s

$$\begin{aligned} z_t &\triangleq \nabla_s x_t \\ &= x_t - x_{t-s} \\ &= (1 - \mathbb{B}^s) x_t \end{aligned} \quad (4.4.5)$$

where s is the number of seasons per period; in the case of monthly data we have $s = 12$ giving us ∇_{12} .

∇_s vs. ∇^s

∇_s is the first difference at separation s and equals $1 - \mathbb{B}^s$, while ∇^s is the s th difference at separation 1 and equals $(1 - \mathbb{B})^s$.

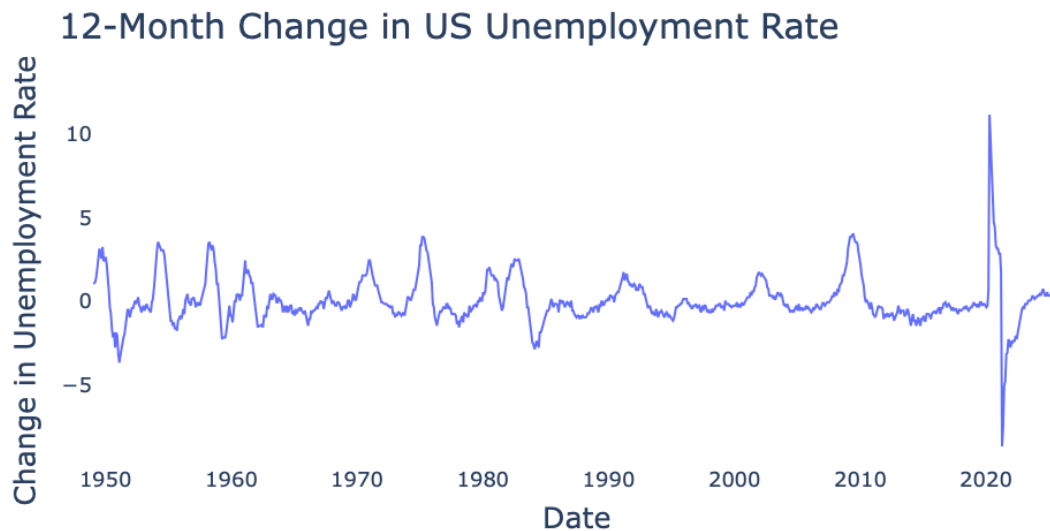


Figure 4.4.7: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) after applying 12-month differencing.

Figure 4.4.7 shows the 12-month change in the US unemployment rate. While it still demonstrates cycles varying from around five to ten years in duration, these [do not negate stationarity](#). Figure 4.4.8 shows the autocorrelation function for the series in Figure 4.4.7. The sinusoidally decaying nature of the autocorrelation is consistent with a [stationary cyclic time series](#).

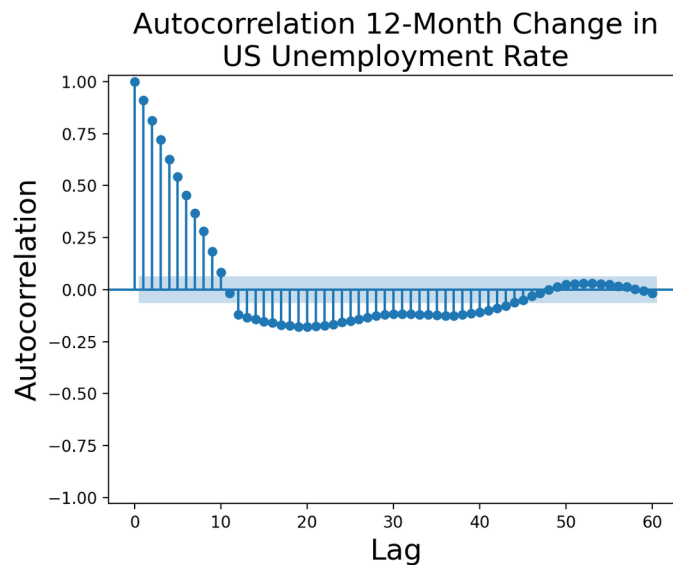


Figure 4.4.8: Autocorrelation function of US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) after applying 12-month differencing.

Seasonal Adjustment vs. Seasonal Differencing

As with [differencing versus detrending](#), seasonal adjustment and seasonal differencing both have valid use cases. While there is no objectively superior method, there are some considerations that might push you toward one over the other.

Arguments for seasonal adjustment:

1. Seasonal adjustment maintains the overall underlying value of a time series (compare Figure 4.4.6: and Figure 4.4.7), making it useful for understanding trends and cycles.
2. Seasonally adjusted plots are generally easier to interpret than plots of differenced time series, especially for an audience with less experience in data science.

3. More sophisticated seasonal adjustment methods allow variations in the seasonal pattern, unlike seasonal differencing which assumes a single seasonal pattern across an entire time series.
4. Similarly, more sophisticated seasonal adjustment methods can maintain the length of the original time series, whereas differencing shortens the sequence by s observations.

Arguments for seasonal differencing:

1. Differencing is less parametric than seasonal adjustment. Even if a given time series cannot be seasonally adjusted with adequate accuracy, seasonal differencing can still yield valid results.
2. Seasonal differencing is more likely to result in stationary time series than seasonal adjustment.
3. As a result of its better ability to create a stationary time series, seasonal differencing is the default method used in the *Seasonal ARIMA*, or SARIMA, family of models (though methods to use ARIMA with seasonal adjustment do exist).
4. Incorrectly applying seasonal differencing will not ruin a time series' stationary nature, whereas incorrectly applying seasonal adjustment to a stationary time series can backfire and introduce spurious seasonal effects.

Exploratory vs. Predictive Data Science

While not an absolute rule, seasonal adjustment is generally preferred in explanatory data science where our primary goal is to understand overall historical behavior. In contrast, seasonal differencing is preferred when performing analysis and forecasting as it tends to be a more reliable method of ensuring stationarity. :

Exercise 4.4.1

Find a few publicly reported time series such as those available from the Federal Reserve Bank of St. Louis. Do they use seasonal adjustment, seasonal differencing, or neither? What do you think motivates the choice?

Footnotes

1. We use the first difference rather than the original series because **differencing removes any trend and random walk characteristic**, allowing us to focus exclusively on the seasonal aspects of the series.

Answers

Exercise 4.4.1

Yes, in addition to a strong seasonality, 4.4.1 also exhibits cycles of increasing and decreasing unemployment rates with periodicities ranging from approximately 3 – 10 years in duration.

Exercise 4.4.1

Exercise 4.4.1

The key element is to have the same number of each season (hours of the day, in our example) for each averaged season. Every single 48-hour window will have two midnights, two 11 AMs, etc. Therefore, 24 or any integer multiple thereof will smooth out seasonal effects. In contrast, a 36-hour window beginning in the morning will have two days but only one night, while one starting in the evening will have two nights but only one day. Consequently, a 36-hour window—or any window that is not a multiple of 24—will not compensate correctly for seasonal effects.

Exercise

We can always convert division to subtraction by taking the logarithm. If the initial time series obeys

$$y_t = \frac{x_t}{S_t} \quad (4.4.6)$$

we can define a new time series $z_t \triangleq \log(y_t)$ which now follows the relation

$$z_t = \log(x_t) - \log(S_t). \quad (4.4.7)$$

4.5: Increasing Variance and the Log Transform

Most of the time series we've seen so far have had constant variance across time[1]. In some scenarios, particularly those involving exponential growth, variance may increase concomitantly with the value. For example, as overall air travel increases, the absolute difference between busier and less busy seasons will also increase (even if the relative difference is constant). Such time series are non-stationary for at least two (and often three) reasons:

[1]: A random walk is an important exception. Random walks [have a linearly increasing variance with respect to time](#).

1. The mean is non-constant over time.
2. The variance $\gamma(0)$ is non-constant over time.
3. Some examples such as air travel may also be non-stationary due to seasonal effects.

Apple Mac Sales

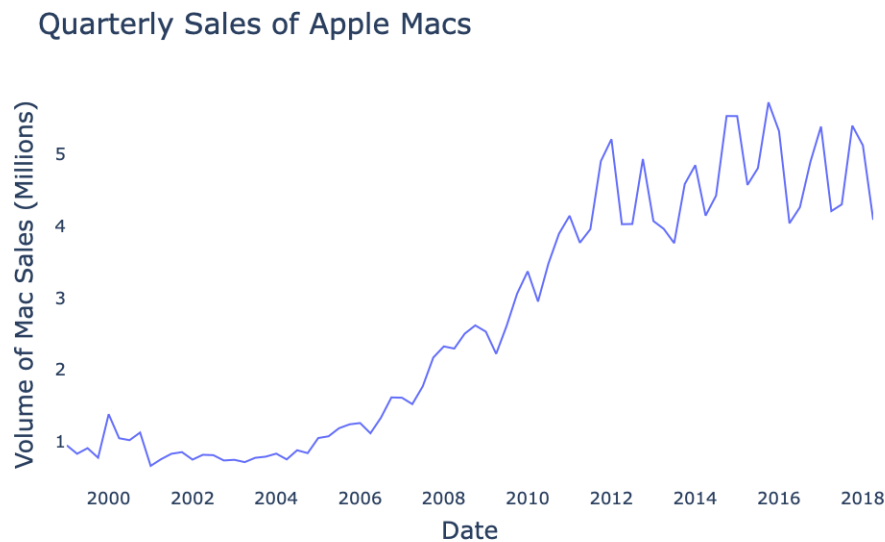


Figure 4.5.1: Quarterly sales figures for Apple Mac computers in millions from [GitHub Apple Data Repository](#).

Figure 4.5.1 appears to be non-stationary for all three reasons listed above. The mean shows a steady upward trend and the volumes appear to exhibit a seasonal effect with higher sales in the autumn and winter and lower sales in the spring and summer. As the size of the seasonal effect appears to depend on the overall trend, the variance also increases with time.

Let's first address the increasing variance by replacing the raw values with their logarithms:

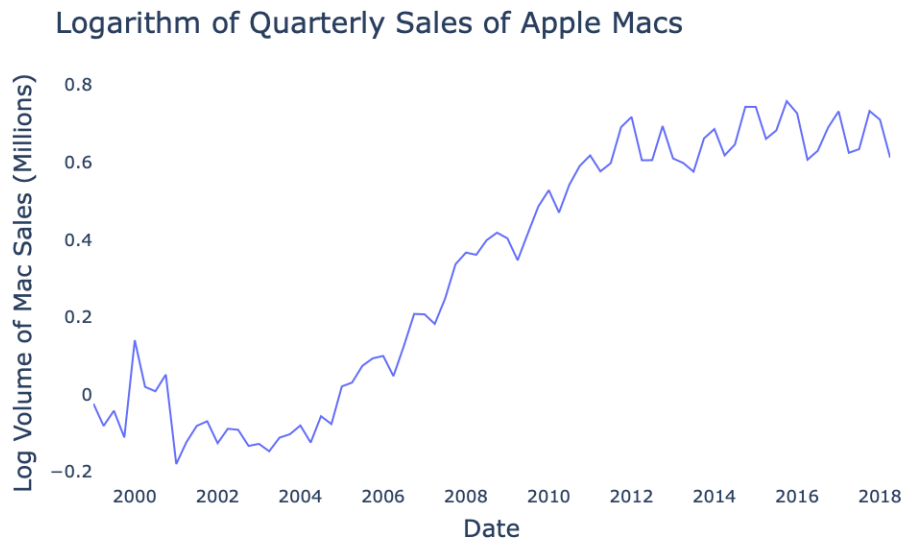


Figure 4.5.2: Logarithm of quarterly sales figures for Apple Mac computers in millions from [GitHub Apple Data Repository](#).

Figure 4.5.2 is beginning to look better—the variance appears to be roughly constant across time. We can address both the trend and seasonality by taking the first and fourth differences of the log values, i.e.

$$z_t \triangleq \nabla_4, \nabla(\log(x_t)) \quad (4.5.1)$$

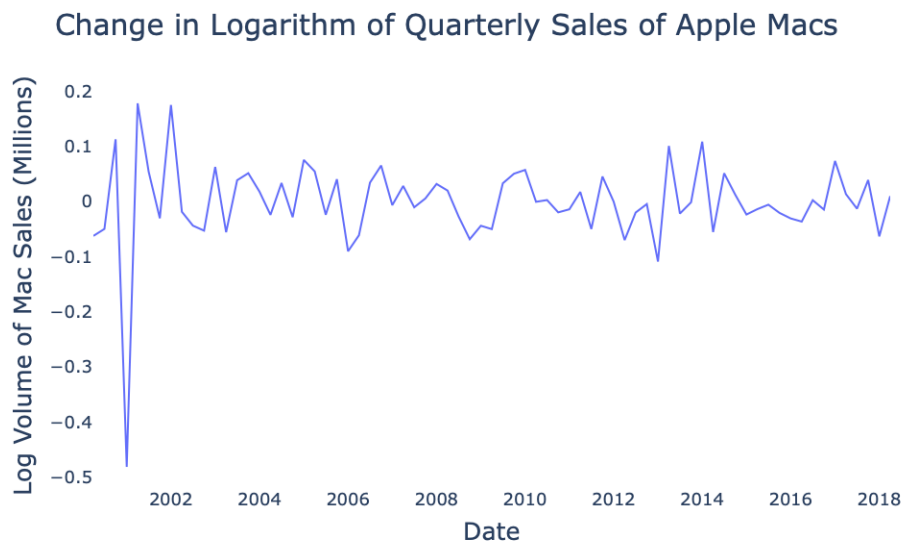


Figure 4.5.3: First and fourth differences of logarithm of quarterly sales figures for Apple Mac computers in millions from [GitHub Apple Data Repository](#).

Figure 4.5.3 certainly looks stationary upon visual inspection. After we've covered autoregressive processes we will revisit this example with statistical tests designed to help determine stationarity.

Exercise

In Eq. 4.5.1 we applied the first difference at lag 1 followed by the first difference at lag 4. Would it have made a difference if we'd switched the order and instead used $\nabla, \nabla_4(\log(x_t))$?

Answer available at the end of this section.

Log Difference and Returns

In Eq. 4.5.1 we used the first (and fourth) difference of the logarithm of the sales figures. This concept closely relates to the *return*, a unitless quantity defined as the difference in value between successive time steps divided by the previous time step, i.e.

$$r_t \triangleq \frac{x_t - x_{t-1}}{x_{t-1}}. \quad (4.5.2)$$

To see this relation, let us rewrite Eq. 4.5.5 as

$$\frac{x_t}{x_{t-1}} = 1 + r_t. \quad (4.5.3)$$

We now have

$$\nabla \log(x_t) = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right) = \log(1 + r_t) \approx r_t \quad (4.5.4)$$

where we have used the Taylor expansion

$$\log(1 + r_t) = r_t - \frac{r_t^2}{2} + \dots \quad (4.5.5)$$

Under the assumption of small r_t , quadratic and higher order terms will be negligible. [log-returns](#) demonstrates that the two functions are essentially identical for returns with an absolute value less than about 20. For this reason, $\nabla \log(x_t)$ is often used interchangeably with the return defined in Eq. 4.5.5.

Comparison of r and $\log(1+r)$

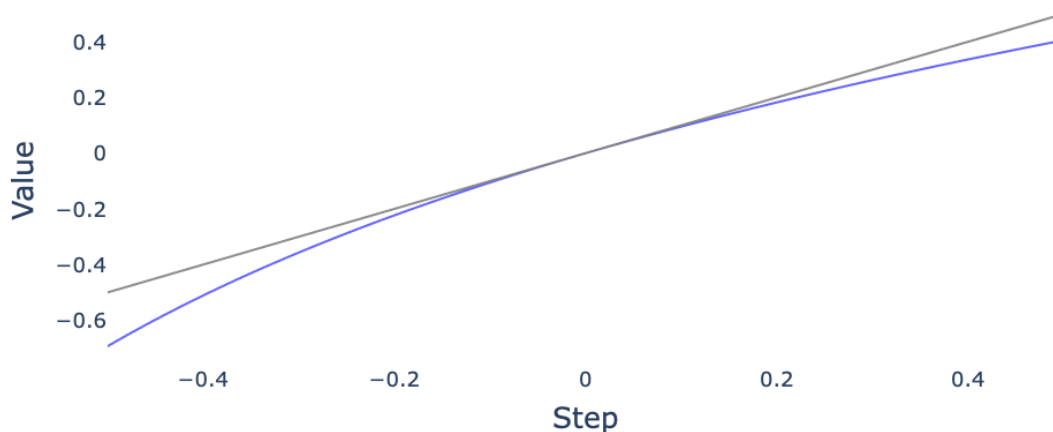


Figure 4.5.4: Linear slope $y = r_t$ (gray) and logarithmic curve $y = \log(1 + r_t)$ (blue) for $r_t \in [-0.5, 0.5]$.

Answers

Exercise

Click to reveal solution **Solution:** Recall the the difference operator can be written in backshift notation as

$$\nabla = 1 - \mathbb{B} \quad (4.5.6)$$

and

$$\nabla_4 = 1 - \mathbb{B}^4. \quad (4.5.7)$$

Multiplying these out, we get

$$\nabla_4 \nabla = (1 - \mathbb{B}^4)(1 - \mathbb{B}) = 1 - \mathbb{B} - \mathbb{B}^4 + \mathbb{B}^5 = (1 - \mathbb{B})(1 - \mathbb{B}^4) = \nabla \nabla_4 \quad (4.5.8)$$

Thus, we see that the two operators commute in this scenario.

This page titled [4.5: Increasing Variance and the Log Transform](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

5: TIME SERIES DECOMPOSITION

5.1: Introduction

5.2: Classical Decomposition

5.3: Local Regression

5.4: STL Decomposition

This page titled [5: Time Series Decomposition](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

5.1: Introduction

In this chapter we will cover the extremely useful tool *time series decomposition*. Decomposition is primarily used for time series analysis, though we will see methods in subsequent chapters to use decomposition as part of a larger forecasting algorithm. As a practicing data scientist you will often be tasked with helping your client or employer understand historical activity with an eye towards identifying both long-term trends and seasonal effects. Time series decomposition—in particular the algorithm known as *STL decomposition*—should generally be among your first tools to answer such questions.

We will begin this chapter by discussing classical decomposition methods developed in the 1920s and 1930s. We cover classical decomposition not because it is the best method currently available (it isn't), but to lay the foundations to understand more sophisticated methodologies.

We will then take a detour to discuss local regression before combining classical decomposition with local regression in the STL decomposition algorithm first published in 1990. STL decomposition is arguably the single most versatile and flexible algorithm for time series decomposition, and is my recommended method for most scenarios.

Note that we will not be covering methods such as the X-11 family. This family of models was first introduced by the US Census Bureau in 1967, with regular updates and refinements since then^[1]. While valuable tools for their intended purposes, the X-11 family is fairly specific to the needs of government statisticians and is not widely adopted elsewhere. STL decomposition is both more flexible in terms of seasonal effects and more widely adopted, and should generally be favored for scenarios other than specific government statistics cases.

Footnotes

1. The newest version is [X-13ARIMA-SEATS](#) introduced in 2020 and available for download free of charge at the link provided.

5.2: Classical Decomposition

Components of Time Series

Many time series contain **trends**, **seasonal effects**, and/or **cycles**. *Time series decomposition*, or simply *decomposition*, is defined as methods to isolate each component's contribution to an overall time series. Since cycles lack a specific length it is difficult to draw a clear distinction between an upward trend followed by a downward trend and a cycle. Consequently, we will generally fold cycles into the trend for the purposes of decomposition.

Note

Trends and Cycles in Decomposition Most time series decomposition methods combine trends and cycles into a single component known as the "trend-cycle," or often merely as the "trend."

Before diving into the details, let's get a feeling for what a time series decomposition might look like.

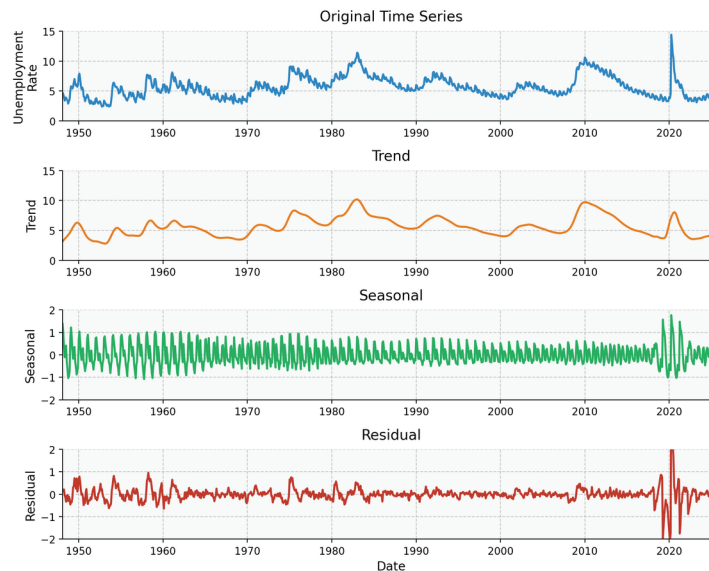


Figure 5.2.1: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) with STL decomposition consisting of original series (top), trends and cycles (second from top), seasonal contribution (third from top), and residual variation not explained (bottom).

We'll delve into the algorithms used to create figures like Figure 5.2.1: below and in subsequent chapters. For now, let's take a moment to understand the components depicted in Figure 5.2.1.

1. T_t : The *trend component*, consisting of both trends and cycles in the data.
2. S_t : The *seasonal component*, consisting of regular periodicity at a specified seasonal length. Note that most decomposition methods only allow a single seasonality.
3. R_t : The *residual* (sometimes also referred to as the *remainder*), consisting of any variation not accounted for by T_t or S_t . The residual can be thought of as roughly equivalent to noise w_t , though I'd caution against taking this analogy too far. R_t is only strictly equivalent to noise if T_t and S_t are a perfect model for the underlying process.

Additive and Multiplicative Models

Most commonly, we assume an additive decomposition in the form of

$$x_t = S_t + T_t + R_t. \quad (5.2.1)$$

In some cases, we may also assume a multiplicative decomposition

$$x_t = S_t \times T_t \times R_t, \quad (5.2.2)$$

(mult-decomp) which is more applicable, for example, in forecasting quarterly sales of a product experiencing compound growth in demand such as [Mac sales in the previous chapter](#). Note that we can always recast Eq. ??? into the form of Eq. 5.2.1 by using a log transform

$$\log(x_t) = \log(S_t) + \log(T_t) + \log(R_t). \quad (5.2.3)$$

While other models are possible (see @Hyndman_2004), for example

$$x_t = (S_t + T_t) \times R_t, \quad (5.2.4)$$

such models are almost never used in practice if for no other reason than that the difficulty in interpreting the output can easily defeat the original purpose of time series decomposition.

Classical Decomposition Algorithm

Obsolete Methodology The classical time series decomposition models are essentially obsolete and have been supplanted by new methods such as STL decomposition. We study the classical method here primarily as a basis for understanding STL decomposition.

Defining Seasonal Length

Before using the algorithm we must first define our seasonal period m . Common values of m include $m = 7$ for weekly data, $m = 12$ for monthly data, and $m = 24$ for hourly data. Note that we cannot define multiple seasonal effects in a given decomposition.

Recall that a moving average with [the length of a season \(or any multiple thereof\)](#) will remove seasonal effects, allowing us to focus on longer term cycles and trends. For seasonal effects with odd numbers of time steps per season such as a $m = 7$ for weekly seasonality, we employ a simple centered moving average of length m . Thus, for data starting on a Sunday, the first Wednesday will be replaced with the average of the first Sunday-Saturday, the first Thursday will be replaced by the average of the first Monday-Sunday, and so on. Note that we do lose the first and last $\frac{m-1}{2}$ observations for odd m or—as we will see shortly—the first and last $\frac{m}{2}$ for even m . This is not an issue for long time series in which the number of observations $n \gg m$, but does pose a challenge for shorter time series.

7-Day Moving Average for Weeks 1 – n

Original Day of Week (Week Number)	Value in Smoothed Series
Sunday (1)	—
Monday (1)	—
Tuesday (1)	—
Wednesday (1)	$\frac{1}{7}(\text{Sunday (1)} + \text{Monday (1)} + \text{Tuesday (1)} + \text{Wednesday (1)} + \text{Thursday (1)} + \text{Friday (1)} + \text{Saturday (1)})$
Thursday (1)	$\frac{1}{7}(\text{Monday (1)} + \text{Tuesday (1)} + \text{Wednesday (1)} + \text{Thursday (1)} + \text{Friday (1)} + \text{Saturday (1)} + \text{Sunday (2)})$
Friday (1)	$\frac{1}{7}(\text{Tuesday (1)} + \text{Wednesday (1)} + \text{Thursday (1)} + \text{Friday (1)} + \text{Saturday (1)} + \text{Sunday (2)} + \text{Monday (2)})$
⋮	⋮
Monday (n)	$\frac{1}{7}(\text{Friday (} n - 1) + \text{Saturday (} n - 1) + \text{Sunday (} n) + \text{Monday (} n) + \text{Tuesday (} n) + \text{Wednesday (} n) + \text{Thursday (} n))$
Tuesday (n)	$\frac{1}{7}(\text{Saturday (} n - 1) + \text{Sunday (} n) + \text{Monday (} n) + \text{Tuesday (} n) + \text{Wednesday (} n) + \text{Thursday (} n) + \text{Friday (} n))$

Original Day of Week (Week Number)	Value in Smoothed Series
Wednesday (n)	$\frac{1}{7}(\text{Sunday } (n) + \text{Monday } (n) + \text{Tuesday } (n) + \text{Wednesday } (n) + \text{Thursday } (n) + \text{Friday } (n) + \text{Saturday } (n))$
Thursday (n)	—
Friday (n)	—
Saturday (n)	—
...	

For even values of m such as quarterly data, it is impossible to center a moving average of length m . As a compromise, we use a $2 \times m$ moving average. A $2 \times m$ moving average is a moving average of moving averages, for example, a 2×4 moving average is defined as:

$$\begin{aligned}\hat{T}_t &= \frac{1}{2} \left(\frac{1}{4} [x_{t-2} + x_{t-1} + x_t + x_{t+1}] + \frac{1}{4} [x_{t-1} + x_t + x_{t+1} + x_{t+2}] \right) \\ &= \frac{1}{8} x_{t-2} + \frac{1}{4} x_{t-1} + \frac{1}{4} x_t + \frac{1}{4} x_{t+1} + \frac{1}{8} x_{t+2}\end{aligned}\quad (5.2.5)$$

A $2 \times m$ moving average allows us to center each observation by using an odd $m + 1$ length window, weighting the first and last observation by $\frac{1}{2m}$ and the others by $\frac{1}{m}$.

Quarterly Moving Average for Years 1 – n

Original Quarter (Year Number)	Value in Smoothed Series
First (1)	—
Second (1)	—
Third (1)	$\frac{1}{8}\text{First } (1) + \frac{1}{4}\text{Second } (1) + \frac{1}{4}\text{Third } (1) + \frac{1}{4}\text{Fourth } (1) + \frac{1}{8}\text{First } (2)$
Fourth (1)	$\frac{1}{8}\text{Second } (1) + \frac{1}{4}\text{Third } (1) + \frac{1}{4}\text{Fourth } (1) + \frac{1}{4}\text{First } (2) + \frac{1}{8}\text{Second } (2)$
⋮	⋮
First (n)	$\frac{1}{8}\text{Third } (n - 1) + \frac{1}{4}\text{Fourth } (n - 1) + \frac{1}{4}\text{First } (n) + \frac{1}{4}\text{Second } (n) + \frac{1}{8}\text{Third } (n)$
Second (n)	$\frac{1}{8}\text{Fourth } (n - 1) + \frac{1}{4}\text{First } (n) + \frac{1}{4}\text{Second } (n) + \frac{1}{4}\text{Third } (n) + \frac{1}{8}\text{Fourth } (n)$
Third (n)	—
Fourth (n)	—
...	

Exercise

Why do we use a $2 \times m$ window for a series with an even m instead of an $m + 1$ window?

Answer available at the end of this section.

The smoothed series is used as our estimate of the series trend T_t , denoted as $\hat{T}_t$ to emphasize that it is an estimate to the true T_t . From here on, classical decomposition subdivides into additive and multiplicative methods.

Exercise 5.2.1

While pandas can easily produce a centered moving average for data with odd seasonality, using a $2 \times m$ window requires a second step. In this exercise we will use the `seasonal_decompose` module from `statsmodels` to automatically apply the correct moving average window based on m .

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

employment_df = pd.read_csv("https://fred.stlouisfed.org/graph/fredgraph.csv?bgcolor=%23
employment_df.index = pd.date_range(start=employment_df.index[0], end=employment_df.inde
employment_df.columns = ["Unemployment"]
employment_df.ffill(inplace=True)

# Decompose the time series to access the trend component. Be sure to specify the correc
classic_decomp = seasonal_decompose(employment_df["Unemployment"], model="additive", per

# Get the trend component and plot it.
employment_df["Moving Average"] = classic_decomp.trend
plt.plot(employment_df)
```

Does your plot look like its removed seasonality? What do you observe regarding trends and cycles?

Additive Method

Having obtained our estimate of the trend $\hat{T}_t$, we are now ready to estimate the seasonality. We first detrend the series by subtracting $\hat{T}_t$

$$x_{t,detrend} = x_t - \hat{T}_t \quad (5.2.6)$$

We then obtain our estimate of the seasonal component $\hat{S}_t$ by taking the average of each detrended value of that season (e.g. the average detrended Thursday or average detrended May). The individual seasonal components are adjusted for an overall baseline of zero, i.e.

$$\sum_t \hat{S}_t \approx 0. \quad (5.2.7)$$

For example, our baseline temperature might be $60^\circ F$, with the summer being 30° higher and the winter 30° lower.

Exercise

This exercise continues with the results of classical decomposition from the previous problem.

```
# Get the seasonal component and plot it.
plt.plot(classic_decomp.seasonal)
```

Does the seasonality appear to be roughly centered around 0? Is it regular, or does the seasonal contribution itself change over time?

Having obtained our estimates $\hat{T}_t$ and $\hat{S}_t$, the estimated residual $\hat{R}_t$ is simply what's left after subtracting the estimated trend and seasonality

$$\hat{R}_t = x_t - \hat{T}_t - \hat{S}_t. \quad (5.2.8)$$

Exercise 5.2.1

This exercise continues with the results of classical decomposition from the previous problems.

```
# Get the residual component and plot it.
plt.plot(classic_decomp.resid)
```

Does the residual appear to be roughly white noise centered around 0? Can you see the effects of the COVID pandemic?

Multiplicative Methods

In certain scenarios, in particular with time series [exhibiting exponential growth](#), a multiplicative decomposition may be more appropriate. While we can always apply additive decomposition to the logarithm of the original series, classical decomposition is capable of directly estimating the multiplicative decomposition (Eq. ???).

As with the additive case, we begin by obtaining our estimated trend $\hat{T}_t$ via a moving average. We detrend the series via *division*

$$x_{t,detrend} = \frac{x_t}{\hat{T}_t} \quad (5.2.9)$$

We obtain our estimated seasonal component $\hat{S}_t$ by averaging each detrended season as before. For multiplicative decomposition the individual seasonal components are adjusted for a baseline of one, i.e.

$$\prod_t \hat{S}_t \approx 1, \quad (5.2.10)$$

For example the first quarter might have a seasonal component value of 125 while the third quarter might have a component value of 80.

The residual is simply what remains after dividing out the estimated trend and seasonal effects

$$\hat{R}_t = \frac{x_t}{\hat{T}_t \hat{S}_t}. \quad (5.2.11)$$

Plotting Decomposition Components

Above, we accessed the components of the classical decomposition and plotted them individually. `statsmodels` does have a nice feature to automatically plot the full decomposition invoked (using the variable names from above) as follows:

```
classic_decomp.plot()
```

This should provide you with a plot like

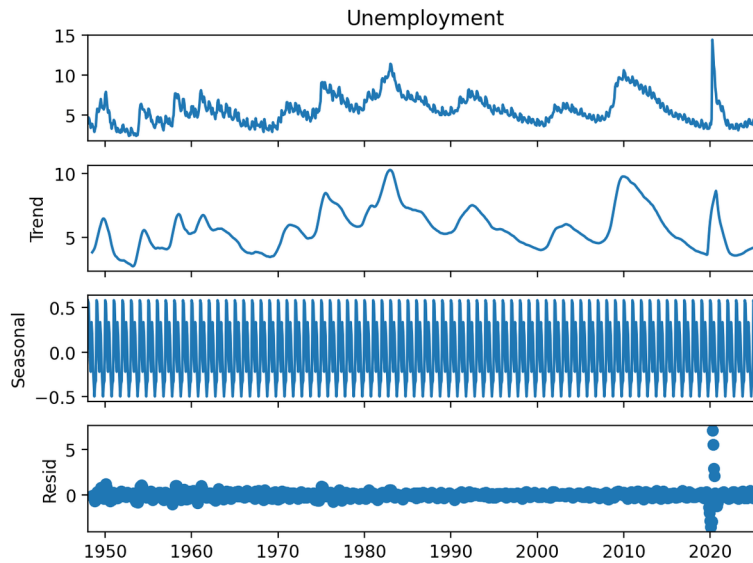


Figure 5.2.2: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) with additive classical decomposition consisting of original series (top), trends and cycles (second from top), seasonal contribution (third from top), and residual variation not explained (bottom).

Magnitude of Contributions By default, `statsmodels` will determine the y-axis scales individually for each component. Whenever comparing the contribution of different elements it is crucial to pay close attention to the y-axes in order to draw valid inference.

Assessing Decomposition Quality

A natural question to ask is how to assess the quality of a time series decomposition. Assuming that Eq. 5.2.1 or Eq. ??? is a reasonable model of the underlying data, a better decomposition can be expected to result in an estimated residual $\hat{R}_t$ far smaller than the trend and seasonal components. While visual inspection of a plot such as [classic-decomp-unemployment](#) or [stl-decomp-unemployment](#) (paying careful attention to the scale of the y-axes!) is a valuable starting point, in certain scenarios we may wish to have a more quantitative metric.

@Hyndman_2026 recommend the [following formula](#) for estimating the strength of the trend

$$F_T = \max \left(0, 1 - \frac{\mathbb{V}(\hat{R}_t)}{\mathbb{V}(\hat{T}_t + \hat{R}_t)} \right), \quad (5.2.12)$$

the logic being that for a time series with a strong trend component, the variance of the residual component alone should be much smaller than the variance of the combined trend and residual components. Thus, for data with a strong trend that has been well isolated by $\hat{T}_t$, F_T will be close to 1. F_T will be close to 0 for data with a minimal trend and/or a poorly isolated $\hat{T}_t$.

[^1]: The quantity $\frac{\mathbb{V}(\hat{R}_t)}{\mathbb{V}(\hat{T}_t + \hat{R}_t)}$ could conceivably be greater than 1 if $\hat{T}_t$ and $\hat{R}_t$ have a [strong negative covariance](#). Eq. 5.2.12 ensures that $F_T \in [0, 1]$ by use of the maximum of 0 or the computed value.

Computing F_T from the unemployment data above using the formula

```
max(0, 1 - (np.var(classic_decomp.resid)/np.var(classic_decomp.trend+classic_decomp.resid)))
```

gives a value of 0.928, indicating a strong trend component that has been well isolated by $\hat{T}_t$.

The strength of the seasonal component F_S is computed in the same manner as F_T using the equation

$$F_S = \max \left(0, 1 - \frac{\mathbb{V}(\hat{R}_t)}{\mathbb{V}(\hat{S}_t + \hat{R}_t)} \right). \quad (5.2.13)$$

Calculating F_S using the code

```
max(0, 1-(np.var(classic_decomp.resid)/np.var(classic_decomp.seasonal +classic_decomp.resi
```

gives a value of 0.379. While not terrible, compared to F_T this value indicates either (1) a weaker seasonal component (unlikely based on visual examination of the original data), or that (2) our decomposition has not done as good a job isolating the seasonal component, resulting in $\hat{S}_t$ having a weaker contribution to the estimated decomposition shown in Figure 5.2.2.

Drawbacks to Classical Decomposition

While not a terrible method, classical decomposition does have a number of drawbacks that make algorithms such as STL preferable. The major issues (highlighted in the derivations and exercises above) are as follows:

1. Use of a moving average removes the first and last $\frac{m}{2}$ (m even) or $\frac{m-1}{2}$ (m odd) observations from the trend (and consequently also the seasonal and residual components). For short time series this will result in sacrificing a fair amount of potential insight into the time series' behavior.
2. Averaging each season across the data assumes there is *exactly* one seasonal value per season that is constant across the entire time series, rather than allowing the seasonal contribution itself to be a function of time.
3. Classical decomposition is not robust to brief but extreme fluctuations such as the COVID pandemic and its effect on unemployment rates.

For these reasons, most statistics texts recommend against using the classical method for anything beyond a baseline to compare to more advanced methods. In the coming sections we will build on our analysis of classical decomposition to understand STL decomposition and how it addresses the issues above.

Answers

Exercise

Click to reveal solution **Solution:** As discussed, our objective in using a moving average with a window the same length as the seasonality is to remove seasonal effects. Even though a $2 \times m$ window has length $m + 1$, it's actually composed of two moving averages of length m as shown in Eq. ???, each of which alone has already smoothed out the seasonality (the second layer of averaging is only necessary to center our observation). In contrast, an $m + 1$ window is not a multiple of m , and hence will not remove seasonal effects.

This page titled 5.2: Classical Decomposition is shared under a CC BY-NC-SA 4.0 license and was authored, remixed, and/or curated by Charles Forgy via source content that was edited to the style and standards of the LibreTexts platform.

5.3: Local Regression

Kernel Smoothing

We've encountered data smoothing in the form of a moving average. Figure 5.3.1 and Figure 5.3.2 present a synthetic white noise process and the same process after smoothing with a moving average of length 10.

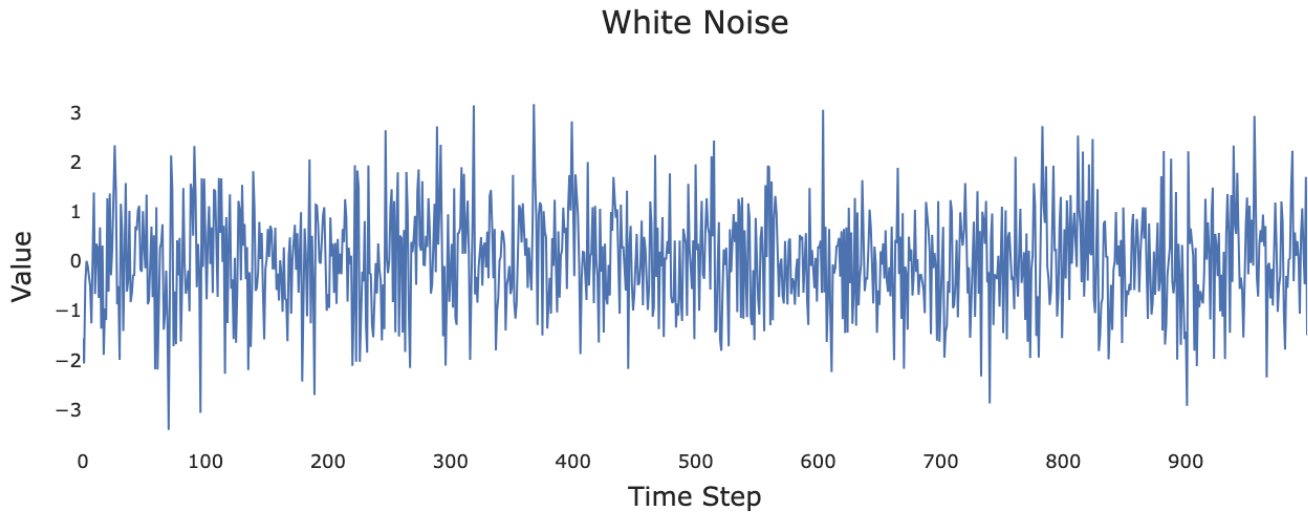


Figure 5.3.1: Simulated white noise process.

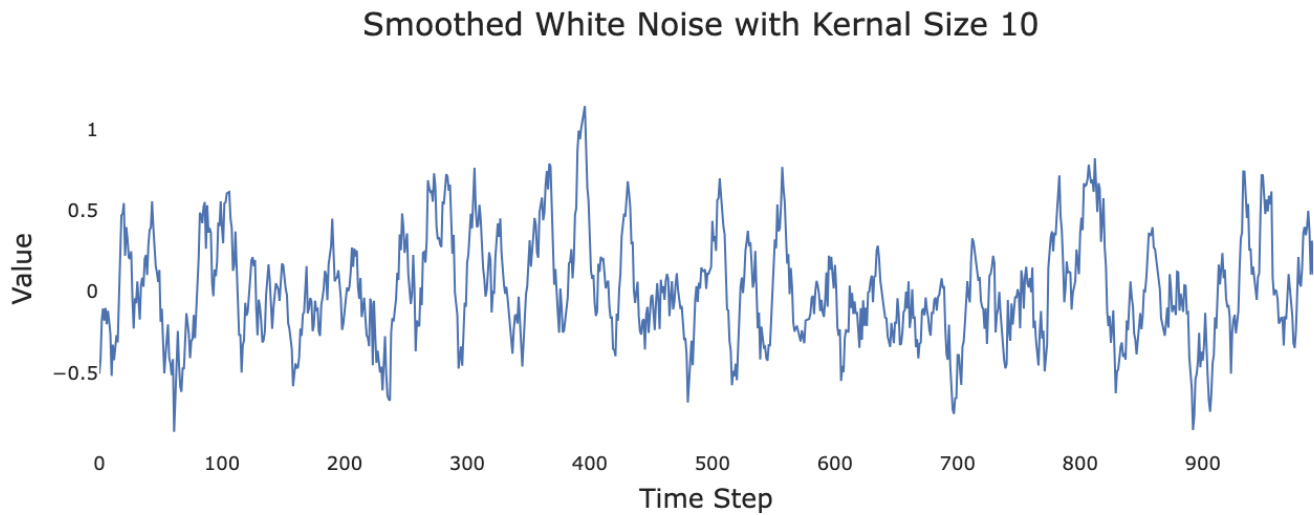


Figure 5.3.2: Smoothed version of white noise after applying the filter $\left[\frac{1}{10}, \frac{1}{10}, \dots, \frac{1}{10}\right]$.

We can think of the smoothing process depicted in Figure 5.3.2: as consisting of sliding a *kernel* (also known as a *filter*) across the data. In this case, the kernel is defined as

$$\left[\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}\right] \quad (5.3.1)$$

which we can envision as being slid across the data. At each point, we take the inner (or dot) product of the data with the kernel.

For the kernel used in Eq. 5.3.1 this gives the unweighted average of 10 points. In later chapters we will learn that this can be thought of as a *convolution* in the frequency domain.

The box (or boxcar) kernel used in Figure 5.3.2 is only one of many possible choices. We will learn in future chapters that a box filter has substantial shortcomings in the frequency domain that are mitigated (though not eliminated) by use of tapered kernels, such as kernels based on Gaussians, the cosine function, or the tricube function introduced in the next section.

Tricube Function

For this chapter, the most important kernel method is generated from the *tricube function* W :

$$W(x_0, x_i) \triangleq \begin{cases} (1 - |u|^3)^3 & |u| \leq 1 \\ 0 & |u| > 1 \end{cases} \quad (5.3.2)$$

where u is the distance between the focal point x_0 and the point x_i calculated as $u = \frac{x_0 - x_i}{\max_j (|x_0 - x_j|)}$ [1]. The maximum j is determined by an adjustable parameter giving the length of the kernel.

[1]: The quantity $\max_j (|x_0 - x_j|)$ is often referred to as the *bandwidth*. We will reserve the term bandwidth for use in the context of the frequency domain.

Normalization of Kernel

Note that in Eq. 5.3.2 we have not normalized the overall window to a value of 1 as we have done with moving average kernels. The reason for this distinction will become clear once we discuss the tricube function's use in the context of linear or polynomial regression below.

For example, for $\max_j (|x_0 - x_j|) = 5$, the tricube function W generates a sharply peaked kernel with $2j - 1 = 9$ non-zero values out of $2j + 1 = 11$ total values

$$[0, 0.116, 0.482, 0.820, 0.976, 1, 0.976, 0.820, 0.482, 0.116, 0] \quad (5.3.3)$$

with all points outside the window weighted as 0 (note that for the first and last elements $u = \pm \frac{5}{5}$, resulting in zero for W). In actual implementation we would simply use a kernel of length $2j - 1$ (9, in our example) instead of using a kernel of length $2j + 1$ (11, in our example) with zeros in the first and last positions. We explicitly included the zeros in Eq. 5.3.3 to emphasize that the kernel falls to zero at $|u| = 1$.

Weighting Points Near the Edges

In order to avoid shortening the time series (as occurs with moving averages), any point less than $\max_j (|x_0 - x_j|)$ from either end has an asymmetric selection of points in order to maintain the same kernel size at the expense of a larger value for $\max_j (|x_0 - x_j|)$. For example, continuing with $\max_j (|x_0 - x_j|) = 5$ from before, the asymmetric window used for the first point in the series will now have $\max_j (|x_0 - x_j|) = 9$ with values given as:

$$[1, 0.996, 0.967, 0.893, 0.759, 0.569, 0.348, 0.148, 0.0264, 0] \quad (5.3.4)$$

LOWESS Algorithm

Local regression is perhaps best thought of as a hybrid between kernel smoothing and linear regression (or more broadly polynomial regression). The most common family of methods is LOWESS (locally weighted scatterplot smoothing), a type of LOESS (locally estimated scatterplot smoothing) [2]. The LOWESS algorithm is generally credited as having been formally introduced in Cleveland (1979) and Cleveland (1981) (though much of the concept had been proposed independently by Savitzky & Golay (1964) about fifteen years beforehand).

[2]: Some sources such as NIST treat the terms LOESS and LOWESS as being interchangeable. Cleveland himself in appears to have used the term LOESS as a multivariate generalization of the previous univariate LOWESS.

LOWESS Demo

Before diving into the underlying algorithm, let's get a feeling for how LOWESS behaves. The following interactive figure plots the function

$$y = \sin(x) + \frac{1}{2}\sin(3x) + w \quad (5.3.5)$$

for some noise w in gray dots. For reference, the original (presumably unknown) clean signal without noise that we are attempting to reconstruct is denoted by a gray dashed line. The toggle button **LOWESS CURVE** turns on or off the LOWESS curve fitted through the points. The button **LOCAL FITS** toggles on and off a selection of the individual local regression lines (more on those soon). Finally, the slider **SPAN (f)** tells the algorithm what fraction of the overall data to consider for each local fit. Note that at 1.00, the line is almost a flat linear regression line with large local regression lines. As you shrink the fraction down, the LOWESS fit becomes more jagged with smaller local regression lines (denoting that the local line is calculated with a smaller portion of the data).

If the above fails to render correctly in your browser you can also open the demo as a new browser window using the [Open Demo in a New Tab](#) button at the top of the frame. Note that you may need to enable popups for this to work.

Exercise 5.3.1

What fraction in **SPAN (f)** does the best job reconstructing the original signal?

Local Regression

The LOWESS algorithm begins by dividing the data into overlapping windows. The size of each window is governed by the parameter f , the fraction of the total observations n . In `statsmodels.nonparametric.smoothers_lowess.lowess`, the fraction is set with the argument `frac` (default `frac=2.0/3.0`). In the demo above, f can be smoothly varied from 0.05 to 1. Once f is chosen, we create n windows of length $\lceil f \times n \rceil$ centered at each data point where $\lceil f \times n \rceil$ is the *ceiling function* indicating the smallest integer greater than or equal to $f \times n$. For each window about a given central point (denoted as x_{center}), we calculate a weighted linear regression defined as

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{\lceil f \times n \rceil} W(x_{center}, x_i) (y_i - \beta_0(x_{center}) - \beta_1(x_{center})x_i)^2 \quad (5.3.6)$$

where $W(x_{center}, x_i)$ is a weight function, most commonly the **tricube function**, that weights the contribution of each x_i based on its distance from x_{center} and where we have expressed the coefficients as $\beta_0(x_{center})$ and $\beta_1(x_{center})$ to emphasize that distinct β 's are calculated for each data point. For points closer than $\frac{f \times n}{2}$ to either end, the window no longer places the point in the center of the window, and uses the modification to W [discussed above](#).

LOWESS can be extended from linear regression to polynomial regression using expressions such as

$$\min_{\beta_0, \beta_1, \beta_2} \sum_{i=1}^{\lceil f \times n \rceil} W(x_{center}, x_i) (y_i - \beta_0(x_{center}) - \beta_1(x_{center})x_i - \beta_2(x_{center})(x_i)^2)^2, \quad (5.3.7)$$

though we will not need to go beyond Eq. 5.3.6 for our purposes.

Having calculated n distinct linear regressions, we use each linear regression about a given x_{center} to calculate the $\hat{y}_{center}$ at that point, leaving us with n $\hat{y}$ values (one per data point). The $\hat{y}$ points are then connected into a single curve—though note in the demo above that the fitted curve becomes notably more jagged as f is decreased.

Robust Smoothing

In the [demo above](#), we calculated the LOWESS curve solely by implementing Eq. 5.3.6. This method works well for data that has relatively few outliers. [Cleveland \(1979\)](#) describes an optional second stage of smoothing using the *bisquare function* $B(x)$ defined as

$$B(x) \triangleq \begin{cases} (1-x^2)^2 & |x| \leq 1 \\ 0 & |x| > 1 \end{cases} \quad (5.3.8)$$

where as in Eq. 5.3.2 $B(x)$ falls to zero at $x = \pm 1$. After the initial LOWESS curve calculation using Eq. 5.3.6, the algorithm then goes through a predetermined number of iterations of the following procedure:

1. Calculate the residual between the predicted and observed points

$$e_i = y_i - \hat{y}_i. \quad (5.3.9)$$

2. Determine new weightings δ for each point by first calculating the median s of the $|e_i|$'s and then applying the formula

$$\delta_k \triangleq B\left(\frac{e_k}{6s}\right). \quad (5.3.10)$$

3. Recalculate the local regressions using Eq. 5.3.6 using $B \times W$ instead of W .

Cleveland refers to this algorithm as "robust locally weighted regression." In `statsmodels`, the number of iterations of the above procedure is controlled by the argument `it` (default `it=3`). Setting `it=0` will result in the algorithm only using Eq. 5.3.6 in its original form.

In general, robust smoothing is primarily required for datasets with large outliers, such as those produced by Cauchy distributed noise. Figure 5.3.3 demonstrates LOWESS smoothing with different numbers of iterations on a synthetic dataset with Cauchy distributed noise. Note in particular that for 0 iterations (non-robust LOWESS) the fitted LOWESS line "chases" the outliers out of the plot. Once we reach 3 iterations, the LOWESS line almost perfectly captures the original signal, after which additional iterations have essentially no effect.

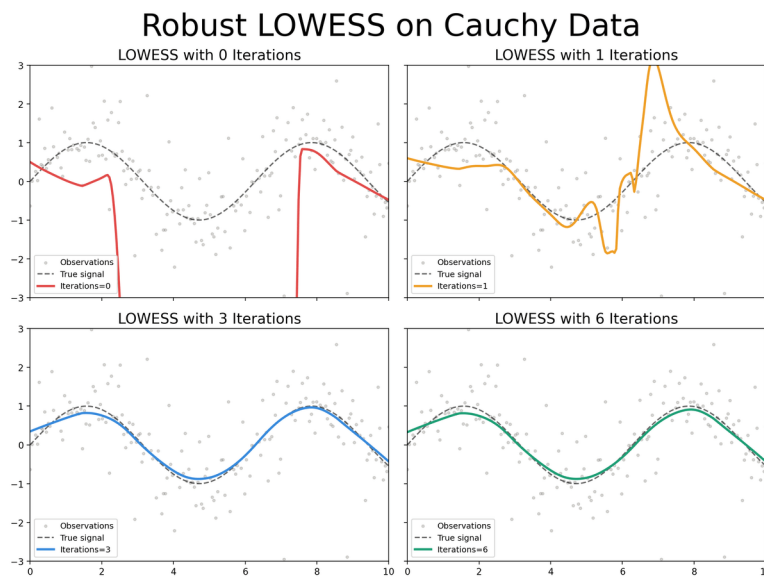


Figure 5.3.4: LOWESS regression using 0, 1, 3, and 6 iterations of robust smoothing on synthetic dataset defined as $y = \sin(x) + w$ with w Cauchy distributed.

Exercise 5.3.1

Generate your own version of **Figure 3** using the following code:

```
import numpy as np
from statsmodels.nonparametric.smoothers_lowess import lowess
import matplotlib.pyplot as plt

x_cauchy = np.linspace(0, 10, 200)
y_cauchy = np.sin(x_cauchy) + np.random.standard_cauchy(size=len(x_cauchy)) * 0.3

# It is advisable to run the following in a new cell to avoid generating new datasets fo
smoothed_data = lowess(exog=x_cauchy, endog=y_cauchy, frac=0.2, it=3, return_sorted=True)
```

```
x_lowess, y_lowess = smoothed_data[:, 0], smoothed_data[:, 1]
lowess_fig, lowess_ax = plt.subplots()
lowess_ax.scatter(x_cauchy, y_cauchy, s=8, alpha=0.3, color="#888780", label="Observatio
lowess_ax.plot(x_lowess, y_lowess, label="LOWESS Curve")
lowess_ax.plot(x_cauchy, np.sin(x_cauchy), "k--", lw=1.5, alpha=0.6, label="True signal"
lowess_ax.set_ylim(-3,3)
lowess_ax.set_xlim(0,10)
plt.legend()
plt.show()
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[1], line 2
      1 import numpy as np
----> 2 from statsmodels.nonparametric.smoothers_lowess import lowess
      3 import matplotlib.pyplot as plt
      5 x_cauchy = np.linspace(0, 10, 200)

ModuleNotFoundError: No module named 'statsmodels'
```

It is advisable to run the following in a new cell to avoid generating new datasets for each new plot.

What happens to the fitted curve as you adjust `it`? What if you adjust `frac`? Do iterations make as large a difference for Gaussian or uniform noise distributions?

Footnotes

1. The quantity $\max_j (|x_0 - x_j|)$ is often referred to as the bandwidth. We will reserve the term bandwidth for use in the context of the frequency domain.
2. Some sources such as NIST treat the terms LOESS and LOWESS as being interchangeable. Cleveland himself in Cleveland & Devlin (1988) appears to have used the term LOESS as a multivariate generalization of the previous univariate LOWESS.

References

1. Cleveland, W. S. (1981). LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *The American Statistician*, 35(1), 54. [10.2307/2683591](#)
2. Savitzky, Abraham., & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8), 1627–1639. [10.1021/ac60214a047](#)
3. Cleveland, W. S., & Devlin, S. J. (1988). Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association*, 83(403), 596–610. [10.1080/01621459.1988.10478639](#)

5.4: STL Decomposition

Having explained [classical approaches to time series decomposition](#) and [local regression](#), we are now in a position to understand *Seasonal-Trend decomposition using LOESS*[1] (STL). @Cleveland_1990 from Bell Labs and the University of Michigan introduced STL to compensate for the [shortcomings of classical decomposition](#). As we shall see, STL grows naturally out of augmenting classical decomposition with LOESS. Using LOESS gives the flexibility to calculate trends for all data points, unlike moving averages which truncate the first and final values. Furthermore, unlike the [X-11 family](#) which only handles quarterly or monthly seasonalities, STL can handle any number of seasons per cycle.

[1]: @Cleveland_1990 write that the "L" stands for LOESS. We will maintain this terminology, though for the purposes of this book we will only need the [narrower univariate LOWESS](#) algorithm rather than the multivariate LOESS.

STL Algorithm

Log Transformation

Unlike with classical decomposition, STL only works with additive trend and seasonal components. Data with a multiplicative trend must be [log transformed to use STL](#). In practice, this is rarely a major impediment to using STL, though applying a log transform does cause interpretability to go slightly down.

STL Loops

The STL algorithm runs a nested loop of functions until the algorithm reaches a set stopping point determined by the user's choice of both inner and outer loops. We will defer discussion of deseasonalization for the moment to first understand the overall procedure. Letting n_i be the number of passes in the inner loop and n_o be the number in the outer loop, the algorithm is as follows:

1. Initialize [robustness weights](#) for LOESS to 1 and the trend to 0.
 - 1.1 Detrend the series using the most recently calculated trend (set at 0 initially).
 - 1.2 Divide the data into m subseries (e.g. the first subseries consisting of each January, the second subseries consisting of each February, etc.). Generate m LOESS (weighted by robustness) curves *independently* for each subseries to smooth the detrended data.
 - 1.3 Calculate the seasonal component ([next section](#)) and deseasonalize the data.
 - 1.4 Apply another round of (weighted) LOESS on the deseasonalized data to determine the trend to be used as the starting point for the next iteration.
 - 1.5 Return to step 1.1 until completing n_i passes.
2. Calculate the residuals. If $n_o > 0$, calculate robustness weights to quantify each residual's degree of extremity for the LOESS algorithm (i.e. to what extent it should be considered an outlier) and return to step 1.1 until completing n_o iterations, otherwise stop.

The fact that LOESS can output smoothed data with the [same length as the original data](#) is crucial for steps 1.2 and 1.4. With a standard moving average, each pass of a length m filter would delete m (m even) or $m - 1$ (m odd) observations; running multiple loops on a small dataset would quickly whittle it down to nothing. It is only because we use LOESS that we can freely run the inner loop for multiple iterations. Additionally, note that the use of m distinct subseries in step 1.2 allows the magnitude of each seasonal component to vary over time.

By default, `statsmodels` does not use robust fitting (i.e. $n_o = 0$) and uses 5 inner loops ($n_i = 5$). This can be changed by setting `robust=True`, which by default will use $n_i = 2$ and $n_o = 15$. n_i and n_o can also be set manually in the `fit` method by calling

```
STL(data).fit(inner_iter=n_i, outer_iter=n_o).
```

Note

Unlike with classical decomposition, `statsmodels.tsa.seasonal.STL` requires explicitly calling `.fit()` to create an STL decomposition.

STL Deseasonalization

In step 1.3 we referenced deseasonalization without explaining how this is performed. A major advantage of STL over classical decomposition is that it allows the magnitudes of the seasonal component to vary over time instead of locking us into a single set of seasonal magnitudes for the entire series. The algorithm used in 1.3 functions as follows:

1. The deseasonalization step starts by creating an intermediate series C_i consisting of the results of step 1.2 reassembled from the m constituent subseries and temporarily padded at both ends.
2. L_i is calculated from C_i by smoothing C_i with a three step "low-pass" filter^[^2] consisting of two sequential unweighted moving averages of length m , followed by an unweighted moving average of length 3 (which together remove the padding from step 1), and finishing with LOESS.
3. Seasonal components S_i are calculated as $S_i = C_i - L_i$ (using the non-padded C_i). Subtracting L_i removes low-frequency long-term components from the seasonality, leaving solely the short-term seasonal effects in S_i . S_i is then subtracted from the series for this iteration's deseasonalized series.

[^2]: We will explore why this filter is referred to as "low-pass" in the chapter covering the frequency domain.

An important advantage of STL is that, while we do still need to specify an m value for the seasonal length, we are no longer constrained to have a single average $\hat{S}_t$ per season as we were in [classical decomposition](#). Instead, the use of LOESS in step 2 allows for variations in the magnitudes of the seasonal component (though not its length) over the course of the data.

Plotting STL Decomposition Components

Plotting the default `statsmodels` implementation of STL results in [{ref}stl-decomp-unemployment-non-robust](#):

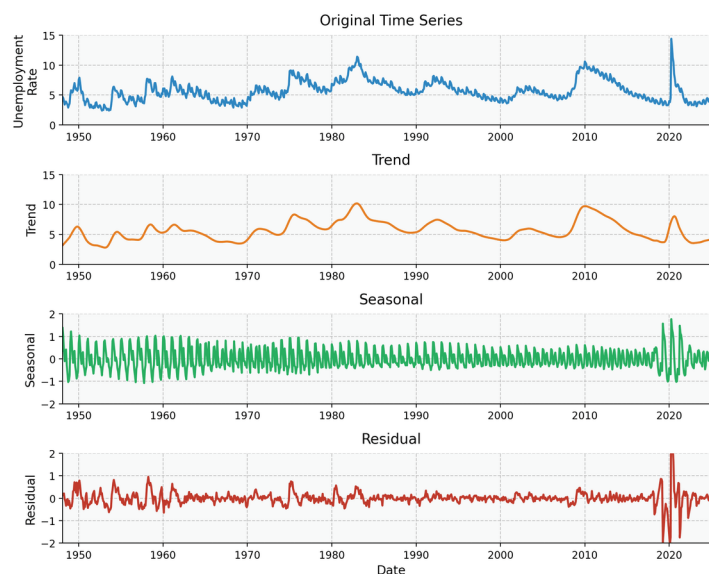


Figure 5.4.1: US unemployment rate from 1948 through 2025 from the [Federal Reserve Bank of St. Louis](#) with STL decomposition using the default settings in `statsmodels` of $n_i = 5$ and $n_o = 0$. Plot consists of original series (top), trends and cycles (second from top), seasonal contribution (third from top), and residual variation not explained (bottom).

Applying the [assessment method from classical decomposition](#) results in $F_T = 0.953$ and $F_S = 0.625$. Compared with the classical

decomposition results of $F_T = 0.928$ and $F_S = 0.379$, this indicates that STL has done a marginally better job isolating the trend but a substantially better job isolating the season component.

Exercise 5.4.1

Go back to **Section 4.2** on classical decomposition and rerun the exercises, this time using

```
from statsmodels.tsa.seasonal import STL
stl_decomp = STL(employment_df["Unemployment"], robust=False, period=12).fit(inner_iter=
```

instead of

```
from statsmodels.tsa.seasonal import seasonal_decompose
classic_decomp = seasonal_decompose(employment_df["Unemployment"], model="additive", per
```

How do the plots of the individual components compare across methods? What happens if you use robust STL instead?

Multiple Time Scales

As presented, STL can only decompose a single seasonality. generalize STL to handle multiple seasonal lengths (e.g. both a daily and yearly seasonality in temperature) in *Multiple Seasonal-Trend decomposition using LOESS* (MSTL). MSTL was implemented in `statsmodels` as `statsmodels.tsa.seasonal.MSTL` in version `0.14.0` released in May of 2023. While MSTL can be a useful tool, in practice it is not currently widely adopted. In general, I would usually recommend use of frequency domain techniques for analyzing multiple seasonal lengths in a single time series. While not as immediately interpretable as STL or MSTL, frequency domain methods more than make up for the increased complexity with their tremendous ability to tease out the contributions of different seasonal trends.

Footnotes

1. Cleveland et al. (1990) write that the "L" stands for LOESS. We will maintain this terminology, though for the purposes this for book we will only need the narrower univariate LOWESS algorithm rather than the multivariate LOESS.
2. We will explore why this filter is referred to as "low-pass" in the chapter covering the frequency domain.

References

1. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, 6(1), 3-73.
2. Bandara, K., Hyndman, R. J., & Bergmeir, C. (2021). MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns. *arXiv*. 10.48550/ARXIV.2107.13462

6: ARMA MODELS

6.1: Introduction

6.2: Autoregressive Models

6.3: Moving Average Models

6.4: ARMA Models

This page titled [6: ARMA Models](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

6.1: Introduction

We are now in a position to use our understanding of time series to create theoretical models to help us both to understand processes from a theoretical point of view (time series analysis) and to predict future values (forecasting). There are two models that lie at the heart of traditional time series analysis:

1. Autoregressive (AR): AR models, which are built on prior observations, can be thought of as an extension of random walks in which the current value depends on one or more prior timesteps. AR models take the form of

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t. \quad (6.1.1)$$

2. Moving average (MA): MA models are built on prior noise and generalize the idea of moving averages to any kernel. Importantly, MA models allow negative values and do not require coefficients to sum to unity. MA models take the form of

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \dots + \theta_q w_{t-q}. \quad (6.1.2)$$

After examining MA and AR models in isolation, we will move on to combining them into autoregressive moving average (ARMA) models. The chapter concludes with a discussion of modifications to ARMA models to deal with non-stationary time series, seasonal effects, and exogenous variables.

6.2: Autoregressive Models

Recall that a random walk is a model in which each time step's value is determined by the previous time step's value plus random noise, i.e.

$$x_t = x_{t-1} + w_t. \quad (6.2.1)$$

A random walk is not stationary due to its non-constant variance, making AR models not applicable. However, we could imagine a slightly different time series defined as

$$x_t = \phi x_{t-1} + w_t, \quad |\phi| < 1. \quad (6.2.2)$$

AR(1) Stationarity

Eq. 6.2.2 is a first order autoregressive process, denoted as AR(1). We will demonstrate that Eq. 6.2.2 describes a stationary process by iterating backwards and examining the properties of the resulting iterated series:

$$\begin{aligned} x_t &= \phi x_{t-1} + w_t \\ &= \phi (\phi x_{t-2} + w_{t-1}) + w_t \\ &= \phi^2 x_{t-2} + \phi w_{t-1} + w_t \\ &= \phi^2 (\phi x_{t-3} + w_{t-2}) + \phi w_{t-1} + w_t \\ &\dots \\ &= \sum_{j=0}^{\infty} \phi^j w_{t-j} \end{aligned} \quad (6.2.3)$$

Exercise 6.2.1

Why will Eq. 6.2.3 fail for $|\phi| \geq 1$?

Answer available at the end of this section.

In coming sections, we will learn that Eq. 6.2.3 is the infinite moving average, or MA(∞) representation of an AR process. For our purposes, we can think of it as a way to shed light on the behavior of an AR process.

AR(1) Mean

From the last line of Eq. 6.2.3, we conclude that the mean of an AR(1) process is

$$\begin{aligned} \mathbb{E}[x_t] &= \mathbb{E}\left[\sum_{j=0}^{\infty} \phi^j w_{t-j}\right] \\ &= \sum_{j=0}^{\infty} \phi^j \mathbb{E}[w_{t-j}] \\ &= 0. \end{aligned} \quad (6.2.4)$$

AR(1) Autocovariance

The autocovariance can be derived analogously. For any stationary AR(1) process with zero mean, $\gamma(h) = \text{Cov}(x_{t+h}, x_t)$. Should the mean not be zero, replace x_t with $x_t - \mu_x$. We can then derive $\gamma(h)$ as

$$\begin{aligned}
 \gamma(h) &= \text{Cov}(x_{t+h}, x_t) \\
 &= \mathbb{E} \left[\left(\sum_{j=0}^{\infty} \phi^j w_{t+h-j} \right) \left(\sum_{k=0}^{\infty} \phi^k w_{t-k} \right) \right] \\
 &= \mathbb{E} [(w_{t+h} + \dots + \phi^h w_t + \phi^{h+1} w_{t-1} + \dots) (w_t + \phi w_{t-1} + \dots)] .
 \end{aligned} \tag{6.2.5}$$

Notation

In the first line of Eq. 6.2.5 we have used x_t and x_{t+h} to refer to the series x indexed by t or $t+h$. Going forward, we will use x_t to refer either to the series x indexed by t or the value of x at time t with the meaning determined by context.

Noting that the covariance for the w_t 's is $\text{Cov}(w_i, w_j) = \delta_{ij} \sigma_w^2$, we can line up all non-zero contributions stemming from cross-terms where w_{t+h-j} in the left series matches w_{t-k} in the right series. This will require $h-j = -k$ i.e. $j = h+k$, giving us

$$\begin{aligned}
 \gamma(h) &= \sigma_w^2 \sum_{k=0}^{\infty} \phi^{h+k} \phi^k \\
 &= \sigma_w^2 \phi^h \sum_{k=0}^{\infty} \phi^{2k} \\
 &= \sigma_w^2 \frac{\phi^h}{1 - \phi^2},
 \end{aligned} \tag{6.2.6}$$

where we have cast the autocovariance as an infinite geometric series. Provided that σ_w^2 is finite, Eq. 6.2.6 will be finite for all values of h . Finally, the last line of Eq. 6.2.6 depends solely on the separation h , completing the proof that Eq. 6.2.2 represents a stationary process.

AR(1) Autocorrelation

Recall that the autocorrelation $\rho(h)$ for a stationary process is given by

$$\rho(h) \triangleq \frac{\gamma(h)}{\gamma(0)}. \tag{6.2.7}$$

Plugging in the result from Eq. 6.2.6, we obtain:

$$\begin{aligned}
 \rho(h) &= \frac{\sigma_w^2 \frac{\phi^h}{1 - \phi^2}}{\sigma_w^2 \frac{1}{1 - \phi^2}} \\
 &= \phi^h
 \end{aligned} \tag{6.2.8}$$

Exponential Decay of Autocorrelations

Eq. 6.2.8 highlights the signal characteristic of stationary AR processes, namely that the autocorrelation $\rho(h)$ dies off exponentially with the number of lags.

In the event that $\phi < 0$, $\rho(h)$ will also die off in a sinusoidal fashion, alternating between positive and negative values due to serial negative correlation. These two possibilities are demonstrated in Figure 6.2.1.

Theoretical autocorrelation for stationary AR(1) processes with ϕ

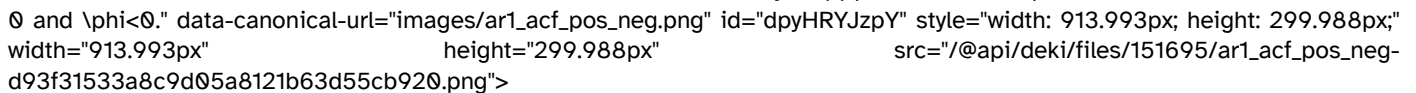
0 and $\phi < 0$.


Figure 6.2.1: Copy and Paste Caption here. (Copyright; author via source)

Exercise 6.2.1

We stated above that we cannot fit a non-stationary AR model, but we haven't explained what the problem is. A full exploration of the problem with non-stationary models requires a deeper dive into the mathematical underpinnings of ARMA models; however, we can begin to see some of the problems by creating a few simulated time series in this exercise inspired by Granger & Newbold (1974). Run the following code to explore spurious correlations in random walks. Note that `\#\\%` denotes a new code cell.

```
import numpy as np
import plotly.express as px
%%
np.random.seed(4400) # set seed for reproducibility
SIZE = 1000
%%
# Create random walks by summing random noise.
rw_1 = np.cumsum(a=np.random.normal(loc=0, scale=1, size=SIZE))
rw_2 = np.cumsum(a=np.random.normal(loc=0, scale=1, size=SIZE))
print("Correlation between random walks: ", np.corrcoef(rw_1, rw_2)[1,0])
%%
# examine quick plot
px.line(x=np.arange(SIZE), y=[rw_1, rw_2])
%%
# In this cell we will create two AR(1) processes that are close to random walks in term
PHI = 0.95
ar1_1 = np.empty(SIZE)
ar1_1[0] = 0
for idx in range(1, SIZE):
    ar1_1[idx] = PHI*ar1_1[idx-1] + np.random.normal(loc=0, scale=1)
ar1_2 = np.empty(SIZE)
ar1_2[0] = 0
for idx in range(1, SIZE):
    ar1_2[idx] = PHI*ar1_2[idx-1] + np.random.normal(loc=0, scale=1)
print("Correlation between AR(1) processes: ", np.corrcoef(ar1_1, ar1_2)[1,0])
%%
px.line(x=np.arange(SIZE), y=[ar1_1, ar1_2])

-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[2], line 2
      1 import numpy as np
----> 2 import plotly.express as px
      3 #%%
      4 np.random.seed(4400) # set seed for reproducibility

ModuleNotFoundError: No module named 'plotly'
```

What happens if you run the code multiple times with different seeds? How frequently do you observe a spurious high correlation between the random walks? What about between the AR(1) processes?

AR(1) with Nonzero Mean

Up to this point, we've assumed that our AR(1) model has a mean of 0 (or that we subtracted the mean prior to our analysis), in which case ϕ exerts a sort of gravitational pull to bring values back to 0 by damping out previous noise. We can extend this to an AR(1) process with a nonzero mean μ by subtracting the mean from each observation in the AR model itself

$$\begin{aligned}
 x_t - \mu &= \phi (x_{t-1} - \mu) + w_t \\
 x_t &= \phi x_{t-1} - \phi \mu + \mu + w_t \\
 &= \phi x_{t-1} + (1 - \phi) \mu + w_t \\
 &= \alpha + \phi x_{t-1} + w_t,
 \end{aligned}
 \tag{6.2.9}$$

where $\alpha \triangleq \mu(1 - \phi)$ functions the same way that an intercept term would in standard linear regression.

Important

The intercept term in an AR model is not equal to the expectation value of the series, but the two terms are related.

AR(p) Models

It's straightforward to generalize AR models to higher order $AR(p)$ models with $p \geq 1$ by regressing the time series onto versions of itself of increasing lag. A general $AR(p)$ model is given as

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t, \quad \phi_p \neq 0. \quad (6.2.10)$$

A series with a nonzero mean is handled as

$$\begin{aligned} x_t - \mu &= \phi_1 (x_{t-1} - \mu) + \phi_2 (x_{t-2} - \mu) + \dots + \phi_p (x_{t-p} - \mu) + w_t \\ x_t &= \mu (1 - \phi_1 - \phi_2 - \dots - \phi_p) + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t \\ &= \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t, \end{aligned} \quad (6.2.11)$$

where as before the intercept is given as $\alpha \triangleq \mu (1 - \phi_1 - \phi_2 - \dots - \phi_p)$.

Deriving the theoretical autocovariance and autocorrelation functions of an $AR(p)$ process is somewhat more involved than for an $AR(1)$ model. We will defer the derivation until after we have covered representing AR models as infinite series of noise terms in section 4.

AR Models in Backshift Notation

Autoregressive Operator

Recall from Eq. 6.2.14 in section 4.3 that the backshift operator $\mathbb{B}$ increments a time series back by one time step. Using this definition, we may express Eq. 6.2.2 as

$$\begin{aligned} x_t &= \phi \mathbb{B} x_t + w_t \\ x_t - \phi \mathbb{B} x_t &= w_t \\ (1 - \phi \mathbb{B}) x_t &= w_t. \end{aligned} \quad (6.2.12)$$

We can expand this definition to represent $AR(p)$ models using backshift notation as

$$(1 - \phi_1 \mathbb{B} - \phi_2 \mathbb{B}^2 - \dots - \phi_p \mathbb{B}^p) x_t = w_t, \quad (6.2.13)$$

or in more compact form

$$\phi(\mathbb{B}) x_t = w_t, \quad (6.2.14)$$

where $\phi(\mathbb{B})$ is the autoregressive operator defined as:

$$\phi(\mathbb{B}) \triangleq 1 - \phi_1 \mathbb{B} - \phi_2 \mathbb{B}^2 - \dots - \phi_p \mathbb{B}^p. \quad (6.2.15)$$

For stationary AR models ($|\phi| < 1$ for $AR(1)$), we can define an inverse autoregressive operator $\phi^{-1}(\mathbb{B})$

$$\phi^{-1}(\mathbb{B}) \phi(\mathbb{B}) x_t = \phi^{-1}(\mathbb{B}) w_t \quad (6.2.16)$$

or

$$x_t = \phi^{-1}(\mathbb{B}) w_t. \quad (6.2.17)$$

For an $AR(1)$ model, given that $\phi^{-1}(\mathbb{B}) = \frac{1}{1-\phi\mathbb{B}}$ and $|\phi| < 1$, we can represent $\phi^{-1}(\mathbb{B})$ as an infinite geometric series

$$\phi^{-1}(\mathbb{B}) = 1 + \phi\mathbb{B} + \phi^2\mathbb{B}^2 + \phi^3\mathbb{B}^3 + \dots \quad (6.2.18)$$

Exercise 6.2.1

Why does Eq. 6.2.18 require $|\phi| < 1$?

Answer available at the end of this section.

Causal AR Models

What is the purpose of representing AR processes in the form of Eq. 6.2.13 or 6.2.14? To answer this, begin by noting that Eq. 6.2.18 implies that $x_t = w_t + \phi w_{t-1} + \phi^2 w_{t-2} + \dots$, as derived in Eq. 6.2.3. This is an example of a causal model, in which a series can be treated as being generated by an infinite series of noise terms. Random walks, on the other hand, are non-causal as they will increase without bound as we include more and more time steps. Series with $|\phi| > 1$ are referred to as explosive as past noise becomes arbitrarily magnified as ϕ is raised to increasing powers-exactly what happens in a physical explosion (at least until the fuel is exhausted).

Non-causal processes such as unit root and explosive processes are not stationary, making many of the tools used in time series analysis unsuitable. We can easily determine if an AR(1) process is causal by examining ϕ , how might we do the same for higher order AR(p) models? Imagine if we could factor an AR model such that

$$(1 - \phi_1\mathbb{B} - \phi_2\mathbb{B}^2 - \dots - \phi_p\mathbb{B}^p) x_t = (1 - \phi'_1\mathbb{B})(1 - \phi'_2\mathbb{B}) \dots (1 - \phi'_p\mathbb{B}) x_t. \quad (6.2.19)$$

Given such a factoring, we could quickly determine if our AR(p) model was causal-and hence stationary-simply by confirming that all ϕ' 's have an absolute value less than 1. While there is no guarantee this will be possible for real ϕ' 's, the fundamental theorem of algebra does ensure this is possible for complex values [2].

Let's explore this with an concrete example. Consider the second order autoregressive, or AR(2), model defined as

$$x_t = 1.25x_{t-1} - 0.375x_{t-2} + w_t. \quad (6.2.20)$$

Looking at Eq. 6.2.20, it's not immediately obvious if its ϕ' 's are greater than one. Let's recast it using the autoregressive operator $\phi(\mathbb{B})$

$$\begin{aligned} x_t &= 1.25x_{t-1} - 0.375x_{t-2} + w_t \\ x_t - 1.25x_{t-1} + 0.375x_{t-2} &= w_t \\ (1 - 1.25\mathbb{B} + 0.375\mathbb{B}^2) x_t &= w_t. \end{aligned} \quad (6.2.21)$$

Evidently, $\phi(\mathbb{B}) = 1 - 1.25\mathbb{B} + 0.375\mathbb{B}^2 = 1 - \frac{5}{4}\mathbb{B} + \frac{3}{8}\mathbb{B}^2$. Now comes the crucial step, we replace the backshift operator with a variable, call it z , and solve for the zeros of the polynomial $\phi(z)$.

$$\begin{aligned} 1 - \frac{5}{4}z + \frac{3}{8}z^2 &= 0 \\ (1 - \frac{1}{2}z) (1 - \frac{3}{4}z) &= 0 \\ z &= 2 \text{ or } z = \frac{4}{3} \end{aligned} \quad (6.2.22)$$

Eq. 6.2.20 is stationary because the ϕ' 's of $\frac{1}{2}$ and $\frac{3}{4}$ have absolute values less than one (lie inside the unit circle [3]). Equivalently, the roots of $\phi(z)$ (and consequently of $\phi(\mathbb{B})$) of 2 and $\frac{4}{3}$ have absolute values greater than one (lie outside the unit circle).

Unit Circle and Stationarity

It is an unfortunate fact of time series analysis that both the definition requiring all ϕ' 's to lie inside the unit circle and the definition requiring all roots of z to lie outside the unit circle are both in use [4]. It is not always obvious which definition is being used. We will follow sources such as statsmodels and Shumway & Stoffer (2025) and require all roots to lie outside the unit circle, but be aware that this convention is not universal.

Sign of ϕ and Complex Roots

As seen above, the signal characteristic of stationary AR models is the presence of exponentially decaying autocovariance (and consequently autocorrelation) functions. For an $AR(p)$ process with $p \geq 2$, the decay may also exhibit sinusoidal oscillations, potentially with a period greater than p .

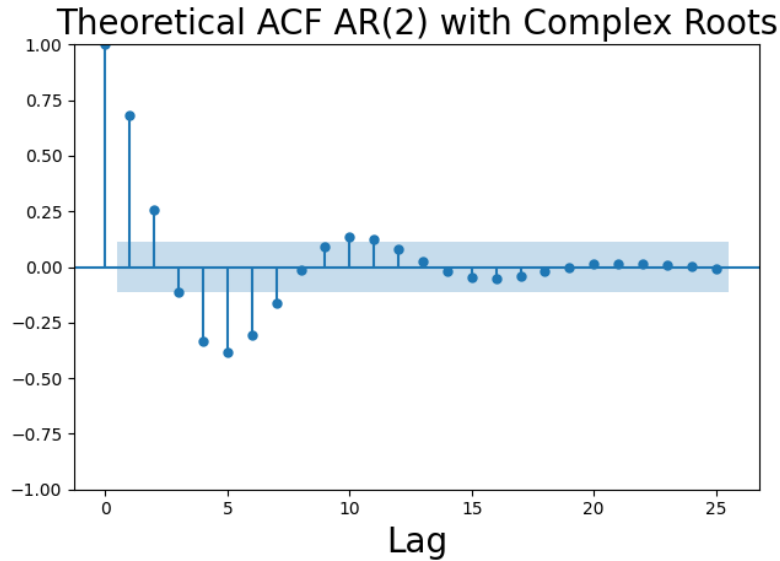


Figure 6.2.2: Copy and Paste Caption here. (Copyright; author via source)

If the above fails to render correctly in your browser you can also open the demo as a new browser window using the Open Demo in a New Tab ↗ button at the top of the frame. Note that you may need to enable popups for this to work.

By the quadratic formula, an $AR(2)$ process with associated polynomial $\phi(z) = 1 - \phi_1 z - \phi_2 z^2$ will be stationary only if

$$\left| \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \right| > 1 \quad (6.2.23)$$

Eq. 6.2.23 can be broken into three distinct requirements:

1. $\phi_1 + \phi_2 < 1$
2. $\phi_2 - \phi_1 < 1$
3. $\phi_2 > -1$

Some sources state the third requirement as $|\phi_2| < 1$, though this is not strictly necessary as combining the first two requirements above already enforces $\phi_2 < 1$.

From Eq. 6.2.23 we see that the roots of an $AR(2)$ model will be complex if $\phi_1^2 + 4\phi_2 < 0$, i.e. if $\phi_2 < -\frac{\phi_1^2}{4}$. In Sec. 4 of this chapter we will see that AR processes with complex roots have a special property of exhibiting a "pseudo-seasonality." These conditions are depicted in the Figure 6.2.3

Copy and Paste
Image here.
Delete this
placeholder image

{Values of ϕ_1 and ϕ_2 for AR(2) process demonstrating the boundary conditions for stationarity and real/complex roots.}

Figure 6.2.1: Values of ϕ_1 and ϕ_2 for AR(2) process demonstrating the boundary conditions for stationarity and real/complex roots.

Higher Order Causal Models

We could expand the process above for finding unit roots in AR(2) models to higher order AR(p) models, but in practice there's no need to do this by hand.

`statsmodels.tsa.arima_process.ArmaProcess` provides theoretical properties of AR (and more broadly ARMA) models for us. The following code demonstrates using this module both to determine stationarity and to extract the roots of an AR model. Note that the sign convention follows $\phi(B)$ from Eq. 6.2.21, not Eq. 6.2.20.

```
from statsmodels.tsa.arima_process import ArmaProcess
ar2 = ArmaProcess(ar=[1, -1.25, 0.375], # use phi values defined via phi(B)
                  ma=None, # pure AR process, no MA component
                  )
print(f"AR(2) model is stationary: {ar2.isstationary}")
print(f"Roots of AR(2) model: {ar2.arroots}")

Hello world!
```

Exercise 6.2.1

What would the root(s) of a random walk be? Repeat the above to confirm your calculation. Find the roots of

$$x_t = -x_{t-1} - 0.5x_{t-2} + w_t. \quad (6.2.24)$$

Is this AR(2) process stationary?

Answer available at the end of this section.

Unit Root vs. Explosive Process

While the mathematical requirement is that all roots lie outside the unit circle (or all ϕ' 's lie inside the unit circle), in practice we are usually only concerned with differentiating between processes with all roots outside the unit circle versus processes with at least one root on the unit circle, i.e. a unit root. This is because roots inside the unit circle correspond to explosive processes, which are both uncommon in time series analysis and generally self-evident when they do occur. Consequently, we usually only worry about the existence of a unit root corresponding to random walk type behavior.

Where do AR Processes Arise?

In economics and related disciplines, AR models are often considered to be "longer-memory" models ^[5]. This makes sense, as exponentially decaying autocorrelation means that a noise term, or "shock," will take many steps to be "forgotten" (i.e. fade to statistical insignificance). Such a model is appropriate for a wide range of scenarios, ranging from climate science to economic inflation and stock market returns. In the following problem, we will explore using AR processes to get a baseline approximation to solar activity.

Exercise 6.2.1

In this problem we will explore fitting $AR(p)$ models to the sunspots dataset the comes with statsmodels and using both AIC and BIC to rank solutions.

```
import numpy as np
import pandas as pd
import plotly.express as px
from statsmodels.datasets import sunspots
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.arima_process import ArmaProcess
#%%
print(sunspots.NOTE)
#%%
sunspots_df = sunspots.load_pandas().data
sunspots_df["YEAR"] = pd.to_datetime(sunspots_df["YEAR"], format='%Y', )
sunspots_df.set_index("YEAR", inplace=True, drop=True)
sunspots_df = sunspots_df.asfreq('YS') # Explicitly set the frequency to 'Year Start'
sunspots_df.head()
#%%
# examine quick plot
px.line(sunspots_df)
#%%
# create and save AR(p) models for p=1,2,3,4
ar_model_dict = {}
for p in range(1,5):
    print(f"AR Model for p={p}")
    ar_model_dict[p] = ARIMA(sunspots_df, order=(p,0,0)).fit()
    print("AIC: ", ar_model_dict[p].aic)
    print("BIC: ", ar_model_dict[p].bic)
    print("\n")
```

You should see that the $AR(1)$ performs notably worse, whereas the $AR(2)$, $AR(3)$, and $AR(4)$ models are roughly tied. AIC slightly favors the $AR(3)$, whereas (consistent with its tendency to choose more parsimonious models) BIC very slightly favors the $AR(2)$, though in both cases the difference between models for $p > 1$ is small enough that I wouldn't read too much into it. Let's look at a summary and the theoretical properties of the $AR(2)$ model. Don't worry if you're not yet familiar with some of the terms in the summary, we'll get to the relevant ones in subsequent sections.

```
# print model summary
ar_model_dict[2].summary()
#%%
# create a theoretical ARMA model to explore its properties
phi_1 = ar_model_dict[2].params['ar.L1']
phi_2 = ar_model_dict[2].params['ar.L2']
ar2 = ArmaProcess(ar=[1, -phi_1, -phi_2], # note change of sign of phi values
                  ma = None, # pure AR process, no MA component
                  )
print(f"AR(2) model is stationary: {ar2.isstationary}")
```

```
print(f"Roots of AR(2) model: {ar2.arroots}")
# get pseudo-periodicity
print(f"Pseudo-period: {round(2*np.pi/np.abs(np.angle(ar2.arroots[0])), 2)} years")
```

Is the pseudo-period consistent with the plot from above?

Next, let's compare the sample autocovariance to the theoretical autocovariance of our proposed AR(2) model.

```
sample_acf = plot_acf(sunspots_df, title="Sample ACF")
sample_acf.show()

theoretical_acf_plot = plot_acf(ar2.acf(), adjusted=False, title='Theoretical ACF')
theoretical_acf_plot.show()
```

Do the two plots appear to agree?

Finally, let's explore using `statsmodels` to generate one-step-ahead in-sample predictions and recursively forecast the next 22 years out-of-sample. Don't worry if you don't get everything in this cell, we'll revisit predicting and forecasting in subsequent sections.

```
# In this cell we will begin exploring using AR models for prediction.
# We will first fit perform one-step-ahead prediction throughout the observation window,
pred = ar_model_dict[2].get_prediction(dynamic=False) # one-step-ahead prediction for in
forecast = ar_model_dict[2].get_forecast(22) # forecast for out-of-sample data
pred_df = pd.DataFrame(index=pred.predicted_mean.index.append(forecast.predicted_mean.in
                      columns=["Observed", "One-Step-Ahead-Forecast"],
                      )
pred_df["Observed"] = sunspots_df
pred_df["One-Step-Ahead-Forecast"] = pd.concat([pred.predicted_mean, forecast.predicted_
px.line(pred_df)
```

Does the in-sample prediction seem reasonable? What about the out-of-sample forecast?

Footnotes

1. Yule's original 1927 paper (Yule (1927)) introducing autoregressive models motivated the concept by drawing an analogy to a randomly perturbed oscillatory system-in effect discretizing the differential equations governing damped harmonic oscillators (though Yule himself did not directly use differential equation terminology). Modern students coming from a data science or statistics background are generally more comfortable interpreting AR models as form of linear regression in which the features are lagged versions of the time series itself.
2. The fundamental theorem of algebra states that any polynomial of degree p has exactly p roots (provided we allow for complex roots). The theorem allows for a single root to appear multiple times such as in the case of $1 + 2x + x^2 = (1 + x)^2$, which has the root $x = -1$ with multiplicity 2.
3. The unit circle is simply the set of all complex numbers such that $|a + bi| = 1$, or equivalently all complex numbers of the form $e^{i\theta}$, $\theta \in [0, 2\pi)$.
4. The actual requirement is derived by using the reciprocal definition of the roots z from the definition we used. Our definition using φ' expresses the same idea from a different angle.
5. I used the term "longer-term" rather than "long-term" as the latter term is reserved for a class of models with even slower decay than AR models.

References

1. Granger, C. W. J., & Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics*, 2(2), 111-120. [https://doi.org/10.1016/0304-4076\(74\)90034-7](https://doi.org/10.1016/0304-4076(74)90034-7)
 2. Shumway, R. H., & Stoffer, D. S. (2025). *Time Series Analysis and Its Applications: With R Examples*. In Springer Texts in Statistics. Springer Nature Switzerland. 10.1007/978-3-031-70584-7
 3. Yule, G. U. (1927). VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646), 267-298.
-

Answers

Exercise 6.2.1

Solution: Eq. 6.2.3 assumes that increasing powers of ϕ have diminishing magnitudes so that $\sum_{j=0}^{\infty} \phi^j w_{t-j}$ converges to a finite sum. For $|\phi| \geq 1$, the sum will instead diverge to infinity. This is exactly what we observed plotting the variance of a random walk in Figure 6.2.1 of section 3.2.

Exercise 6.2.1

A geometric series will only converge for terms with an absolute value less than 1, and additionally will have a division by zero error at 1. Conceptually, for $|\phi| > 1$, Eq. 6.2.18 will blow up to infinity as we raise ϕ to higher and higher powers. At $|\phi| = 1$, even though ϕ itself will not go to infinity, the overall infinite summation of a non-decreasing series will still go to infinity.

Exercise 6.2.1

A random walk has $\phi(\mathbb{B}) = 1 - \mathbb{B}$, resulting in it having a unit root of 1. The AR(2) process has roots of $-1 \pm i$. The roots have an absolute value of $\sqrt{2}$, which is outside the unit circle, making the process stationary.

6.3: Moving Average Models

Recall that we might construct a three-point moving average as

$$v_t = \frac{1}{3}(w_{t-1} + w_t + w_{t+1}), \quad (6.3.1)$$

which will result in v_t having the same expectation value as w_t , but will introduce autocorrelation at lags $h = 1, 2$.

Moving average (MA) models of order $q > 1$ generalize this idea with the model

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \dots + \theta_q w_{t-q}, \quad \theta_q \neq 0. \quad (6.3.2)$$

By convention, the number of lags included in a given MA model is represented as q , and the overall model is referred to as an MA(q) model. Unlike a true moving average, MA models do not require the θ 's to sum to unity, or even to be positive.

Exercise 6.3.1

Why doesn't an MA model use w_{t+1} ?

Answer available at the end of this section.

Moving Average Terminology

We are arguably abusing the terminology by referring to Eq. 6.3.2 as a "moving average" without ever requiring the θ 's to actually average w_t . While MA models might have been better named something like "noise convolution" models, the MA terminology has been universally adopted and we will use that convention here.

Parameter Estimation

At this point, you may well be wondering how we could ever expect to estimate the θ 's in Eq. 6.3.2 in real life. AR models can be calculated by lining up past observations and running a least squares regression (though in practice related methods such as Yule-Walker are often favored). In contrast, how are we supposed to know the w_t 's necessary to estimate an MA model? We observed the values x_t , not the noise w_t !

To answer this, let us take a step back and imagine we knew for a fact that a given process was described by the MA(1) model

$$x_t = w_t + \theta w_{t-1}. \quad (6.3.3)$$

In this case, our estimate of $\hat{w}_t$ would be given by $x_t - \theta \hat{w}_{t-1}$. By the same token we would estimate $\hat{w}_{t-1} = x_{t-1} - \theta \hat{w}_{t-2}$, $\hat{w}_{t-2} = x_{t-2} - \theta \hat{w}_{t-3}$, etc. We can flip this around the use maximum likelihood estimation (MLE) to estimate the value $\hat{\theta}$ most consistent with our observations. This procedure can be extended to higher level MA (q) processes using the same logic. Sources such as Shumway & Stoffer (2025) chapter 3.5 and Brockwell & Davis (1991) chapter 5.1-5.2 go into some detail regarding methods such as Newton-Raphson, Gauss-Newton, and the innovations algorithm. Understanding the exact algorithms used for implementing MLE for MA (and ARMA) models is not as important for a practicing data scientist as understanding that MLE is being used ^[1]. The use of MLE has two important ramifications you should be aware of:

1. MA models (and the MA component of ARMA models) are generally less numerically stable than pure AR models, resulting in them potentially being less reliable.
2. MA models are less interpretable than AR models. An AR model is purely determined by previous observations and its meanings and ramifications can be easily explained to clients. In contrast, an MA model relies on inferring an unseen noise term which can be challenging to explain to a less technical audience.

For these reasons, AR models are often preferred over MA models when they give comparable results with a similar number of parameters [2].

Moving Average Operator

As with AR models, we define the moving average operator as:

$$\theta(B) \triangleq 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q, \quad (6.3.4)$$

allowing us to express Eq. 6.3.2 as

$$\begin{aligned} x_t &= w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \dots + \theta_q w_{t-q} \\ &= \theta(B) w_t. \end{aligned} \quad (6.3.5)$$

Note the change in sign convention from the autoregressive operator.

MA(1) Mean & Autocovariance

Provided that w_t has a finite second moment, all MA processes will be stationary. The mean of a pure MA process will always be zero. The autocovariance is calculated by matching w_t terms, for an MA(1) process we have:

$$\gamma(h) = \begin{cases} (1 + \theta^2) \sigma_w^2 & h = 0 \\ \theta \sigma_w^2 & h = 1 \\ 0 & h > 1. \end{cases} \quad (6.3.6)$$

The autocorrelation for an MA(1) process is:

$$\rho(h) = \begin{cases} \frac{\theta}{1 + \theta^2} & h = 1 \\ 0 & h > 1 \end{cases} \quad (6.3.7)$$

Sudden Cutoff of Autocorrelation

A signature characteristic of an MA process is having statistically significant autocorrelation up to lag q followed by an immediate drop to statistical insignificance. This draws a strong contrast with AR processes, which exhibit exponentially decaying autocovariance.

Exercise 6.3.1

Derive Eqs. (6) and (7).

Answer available at the end of this section.

Sign of θ

Similar to what we observed with AR models in the previous section, an MA(1) model with $\theta < 0$ will introduce serial negative correlation, resulting in a jagged time series. Unlike AR models, a negative θ will not introduce oscillations in the autocorrelation for any lag greater than q . The following tool helps demonstrate the effects of different θ values for MA(1) and MA(2) processes.

If the above fails to render correctly in your browser you can also open the demo as a new browser window using the Open Demo in a New Tab ↗ button at the top of the frame. Note that you may need to enable popups for this to work.

Autocovariance of Higher Order MA Processes

An MA(q) process can be written as $x_t = \sum_{j=0}^q \theta_j w_{t-j}$, where $\theta_0 = 1$. The mean of an MA process is $\mathbb{E} \left[\sum_{j=0}^q \theta_j w_{t-j} \right]$

$= \sum_{j=0}^q \theta_j \mathbb{E}[w_{t-j}] = 0$. The autocovariance $\gamma(h)$ will depend on the number of w_t terms in common between x_t and x_{t+h} :

$$\begin{aligned}\gamma(h) &= \text{Cov}(x_{t+h}, x_t) \\ &= \text{Cov}\left(\sum_{j=0}^q \theta_j w_{t+h-j}, \sum_{k=0}^q \theta_k w_{t-k}\right) \\ &= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h}, & 0 \leq h \leq q \\ 0 & h > q. \end{cases}\end{aligned}\quad (6.3.8)$$

Autocorrelation of Higher Order MA Processes}

From Eq. 6.3.12, we see that

$$\gamma(0) = \sigma_w^2 (1 + \theta_1^2 + \dots + \theta_q^2), \quad (6.3.9)$$

thus giving us the autocorrelation

$$\begin{aligned}\rho(h) &= \frac{\sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{\sigma_w^2 (1 + \theta_1^2 + \dots + \theta_q^2)} \\ &= \begin{cases} \frac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1 + \theta_1^2 + \dots + \theta_q^2}, & 1 \leq h \leq q \\ 0 & h > q. \end{cases}\end{aligned}\quad (6.3.10)$$

When analyzing time series in real-life, observing a $\gamma(h)$ that is statistically significant for q terms and then drops to insignificance is a strong indicator of an MA(q) process.

Non-uniqueness of MA Models

While we theoretically understand MA processes as being the sum of noise terms, we do not directly observe the noise. We use functions such as the autocovariance and autocorrelation of observed values to derive the form of an MA process. MA models are, in general, not unique. Let us examine an MA(1) model (higher order q 's are derived analogously); for an MA(1) model θ and $\frac{1}{\theta}$ yield the same ρ :

$$\begin{aligned}\rho(1) &= \frac{\theta}{1 + \theta^2} \\ &= \frac{\theta}{1 + \theta^2} \times \frac{\frac{1}{\theta^2}}{\frac{1}{\theta^2}} \\ &= \frac{\frac{1}{\theta}}{\left(\frac{1}{\theta}\right)^2 + 1}\end{aligned}\quad (6.3.11)$$

Using the autocovariance $\gamma(h)$ won't help, either. To demonstrate this for an MA(1) model, let us return to Eq. 6.3.6, where we found that for an MA(1) process

$$\gamma(h) = \begin{cases} (1 + \theta^2) \sigma_w^2 & h = 0 \\ \theta \sigma_w^2 & h = 1 \\ 0 & h > 1. \end{cases}\quad (6.3.12)$$

To use an example from Shumway & Stoffer (2025), consider an MA(1) model $\sigma_w^2 = 1$ and $\theta = 5$. In this case, $\gamma(0) = 1 + 5^2 = 26$, and $\gamma(1) = 5$. Now consider a model with $\sigma_w^2 = 25$ and $\theta = \frac{1}{5}$. Here too, $\gamma(0) = (1 + \frac{1}{5^2}) 25 = (\frac{26}{25}) 25 = 26$, and $\gamma(1) = (\frac{1}{5}) 25 = 5$.

If we could somehow directly observe the noise w_t , we would be able to differentiate the models by looking at the variance of the noise. Unfortunately, we can only infer the variance of w_t from looking at $\gamma(h)$. We thus see that two distinct MA(1) models can equally well describe the same underlying process with no way to distinguish the "true" model (though I would caution any practicing data scientist to be wary of conflating reality itself with a model of reality such as an MA model).

Invertibility of MA Process

As seen in Eq. 6.3.15, models with $\theta = 5$ and $\theta = \frac{1}{5}$ yield the same ρ . Moreover, $\theta = 5, \sigma_w^2 = 1$ and $\theta = \frac{1}{5}, \sigma_w^2 = 25$ yield the same γ . We choose the model with $|\theta| < 1$ as this model is invertible. In statsmodels, this is enforced by the argument `enforce_invertibility` with `default=True`. This allows the infinite AR representation

$$\begin{aligned} w_t &= \theta^{-1}(\mathbb{B})x_t \\ &= \frac{1}{1 + \theta\mathbb{B}}x_t \\ &= \sum_{j=0}^{\infty} (-\theta)^j x_{t-j} \end{aligned} \tag{6.3.13}$$

where the negative sign arises from defining $\theta(\mathbb{B}) = 1 + \theta\mathbb{B} = 1 - (-\theta)\mathbb{B}$.

Exercise 6.3.1

How might a non-invertible MA(1) model arise?

Answer available at the end of this section.

AR(∞) and MA(∞) Representations

Invertibility plays the same role for MA models that causality (or stationarity) does for AR models. Indeed, we could repeat most all of the analysis performed for AR models simply by renaming the variables. Swapping x_t and w_t and renaming ϕ_i to $-\theta_i$ will allow us to transform an MA(1) model into an AR(∞) model ^[3] (as explicitly derived in Eq. 6.3.17). Thus, we arrive at the conclusion that any causal AR(1) can be represented as an MA(∞), and any invertible MA(1) model can be represented as an AR(∞). In the next section on ARMA we will develop tools to transform any causal AR(p) into an MA(∞), and likewise to convert any invertible MA(q) model into an AR(∞) model.

Where do MA Processes Arise?

Pure MA(q) models with a finite q are sometimes referred to as "short-memory" models to contrast them with the longer memory of AR models. Pure MA models are somewhat less prevalent but do arise in scenarios such as items with a shelf-life which inherently have a short "memory." As an example, a "shock" in the prices of dairy will quickly die off as we approach the end of the products' shelf-lives. A glut in production will become irrelevant once the extra products expire, whereas a scarcity of production will rapidly reset as future purchases return to their baseline (as there is no need to refill long term stockpiles) ^[4].

Arguably, however, MA models truly shine in the context of analyzing AR (or ARMA) models in their MA(∞) representations. We will return to this topic once we've laid additional foundation in the next section covering ARMA.

Footnotes

1. Mathematically, ordinary least squares (OLS) regression is the maximum likelihood estimator for a linear model under the assumption of iid normal errors. However, unlike most MLE that relies on various optimization algorithms, OLS has a closed form exact solution. As a result, OLS usually lacks the numerical instability issues that frequently plague other MLEs.
2. We will explore methods for converting between ARMA, pure AR, and pure MA models in section 4.
3. The function `arma_impulse_response` in statsmodels actually does convert to either AR(∞) or MA(∞) simply by switching which set of variables are used as ϕ_i and which are used as θ_i .
4. This is something of an oversimplification as in the United States the government maintains long-term cold cheese storage, in part to help smooth out supply chain shocks by absorbing excess production and providing relief during scenarios such as disaster relief. Nevertheless, our model is reasonable for local markets that may not participate in government cheese programs.

References

1. Shumway, R. H., \& Stoffer, D. S. (2025). Time Series Analysis and Its Applications: With R Examples. In Springer Texts in Statistics. Springer Nature Switzerland. 10.1007/978-3-031-70584-7
2. Brockwell, P. J., \& Davis, R. A. (1991). Time Series: Theory and Methods. In Springer Series in Statistics. Springer New York. 10.1007/978-1-4419-0320-4

Answers

Exercise 6.3.1

MA models (and ARMA in general) are designed to be used both for time series analysis and time series forecasting. Clearly, a model that needs to know the future in order to forecast it is not a particularly useful tool.

Exercise 6.3.1

Solution: This follows the same pattern derived for the autocovariance of a true moving average. Going through the cases in order and noting that $\text{Cov}(w_i, w_j) = \mathbb{E}[w_i w_j] = \sigma_w^2 \delta_{ij}$:

$$\begin{aligned}\gamma(0) &= \text{Cov}(x_t, x_t) \\ &= \mathbb{E}[x_t x_t] \\ &= \mathbb{E}[(w_t + \theta w_{t-1})(w_t + \theta w_{t-1})] \\ &= \mathbb{E}[w_t w_t] + \theta^2 \mathbb{E}[w_{t-1} w_{t-1}] \\ &= \sigma_w^2 + \theta^2 \sigma_w^2 \\ &= (1 + \theta^2) \sigma_w^2,\end{aligned}\tag{6.3.14}$$

$$\begin{aligned}\gamma(1) &= \text{Cov}(x_{t+1}, x_t) \\ &= \mathbb{E}[x_{t+1} x_t] \\ &= \mathbb{E}[(w_{t+1} + \theta w_t)(w_t + \theta w_{t-1})] \\ &= \theta \mathbb{E}[w_t w_t] \\ &= \theta \sigma_w^2,\end{aligned}\tag{6.3.15}$$

$$\begin{aligned}\gamma(2) &= \text{Cov}(x_{t+2}, x_t) \\ &= \mathbb{E}[x_{t+2} x_t] \\ &= \mathbb{E}[(w_{t+2} + \theta w_{t+1})(w_t + \theta w_{t-1})] \\ &= 0.\end{aligned}\tag{6.3.16}$$

Any $h > 2$ will likewise have no noise terms in common and hence zero autocovariance. Eq. 6.3.7 is derived by dividing Eq. 6.3.9 by Eq. 6.3.8:

$$\begin{aligned}\rho(1) &= \frac{\theta \sigma_w^2}{(1 + \theta^2) \sigma_w^2} \\ &= \frac{\theta}{1 + \theta^2}\end{aligned}\tag{6.3.17}$$

Exercise 6.3.1

An MA(1) model will be non-invertible if $\theta = \frac{1}{\theta}$, i.e. if $\theta = \pm 1$. This frequently arises as the result of excessive differencing when trying to make a model stationary. To give a basic example, consider the MA(1) model of white noise

$$x_t = w_t.\tag{6.3.18}$$

If we apply differencing to white noise, we get the series

$$\nabla x_t = w_t - w_{t-1},\tag{6.3.19}$$

which is a non-invertible MA(1) model. This reinforces why we should limit differencing to cases where it is strictly necessary.

This page titled [6.3: Moving Average Models](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

6.4: ARMA Models

In this section, we will combine AR and MA models to create autoregressive moving average (ARMA) models. A model is defined as $\text{ARMA}(p, q)$ if it is stationary and can be expressed as

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}. \quad (6.4.1)$$

If $\mathbb{E}[x_t] \neq 0$, we instead define the model as

$$x_t = \alpha + \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}, \quad (6.4.2)$$

with $\alpha = \mu(1 - \phi_1 - \dots - \phi_p)$. In general, we will assume that the mean has been subtracted from our time series so that $\mu = \alpha = 0$.

In the process of developing ARMA models we will build tools that allow us to better understand the autocorrelation of various processes and how to apply this knowledge to practical data science applications.

Operator Notation and Parameter Redundancy

In keeping with the operators derived in sections 6.2 and 6.3, an $\text{ARMA}(p, q)$ model may be expressed in operator form as

$$\phi(\mathbb{B})x_t = \theta(\mathbb{B})w_t \quad (6.4.3)$$

Eq. 6.4.3 presents a potential problem inasmuch as we can multiply both sides by the same term and still maintain a valid model, for example

$$\eta(\mathbb{B})\phi(\mathbb{B})x_t = \eta(\mathbb{B})\theta(\mathbb{B})w_t \quad (6.4.4)$$

To give a concrete example, take the white noise process $x_t = w_t$, and multiply it by some $\eta(\mathbb{B})$, say $(1 - 0.5\mathbb{B})$:

$$(1 - 0.5B)x_t = (1 - 0.5B)w_t, \quad (6.4.5)$$

or

$$x_t = 0.5x_{t-1} - 0.5w_{t-1} + w_t, \quad (6.4.6)$$

which is now an $\text{ARMA}(1,1)$ model, despite only describing white noise. You might be tempted to rely on the statistical significance of the fitted ϕ and θ values to identify parameter redundancy. The following code helps demonstrate why we cannot use this tactic. Note that it is particularly strongly recommended that you run this code yourself to appreciate parameter redundancy.

```
!pip install statsmodel
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

###
# initialize empty lists
phi_estimates = []
theta_estimates = []
phi_significance = []
theta_significance = []
# create 100 white noise simulations and fit ARMA(1,1) model
for idx in range(100):
    white_noise = np.random.normal(size=1000)
    arma_1_1 = ARIMA(white_noise, order=(1,0,1)).fit()
    phi_estimates.append(arma_1_1.params[1]) # estimated phi
    theta_estimates.append(arma_1_1.params[2]) # estimated theta
```

```

phi_significance.append(arma_1_1.pvalues[1]) # p-value of estimated phi
theta_significance.append(arma_1_1.pvalues[2]) # p-value of estimated theta

###
# plot phi and theta values
fig, ax = plt.subplots()
ax.scatter(phi_estimates, theta_estimates, c=theta_significance, cmap='viridis')
ax.set_xlabel('φ Estimates', fontsize=18)
ax.set_ylabel('θ Estimates', fontsize=18)
ax.set_title('Scatter Plot of φ vs θ Estimates', fontsize=20)

###
# plot histogram of p-values
fig, ax = plt.subplots()
ax.hist(theta_significance, bins=50, alpha=0.7, label='θ Estimates')
ax.hist(phi_significance, bins=50, alpha=0.7, label='φ Estimates')
ax.set_xlabel('Parameter Significance', fontsize=18)
ax.set_ylabel('Frequency', fontsize=18)
ax.set_title('Histogram of φ and θ p-values', fontsize=20)
ax.legend()

```

In the above code, you should see that in almost all cases $\theta \approx -\phi$, with values in $(-1, 1)$ (by default, statsmodels will attempt to enforce both stationarity and invertibility). You should also see that the majority of all estimates were statistically significant.

ARMA to AR(∞) and MA(∞)

AR(∞) and MA(∞) Terminology

Before deriving them, we first define the terminology for AR(∞) and MA(∞) representations. An MA(∞) representation is traditionally denoted using the variable ψ as

$$\begin{aligned}
 x_t &= \sum_{j=0}^{\infty} \psi_j w_{t-j} \\
 &= \sum_{j=0}^{\infty} \psi_j \mathbb{B}^j w_t \\
 &= \psi(\mathbb{B}) w_t
 \end{aligned} \tag{6.4.7}$$

where $\psi_0 \triangleq 1$.

Following the same logic, the AR(∞) representation, often denoted by the variable π (apologies for using π as a variable instead of a constant), is given as

$$\begin{aligned}
 w_t &= \sum_{j=0}^{\infty} \pi_j x_{t-j} \\
 &= \sum_{j=0}^{\infty} \pi_j \mathbb{B}^j x_t \\
 &= \pi(\mathbb{B}) x_t
 \end{aligned} \tag{6.4.8}$$

where $\pi_0 \triangleq 1$.

Autocovariance and Autocorrelation of AR(p) Models

Deriving Weights for Causal Process

We've seen that the autoregressive operator $\phi(\mathbb{B})$ is extremely useful for determining stationarity. In this section, we will explore another useful application of $\phi(\mathbb{B})$, namely converting a finite AR model into a MA(∞) consisting of an infinite series of noise

(or shock) terms. This in turn will help us understand the autocovariance of AR processes and derive confidence intervals for predictions.

We've already seen one example of converting to an infinite series of noise for AR(1) models

$$\begin{aligned} x_t &= (1 + \phi\mathbb{B} + \phi^2\mathbb{B}^2 + \phi^3\mathbb{B}^3 + \dots) w_t \\ &= w_t + \phi w_{t-1} + \phi^2 w_{t-2} + \phi^3 w_{t-3} + \dots \end{aligned} \quad (6.4.9)$$

How might we go about finding weights, call them ψ_j 's with associated operator $\psi(\mathbb{B})$, for higher order AR(p) processes? In other words, we want to find ψ weights to satisfy the equation

$$x_t = \psi(\mathbb{B})w_t \quad (6.4.10)$$

where we have defined $\psi_0 \triangleq 1$. Given that an AR model is defined as

$$w_t = \phi(\mathbb{B})x_t, \quad (6.4.11)$$

we may combine Eqs. (10) and (11) to arrive at

$$\begin{aligned} w_t &= \phi(\mathbb{B})x_t \\ &= \phi(\mathbb{B})\psi(\mathbb{B})w_t \\ &= (1 - \phi_1\mathbb{B} - \phi_2\mathbb{B}^2 - \dots - \phi_p\mathbb{B}^p) (1 + \psi_1\mathbb{B} + \psi_2\mathbb{B}^2 + \dots) \\ &= 1 + (\psi_1 - \phi_1)\mathbb{B} + (\psi_2 - \phi_2 - \psi_1\phi_1)\mathbb{B}^2 + \dots \end{aligned} \quad (6.4.12)$$

Given that the left-hand side of Eq. 6.4.12 does not have any backshifted terms, we conclude that the coefficients of each backshift operator must be zero, i.e.

$$\begin{aligned} \psi_0 &= 1 \\ \psi_1 - \phi_1 &= 0 \\ \psi_1 &= \phi_1 \\ \psi_2 - \phi_2 - \psi_1\phi_1 &= 0 \\ \psi_2 - \phi_2 - \phi_1^2 &= 0 \quad \text{substituting } \psi_1 = \phi_1 \\ \psi_2 &= \phi_2 + \phi_1^2 \end{aligned} \quad (6.4.13)$$

Exercise 6.4.1

Derive the first three ψ weights (i.e. ψ_0, ψ_1 , and ψ_2) for the AR(2) model in Eq. 6.4.21.

Answer available at the end of this section.

Answer available at the end of this section.

Exponential Decay of ψ Weights

While not as obvious in Eq. 6.4.13 as it had been in Eq. 6.4.3, the ψ weights of any stationary AR process will decay exponentially. Logically, this should make sense as the lack of a unit root in stationary process means that it will eventually "forget" past noise and revert to its mean value. Consequently, we can often truncate $\psi(\mathbb{B})$ at around ten to twenty terms with minimal loss in accuracy.

Fortunately, statsmodels performs the operations in Eq. 6.4.13 for us using the property `impulse_response` ^[4]. Continuing with the example from above, add the following line to the code:

```
print(f"psi weights 0-9: {ar2.impulse_response(10)}")
```

Do the results agree with Eq. 6.4.13 applied in the problem above?

Where do MA Processes Arise? Part 2

The $MA(\infty)$ representation—referred to as the impulse response or impulse response function in disciplines such as signal processing—allows us to immediately determine how long a noise (or “impulse”) continues to generate observations outside of the system’s normal behavior. In the case of an $AR(1)$ model whose $MA(\infty)$ is simply

$$\sum_{j=0}^{\infty} \phi^j w_{t-j}, \quad (6.4.14)$$

it is straightforward in both the $AR(1)$ and $MA(\infty)$ representations to determine that for, say, $\phi = \pm 0.75$, the influence of an anomalous w_t will decay to roughly 10% of its initial value after 8 timesteps, and roughly 1% after 16. For $AR(p)$ processes higher p values, extracting this information directly from the AR representation becomes far more challenging. Representing the process in its $MA(\infty)$ form allows to quickly determine how a shock will decay by examining the ψ weights. Moreover, for $p \geq 2$, there is no guarantee that the ψ weights will decay monotonically. An $AR(p)$ process with complex roots will exhibit correlations (and hence ψ weights) that decay both exponentially and sinusoidally. The sinusoidal nature will result in ψ weights that appear to reawaken at regular intervals and change the direction of their influence between positive and negative. Analyzing this decay of the ψ weights allows us to avoid being surprised when we thought a shock had completely died off.

ψ Weight Representation

Brockwell & Davis (1991) chapter 3.3 provides three broad methods to calculate the theoretical autocovariance and autocorrelation of AR models (and $ARMA$ models in general). We will focus on the first here.

The first method relies on representing the model in the form of Eq. 6.4.10. $\gamma(h)$ is then calculated in the same fashion as Eq. 6.4.5

$$\begin{aligned} \gamma(h) &= \text{Cov}(x_{t+h}, x_t) \\ &= \mathbb{E}[x_{t+h}x_t] \\ &= \mathbb{E}\left[\left(\sum_{j=0}^{\infty} \psi_j w_{t+h-j}\right)\left(\sum_{k=0}^{\infty} \psi_k w_{t-k}\right)\right] \\ &= \sigma_w^2 \sum_{i=0}^{\infty} \psi_i \psi_{i+h} \end{aligned} \quad (6.4.15)$$

where we have used the fact that the noise is iid resulting in zero covariance for different noise terms. Unfortunately, $AR(p)$ models with $p \geq 2$ do not have the same convenient geometric series representation seen with an $AR(1)$ model. Nevertheless, Eq. 6.4.15 provides an elegant method for calculating the autocovariance, and hence the autocorrelation, of any stationary AR process—or, as we shall see, any stationary $ARMA$ process. The fact that the ψ weights decay exponentially results in it being possible to truncate Eq. 6.4.15 after say, thirty terms, with relatively little loss in accuracy.

Recursion Relations

The second and third methods in Brockwell & Davis (1991) rely on using recursion relations relating higher h values to lower ones. We will not go into great depth regarding this method, but it is instructive to see how it might be applied to an $AR(2)$ model. Let us begin with the $AR(2)$ process $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$. We can multiply through by x_{t-h} and take the expectation:

$$\begin{aligned} x_t x_{t-h} &= \phi_1 x_{t-1} x_{t-h} + \phi_2 x_{t-2} x_{t-h} + w_t x_{t-h} \\ \mathbb{E}[x_t x_{t-h}] &= \mathbb{E}[\phi_1 x_{t-1} x_{t-h}] + \mathbb{E}[\phi_2 x_{t-2} x_{t-h}] + \mathbb{E}[w_t x_{t-h}] \\ \mathbb{E}[x_t x_{t-h}] &= \phi_1 \mathbb{E}[x_{t-1} x_{t-h}] + \phi_2 \mathbb{E}[x_{t-2} x_{t-h}] + \mathbb{E}[w_t x_{t-h}] \\ \gamma(h) &= \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2) + \mathbb{E}[w_t x_{t-h}]. \end{aligned} \quad (6.4.16)$$

Provided that our $AR(2)$ process is causal (stationary), we can rewrite the final term in Eq. 6.4.16 as

$$\mathbb{E}[w_t x_{t-h}] = \mathbb{E}\left[w_t \sum_{j=0}^{\infty} \psi_j w_{t-h-j}\right] = 0. \quad (6.4.17)$$

Combining Eqs. 6.4.16 and 6.4.17 we arrive at the desired recursion relation

$$\gamma(h) = \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2) \quad (6.4.18)$$

Without additional information, we cannot further simplify in Eq. 6.4.18. However, dividing through by $\gamma(0)$ generates a recursion relation for the autocorrelation

$$\rho(h) = \phi_1 \rho(h-1) + \phi_2 \rho(h-2). \quad (6.4.19)$$

We know that $\rho(0) = 1$, and we can evaluate $\rho(1)$ using Eq. 6.4.18:

$$\begin{aligned} \gamma(1) &= \phi_1 \gamma(0) + \phi_2 \gamma(-1) \\ &= \phi_1 \gamma(0) + \phi_2 \gamma(1) \\ (1 - \phi_2) \gamma(1) &= \phi_1 \gamma(0) \\ \gamma(1) &= \frac{\phi_1}{1 - \phi_2} \gamma(0) \\ \rho(1) &= \frac{\phi_1}{1 - \phi_2} \end{aligned} \quad (6.4.20)$$

Thus we can derive $\rho(2)$ as

$$\rho(2) = \frac{\phi_1^2}{1 - \phi_2} + \phi_2, \quad (6.4.21)$$

and continue extending the recursion relation to higher values of h .

Footnotes

1. The term "impulse response" comes from signal processing and denotes that fact that $\psi(\mathbb{B})$ dictates how noise, or an "impulse," decays with time. We will elaborate on this in the next subsection.

References

1. Brockwell, P. J., & Davis, R. A. (1991). Time Series: Theory and Methods. In Springer Series in Statistics. Springer New York. 10.1007/978-1-4419-0320-4

Answers

Exercise 6.4.1

Exercise 6.4.1

ψ_0 is always 1. Using Eq. 6.4.13 we have $\psi_1 = \phi_1 = 1.25$. Similarly, $\psi_2 = \phi_2 + \phi_1^2 = -0.375 + 1.25^2 = 1.1875$.

7: Appendices

The following table gives a brief overview of the notations used in this book along with their definitions and examples of how they might appear in the text.

Symbol	Meaning	Examples
Lower case unbolded letter	Scalar or realization/observation of random variable	x, α
Upper case unbolded letter	Random variable	X, Y
Lower case bolded letter	Vector	$\mathbf{x}, \boldsymbol{\alpha}$
Upper case bolded letter	Matrix	$\boldsymbol{\Phi}, \mathbf{R}$
$\triangleq$	Defined as	$\boldsymbol{\theta} \triangleq (\theta_0, \theta_1, \dots, \theta_{p-1})$
$\hat{\cdot}$ over letter	Statistical estimator	$\hat{\mu}, \hat{\sigma}^2, \hat{S}_t$
$\mathbb{E}[\cdot]$	Expectation	$\mathbb{E}[x], \mathbb{E}[(x_t - \mu_x)(y_t - \mu_y)]$
$\mathbb{P}(\cdot)$	Probability	$\mathbb{P}(x), \mathbb{P}(x_i \theta)$
$P(\cdot)$	Probability function (discrete) or probability density function (continuous)	$\sum_{x_i} P(x_i)$
$\mathbb{B}$	Backshift operator	$\mathbb{B}x_t = x_{t-1}$
$\mathbb{B}^{-1}$	Forward-shift operator	$\mathbb{B}^{-1}x_{t-1} = x_t$
Variable with subscript	Value of <i>discrete</i> variable at given index or series indexed by given variable	x_{t-2}, θ_q
Variable followed by $(\bullet)$	Value of <i>continuous</i> variable at given value or function of given variable	$x(t-2)$
Variable with subscript and superscript	Value of discrete variable at lower index <i>conditioned on</i> (i.e. given) known values up to upper index	x_t^{t-1}
$\mathbb{V}[\cdot]$ or σ^2	Variance	$\mathbb{V}[x_t], \sigma_x^2$
$\text{Cov}(\cdot, \cdot)$ or $\sigma_{\cdot, \cdot}$	Covariance	$\text{Cov}(\mathbf{X}, \mathbf{Y}), \sigma_{x,y}$
$(\operatorname{Cor}(\cdot, \cdot))$	Correlation	$\text{Cor}(\mathbf{X}, \mathbf{Y})$
$\gamma(h)$ or $\gamma_x(h)$	Autocovariance of series x at lag h	$\gamma(2), \gamma_x(0)$
$\rho(h)$ or $\rho_x(h)$	Autocorrelation of series x at lag h	$\rho(2), \rho_x(0)$
$\kappa(h)$ or $\kappa_x(h)$	Partial autocorrelation of series x at lag h	$\kappa(2)$
$\gamma_{x,y}(h)$	Cross-covariance of series x and y at lag h	$\gamma_{x,y}(2), \gamma_{x,y}(0)$
$\rho_{x,y}(h)$	Cross-correlation of series x and y at lag h	$\rho_{x,y}(2), \rho_{x,y}(0)$
$\ \cdot\ $	Vector norm	$\ \mathbf{u}\ $
$\lfloor \cdot \rfloor$	Floor function	$\lfloor \pi \rfloor = 3, \lfloor 1 \rfloor = 1$
$\lceil \cdot \rceil$	Ceiling function	$\lceil \pi \rceil = 4, \lceil 1 \rceil = 1$
$\sim$	Distributed as	$x_t \sim \chi_1^2$
$wn(\cdot, \cdot)$	White noise process or distribution	$w_t \sim wn(0, \sigma_w^2)$
$\mathcal{N}(\cdot, \cdot)$	Gaussian process or distribution	$w_t \sim \mathcal{N}(0, \sigma_w^2)$
$\Re(\cdot)$	Real portion of complex number	$\Re(z)$

$\Im(\bullet)$	Imaginary portion of complex number	$\Im(z)$
----------------	-------------------------------------	----------

INDEX

No index terms found.

This page titled [Index](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

GLOSSARY

S

Sample Word 1 | Sample Definition 1

This page titled [Glossary](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

DETAILED LICENSING

Overview

Title: Time Series Analysis for Data Scientists (Forgy)

Webpages: 47

Applicable Restrictions: Noncommercial

All licenses found:

- [CC BY-NC-SA 4.0](#): 80.9% (38 pages)
- [Undeclared](#): 19.1% (9 pages)

By Page

- Time Series Analysis for Data Scientists (Forgy) - [CC BY-NC-SA 4.0](#)
 - Front Matter - [CC BY-NC-SA 4.0](#)
 - Preface - [CC BY-NC-SA 4.0](#)
 - TitlePage - [CC BY-NC-SA 4.0](#)
 - InfoPage - [CC BY-NC-SA 4.0](#)
 - Table of Contents - [CC BY-NC-SA 4.0](#)
 - Licensing - [CC BY-NC-SA 4.0](#)
 - 1: Introduction - [CC BY-NC-SA 4.0](#)
 - 1.1: What is a Time Series? - [CC BY-NC-SA 4.0](#)
 - 1.2: What Makes Time Series Analysis Unique? - [CC BY-NC-SA 4.0](#)
 - 1.3: Time Series Analysis vs. Forecasting - [Undeclared](#)
 - 1.4: Basic Time Series Models - [Undeclared](#)
 - 1.5: Additional Resources - [Undeclared](#)
 - 2: Background Math - [CC BY-NC-SA 4.0](#)
 - 2.1: Introduction - [CC BY-NC-SA 4.0](#)
 - 2.2: Geometric Series - [CC BY-NC-SA 4.0](#)
 - 2.3: Complex Numbers - [CC BY-NC-SA 4.0](#)
 - 2.4: Properties of Variance and Covariance - [CC BY-NC-SA 4.0](#)
 - 2.5: Information Criteria - [Undeclared](#)
 - 3: Autocovariance and Autocorrelation - [CC BY-NC-SA 4.0](#)
 - 3.1: Introduction - [CC BY-NC-SA 4.0](#)
 - 3.2: Autocovariance - [CC BY-NC-SA 4.0](#)
 - 3.3: Stationarity - [Undeclared](#)
 - 3.4: Autocorrelation - [CC BY-NC-SA 4.0](#)
 - 3.5: Cross-Covariance and Cross-Correlation - [Undeclared](#)
 - 3.6: Estimation of Autocovariance, Autocorrelation, and Cross-Correlation - [CC BY-NC-SA 4.0](#)
 - 4: Methods to Enforce Stationarity - [CC BY-NC-SA 4.0](#)
 - 4.1: Introduction - [CC BY-NC-SA 4.0](#)
 - 4.2: Time Series with Trends - [CC BY-NC-SA 4.0](#)
 - 4.3: Differencing and the Backshift Operator - [CC BY-NC-SA 4.0](#)
 - 4.4: Time Series with Seasonality - [Undeclared](#)
 - 4.5: Increasing Variance and the Log Transform - [CC BY-NC-SA 4.0](#)
 - 5: Time Series Decomposition - [CC BY-NC-SA 4.0](#)
 - 5.1: Introduction - [CC BY-NC-SA 4.0](#)
 - 5.2: Classical Decomposition - [CC BY-NC-SA 4.0](#)
 - 5.3: Local Regression - [CC BY-NC-SA 4.0](#)
 - 5.4: STL Decomposition - [Undeclared](#)
 - 6: ARMA Models - [CC BY-NC-SA 4.0](#)
 - 6.1: Introduction - [CC BY-NC-SA 4.0](#)
 - 6.2: Autoregressive Models - [CC BY-NC-SA 4.0](#)
 - 6.3: Moving Average Models - [CC BY-NC-SA 4.0](#)
 - 6.4: ARMA Models - [Undeclared](#)
 - 7: Appendices - [CC BY-NC-SA 4.0](#)
 - Back Matter - [CC BY-NC-SA 4.0](#)
 - Index - [CC BY-NC-SA 4.0](#)
 - Glossary - [CC BY-NC-SA 4.0](#)
 - Detailed Licensing - [CC BY-NC-SA 4.0](#)

This page titled [Time Series Analysis for Data Scientists \(Forgy\)](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [Charles Forgy](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.