universität
wien

BACHELOR'S THESIS

Title of the Bachelor's Thesis

Time Series Analysis: Theoretical Foundations and Applications

submitted by
Stefan Krieger

in partial fulfilment of the requirements for the degree of
Bachelor of Science (BSc)

Vienna, January 23, 2025

Study Identification Number:	UA 033 621
Field of Study:	Bachelor's Program Mathematics
Supervisor:	Ass.-Prof. Dr. Julio Daniel Backhoff-Veraguas

Abstract

This thesis provides a comprehensive exploration of time series analysis, focusing on the theoretical foundations and practical applications of linear models. Core concepts such as stationarity, autocorrelation, and partial autocorrelation are examined in detail, forming the basis for the analysis of Autoregressive (AR) and Moving Average (MA) processes. Emphasis is placed on parameter estimation techniques, including the Yule-Walker equations, to highlight the nuances of modeling AR and MA processes.

The practical component includes two approaches: synthetic data generation and analysis of real-world financial data, specifically the Nikkei 225 index. Synthetic data enables controlled demonstrations of theoretical principles, offering clarity in the understanding of model behavior. In contrast, the analysis of the Nikkei 225 illustrates the complexities and limitations of applying linear models to real-world time series, such as noise and structural dependencies. These findings underline the challenges of relying solely on simple linear models for financial forecasting and suggest the need for more advanced methodologies to address such complexities effectively.

This study provides a bridge between theoretical insights and their practical applications, contributing to a nuanced understanding of time series analysis.

Acknowledgements

I would like to thank my supervisor, Dr. Julio Daniel Backhoff-Veraguas, for his guidance and constructive feedback throughout the development of this thesis. His expertise was invaluable in helping me navigate the challenges of time series analysis and ensuring the academic rigor of this work. I am deeply grateful to my friend and colleague, Khashayar Hani for his invaluable assistance and guidance throughout my journey in completing my bachelors studies and during the process of writing this thesis. His insights and feedback were instrumental in shaping this work. I would also like to thank my colleagues Mina Joksimovi and Remiziye Shemsidini for their collaborative spirit and stimulating discussions, which made my academic journey more enriching. A special thank you goes to my girlfriend, Natalija, for her unwavering support, encouragement, and understanding throughout this endeavor. Her belief in me made even the most challenging moments more manageable. Lastly, I extend my gratitude to my professors and peers in the Bachelors Program in Mathematics for cultivating an inspiring academic environment, as well as to my family and friends for their continuous encouragement and support.

Contents

1 Linear Models **1**
1.1 Stationarity . . . 2
1.2 Testing for Unit Root . . . 5
1.3 Autoregressive and Moving Average Models . . . 8
1.4 The Yule-Walker Equations . . . 10
1.5 Partial Autocorrelation Function . . . 13

2 Estimation of Linear Processes **15**
2.1 Estimation of AR(p) Models . . . 16
2.2 Estimation of MA(q) Models . . . 18

3 Applications of the Theory **19**
3.1 Analysis with Synthetic Data . . . 20
3.1.1 Motivation . . . 20
3.1.2 Generating Synthetic Data . . . 20
3.1.3 Application of the Theory . . . 21
3.1.4 Conclusion . . . 25
3.2 Analysis of the Nikkei 225 Index . . . 26
3.2.1 Data Description . . . 26
3.2.2 Application of the Theory . . . 27
3.2.3 Conclusion . . . 32

References **33**

Introduction

Time series analysis plays a critical role in understanding and modeling data that evolves over time. This thesis explores both the theoretical foundations and practical applications of time series analysis, focusing primarily on linear models such as Autoregressive (AR), Moving Average (MA), and their combination, ARMA. These models form the basis for analyzing time-dependent data and provide a framework for understanding the underlying dynamics of various processes.

The theoretical part of this work delves into key concepts such as stationarity, autocorrelation, and partial autocorrelation, which are essential for constructing and evaluating time series models. It also examines parameter estimation techniques, discussing the challenges and solutions associated with different methods, particularly for AR and MA processes.

To bridge the gap between theory and practice, this thesis applies the developed methods to both synthetic and real-world data. The synthetic data, generated under controlled conditions, serves to validate the theoretical models and demonstrate their behavior. In contrast, the analysis of the Nikkei 225 index highlights the complexities and limitations encountered when dealing with real-world financial time series. These practical applications underscore the challenges of modeling noisy and non-stationary data, emphasizing the need for advanced techniques to improve predictive performance.

Through this combination of theory and application, the thesis aims to provide insights into the strengths and weaknesses of linear time series models, as well as their practical relevance and limitations in the context of financial data analysis.

1 Linear Models

Linear models are fundamental building blocks in time series analysis. By providing a clear framework for understanding the relationship between past and present values of a series, they serve as a starting point for analyzing time-dependent data. The simplicity of linear models allows for initial insights into the dynamics of a dataset, making them a convenient first step before introducing more complex models.

In particular, linear models help to establish a foundational understanding of patterns such as trends, seasonality, and autocorrelation, which are essential in many applied fields, including economics. For example, in econometric forecasting, linear models can be used to capture the basic relationships between financial variables, such as stock prices, interest rates, or inflation rates.

Before we dive into linear models, let's go over some key definitions that are important for understanding the rest of the text. This will help clarify the concepts we will discuss later. In the following sections, the analysis and discussion will draw heavily on the foundational work presented in [1] and the insights provided in [6].

Definition 1.1 (Stochastic process)**.** A stochastic process is a sequence of random variables $(X_t)_{t\in\mathbb{Z}}$, defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In our case, we are

going to work with real valued stochastic processes, i.e. $X_t \colon \Omega \to \mathbb{R}$ is $\mathcal{F}$-$\mathcal{B}_{\mathbb{R}}$-measurable, where $\mathcal{B}_{\mathbb{R}}$ is the Borel σ-algebra on $\mathbb{R}$.

Definition 1.2 (Lag and difference operator)**.** The lag-operator L is defined as $LX_t = X_{t-1}$ and therefore $L^k X_t = X_{t-k}$.

Definition 1.3 (White noise)**.** A stochastic process $(X_t)_{t\in\mathbb{Z}}$ is called white noise, if

$$\mathbb{E}[X_t] = 0, \quad \text{and} \quad \operatorname{Cov}[X_t, X_s] = \begin{cases} \sigma^2 < \infty & \text{if } t = s, \\ 0 & \text{else.} \end{cases} \tag{1}$$

An example would be a stochastic process $(X_t)_{t\in\mathbb{Z}}$, where all X_t are independent and identically distributed (i.i.d.) with expectation 0 and finite variance.

Definition 1.4. A sequence $(X_t)_{t\in\mathbb{Z}}$ is called a linear process if it can be written as

$$X_t = c + \sum_{j=-\infty}^{\infty} a_j \varepsilon_{t-j}, \tag{2}$$

where $(\varepsilon_t)_{t\in\mathbb{Z}}$ is white noise, c and $(a_j)_{j\in\mathbb{Z}}$ are constants where $(a_j)_{j\in\mathbb{Z}}$ are absolute summable, i.e. $\sum_{j=-\infty}^{\infty} |a_j| < \infty$.

1.1 Stationarity

In most applictaions in time series analysis we only have a limited amount of observations from the stochastic process $(X_t)_{t\in\mathbb{Z}}$. Even when focusing on expected values, variances, and covariances, estimating these quantities from N data points is a task that can be addressed using standard mathematical and statistical methods. In order to make the estimation of these quantities possible, we need to make some assumptions about the underlying stochastic process. One of the most important assumptions is the stationarity of the process.

Stationarity is an essential concept in time series analysis, as many statistical models assume that a series' key properties, such as mean and variance, remain constant over time. However, many economic time series display non-stationarity, often due to trends, which must be addressed for accurate modeling. There are two primary approaches to handle non-stationarity: trend-stationary (TS) and difference-stationary (DS) models. A TS model assumes that non-stationarity arises from a predictable trend, such as a linear progression over time. By removing this trend, the series becomes stationary, meaning it fluctuates around a stable mean. In this case, external influences, known as shocks, temporarily impact the series, which eventually returns to its long-term pattern. A DS model, on the other hand, achieves stationarity by differencing the series, which involves analyzing the changes between consecutive values instead of the original values themselves. In this framework, shocks cause lasting changes to the series' values, meaning they alter the overall trajectory of the series permanently. Distinguishing between TS and DS models is vital for selecting the right approach and understanding the long-term implications of external influences on economic time series.

Next we are going to introduce the autocovariance function, which is necessary for the definition of stationarity.

Definition 1.5 (Autocovariance function)**.** The autocovariance function of a stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ is defined as

$$\gamma(t,\tau) = \mathbb{E}\left[(X_t - \mu_t)(X_{t-\tau} - \mu_{t-\tau})\right] = \mathrm{Cov}\left[X_t, X_{t-\tau}\right], \tag{3}$$

for all $t, \tau \in \mathbb{Z}$. Here μ_t denotes the mean of X_t.

A short line of computation shows that $\gamma(t,\tau) = \gamma(t-\tau, -\tau)$ and in the special case where $\tau = 0$ we get $\gamma(t,0) = \mathrm{Var}[X_t]$.

Definition 1.6. A time series $(X_t)_{t\in\mathbb{Z}}$ is called covariance stationary if the mean and the autocovariance function are time-independent, i.e.

$$\mu_t = \mu \quad \text{and} \quad \gamma(t,\tau) = \gamma(\tau), \quad \forall t, \tau \in \mathbb{Z}, \tag{4}$$

Having defined covariance stationarity, it is important to note that this is just one of several types of stationarity. While other forms, such as strict stationarity (as defined in [1], chapter 11, definition 11.6), also exist, our focus will remain on covariance stationarity. For simplicity, we will refer to covariance stationarity simply as stationarity throughout the remainder of this work.

A well-known example of a stochastic process is the random walk, which serves as a classic illustration of a non-stationary process. In the following, we will take a closer look at its properties and significance in the context of time series analysis.

Definition 1.7 (Random walk)**.** A stochastic process $(X_t)_{t\in\mathbb{Z}}$ is called a random walk, if it can be written as

$$X_t = c + X_{t-1} + \varepsilon_t \tag{5}$$

where ε_t is white noise and c is a constant. When $c \neq 0$, we call it a random walk with drift.

We can demonstrate that the random walk is not stationary, as its statistical properties, such as mean and variance, change over time. To illustrate this, we begin by calculating its key characteristics. The random walk with the initial value $X_0 = 0$ can also be rewritten as,

$$\begin{aligned} X_t = c + X_{t-1} + \varepsilon_t &= c + (c + X_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &\vdots \\ &= tc + \sum_{j=1}^{t} \varepsilon_j. \end{aligned}$$

Now we can calculate the expected value

$$\mathbb{E}\left[X_t\right] = tc + \sum_{j=1}^{t} \mathbb{E}\left[\varepsilon_j\right] = tc$$

and the autocovariance function,

$$\begin{aligned}
\gamma(t,\tau) &= \operatorname{Cov}\left[tc + \sum_{j=1}^{t}\varepsilon_j, (t-\tau)c + \sum_{i=1}^{t-\tau}\varepsilon_i\right] \\
&= \operatorname{Cov}\left[tc, (t-\tau)c\right] + \operatorname{Cov}\left[tc, \sum_{i=1}^{t-\tau}\varepsilon_i\right] && \text{(bilinearity)} \\
&\quad + \operatorname{Cov}\left[\sum_{j=1}^{t}\varepsilon_j, (t-\tau)c\right] + \operatorname{Cov}\left[\sum_{j=1}^{t}\varepsilon_j, \sum_{i=1}^{t-\tau}\varepsilon_i\right] \\
&= \sum_{j=1}^{t}\sum_{i=1}^{t-\tau}\operatorname{Cov}\left[\varepsilon_j, \varepsilon_i\right] && \text{(white noise)} \\
&= (t-\tau)\sigma^2,
\end{aligned}$$

hence the random walk is not stationary, because both functions are time dependent.

It is worth mentioning that the random walk is a key concept in finance, illustrating the unpredictable nature of asset prices, where past trends cannot reliably predict future movements. It is also fundamental in physics, modeling phenomena like Brownian motion, and in mathematics, serving as the basis for stochastic processes such as Markov chains. Later, we will derive the autoregressive (AR) process from this framework.

Definition 1.8. A time series $(X_t)_{t\in\mathbb{Z}}$ is called an integrated process of order $d \in \mathbb{N}$, denoted by $I(d)$, if there exists a minimal d, such that $(1-L)^{d-1}X_t$ is non-stationary, but $(1-L)^d X_t$ is.

Having defined integrated processes, we now demonstrate that the random walk is an integrated process of order 1. By construction after differencing we obtain:

$$X_t - X_{t-1} = (1-L)X_t = c + \varepsilon_t, \tag{6}$$

which is a stationary process, because $\mathbb{E}\left[(1-L)X_t\right] = c$ and $\gamma(t,\tau) = 0$ for $\tau \neq 0$.

Figure 1 illustrates the distinction between stationary and non-stationary time series. The top two graphs display the stock prices of CoreCivic and Nvidia, obtained using python and the yfinance package. Both of which demonstrate non-stationary behavior due to their upward trends. The bottom graph presents a random walk, a classic example of a non-stationary process, alongside its differenced version, which serves as an example of a stationary time series. These examples demonstrate the different characteristics of stationary and non-stationary processes, highlighting the need for appropriate transformations when analyzing non-stationary time series. Understanding these distinctions ensures that statistical methods are applied correctly, avoiding biased results or misleading conclusions.

While stationarity is crucial for many theoretical and practical applications, non-stationary processes are equally important across various disciplines. They capture real-world phenomena such as evolving systems, growth patterns, and structural changes,

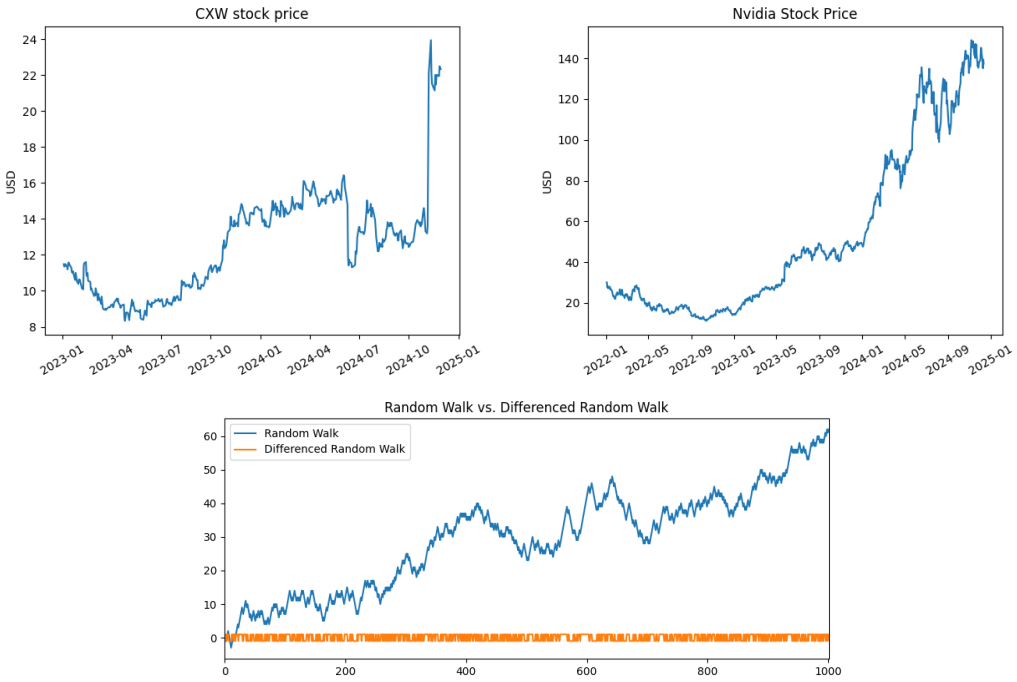

Figure 1: Examples of non-stationary and stationary time series

providing insights that stationary models often overlook. Non-stationary data contains valuable information about long-term trends, cyclical behaviors, and regime shifts, which are critical for tasks like forecasting, analyzing relationships between variables, and understanding dynamic systems. Despite the challenges they pose, these processes enable more realistic modeling in fields such as economics, environmental science, engineering, and social studies.

1.2 Testing for Unit Root

The Unit Root Test is an important tool in time series analysis, particularly for determining whether a time series is stationary or non-stationary. The presence of a unit root indicates a non-stationary process, which has significant implications for modeling and forecasting. As we have discussed, non-stationary processes can exhibit trends or structural shifts, and it is essential to identify this behavior before applying certain modeling techniques. I will introduce this concept with the help of an example. The following example can be found in [9] on page 39.

The process

$$X_t = \alpha X_{t-1} + \varepsilon_t \tag{7}$$

with ε_t being white noise, is called autoregressive process of order 1 or short AR(1). We also assume that $X_0 = 0$ and ε_t exists for $t \in \mathbb{Z}$. Through resubstituting X_{t-1} into (7)

we get

$$X_t = \alpha(\alpha X_{t-2} + \varepsilon_{t-1}) + \varepsilon_t. \tag{8}$$

Doing this k times we arrive at

$$X_t = \alpha^k X_{t-k} + \sum_{j=0}^{k-1} \alpha^j \varepsilon_{t-j}. \tag{9}$$

Under the assumption that $\mathbb{E}\left[X_t^2\right] < \infty$ and $|\alpha| < 1$, we have that

$$\lim_{k\to\infty} \mathbb{E}\left[\left(X_t - \sum_{j=0}^{k} \alpha^j \varepsilon_{t-j}\right)^2\right] = 0.$$

Therefore, we can rewrite (9) as

$$X_t = \sum_{j=0}^{\infty} \alpha^j \varepsilon_{t-j}$$

and we can see that $\mathbb{E}\left[X_t\right] = 0$ for all t and

$$\gamma(t,\tau) = \mathrm{Cov}\left[\sum_{l=0}^{\infty} \alpha^l \varepsilon_{t-l}, \sum_{k=0}^{\infty} \alpha^k \epsilon_{t-(\tau+k)}\right] \tag{10}$$

$$= \sum_{l=0}^{\infty}\sum_{k=0}^{\infty} \alpha^l \alpha^k \mathrm{Cov}\left[\varepsilon_{t-l}, \varepsilon_{t-(\tau+k)}\right] \tag{11}$$

$$= \sum_{l=0}^{\infty}\sum_{k=0}^{\infty} \alpha^{k+l} \sigma^2 \delta_{l,\tau+k} \tag{12}$$

$$= \sum_{k=0}^{\infty} \alpha^{\tau+2k} \sigma^2 \tag{13}$$

$$= \sigma^2 \alpha^\tau \sum_{k=0}^{\infty} (\alpha^2)^k = \frac{\alpha^\tau \sigma^2}{1-\alpha^2}. \tag{14}$$

So we see that both functions are essentially time independent, hence stationary. Sometimes an $AR(1)$ process is also called stable when $|\alpha| < 1$. In the case where $|\alpha| > 1$, the process is called explosive and is not that interesting because the shocks $1 < \alpha < \alpha^2 < \alpha^3 < \dots$ diverge over time.

In the following, I intend to make a brief excursion into functional analysis to provide a deeper theoretical foundation for understanding the unit root test. This approach also offers an alternative perspective to demonstrate why $|\alpha| < 1$ is necessary for stationarity. The following section is inspired by the work presented in [6]. To proceed, we will require a fundamental theorem from functional analysis. Recall that the norm of an operator A is defined as follows

$$\|A\| = \sup_{\|x\|\neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|.$$

Theorem 1.9. *[see [4], Thm. 4.40] Let* $\mathbf{X}$ *be a Banach space and* $A\colon \mathbf{X} \to \mathbf{X}$ *a bounded linear operator. If* $\|A\|_{\mathbf{X}} < 1$ *then* $\mathbb{I} - A$ *is invertible with,*

$$(\mathbb{I} - A)^{-1} = \sum_{j=0}^{\infty} A^j$$

where $\mathbb{I}$ *is the identity operator and* $\|.\|_{\mathbf{X}}$ *the operator norm.*

Proof. A quick line of computation shows that the series is finite

$$\left\| \sum_{j=0}^{\infty} A^j \right\| \leq \sum_{j=0}^{\infty} \|A^j\| \leq \sum_{j=0}^{\infty} \|A\|^j < \infty$$

The last inequality follows directly from the condition that $\|A\| < 1$ and therefore $\sum_{j=0}^{\infty} A^j$ converges as well (see Theroem 2.30 in [4]). Now we can check if the sum is the inverse of $\mathbb{I} - A$ by a simple calculation.

$$(\mathbb{I} - A) \sum_{j=0}^{\infty} A^j = \sum_{j=0}^{\infty} A^j - \sum_{j=0}^{\infty} A^{j+1} = A^0 = \mathbb{I}$$

Analog for the left inverse. Hence,

$$\sum_{j=0}^{\infty} A^j = (\mathbb{I} - A)^{-1}$$

□

To make use of 1.9, we need to further specify the framework in which we are working. The lag operator L is a deterministic operator, meaning it acts on a realization of a stochastic process. In our context, this realization takes the form of a sequence of real numbers. We assume that the resulting sequence is an elemeny of ℓ^p, the space of sequences, with $p \in [1, \infty]$. We already know that these spaces form a Banach space with the norm $\|x\|_p = \left(\sum_{j=0}^{\infty} |x_j|^p\right)^{1/p}$. For the remainder of this thesis, we will work specifically with ℓ^1.

Next we want to rewrite 7 with the help of the lag operator L. This yields

$$(\mathbb{I} - \alpha L) X_t = \epsilon_t \tag{15}$$

and with 1.9 we know that, $\mathbb{I} - \alpha L$ in invertible when $\|\alpha L\| < 1$. First let us compute the operator norm of L

$$\|L\| = \sup_{\|X\| \leq 1} \|LX\| = \sup_{X \in \mathbf{X}\colon \sup_t \mathbb{E}[|X_t|] \leq 1} \left(\sup_t \mathbb{E}\left[|X_{t-1}|\right] \right) = 1$$

With this result we are able to calculate

$$\|\alpha L\| = |\alpha| \|L\| = |\alpha|$$

So if $|\alpha| < 1$, then $\mathbb{I} - \alpha L$ is invertible and the inverse is given through 1.9 as

$$(\mathbb{I} - \alpha L)^{-1} = \sum_{j=0}^{\infty} \alpha^j L^j$$

Now we can apply this to (15) and get

$$(\mathbb{I} - \alpha L)^{-1}(\mathbb{I} - \alpha L)X_t = (\mathbb{I} - \alpha L)^{-1}\epsilon_t \tag{16}$$

$$X_t = \sum_{j=0}^{\infty} \alpha^j L^j \epsilon_t \tag{17}$$

We already showed that this process is stationary. To close the connection between the previous results and the unit circle we can look at the lag polynomial $\varphi(L) \coloneqq \mathbb{I} - \alpha L$ as a complex polynomial and get $\varphi(z) = 1 - \alpha z$. The root of $\varphi(z)$ is $\frac{1}{\alpha}$. The fact that we need $|\alpha| < 1$ implies that $|z| > 1$. Hence, the root of the polynomial must lie outside the unit circle. Later we will generalize this result for $AR(p)$ processes.

The name "unit root test", also known as the Dickey-Fuller test ([1], chapter 11.6.1), arises from its focus on the roots of the characteristic polynomial associated with the time series model. Specifically, we use a hypothesis test to determine whether a unit root exists. Therefore, we can define the null and alternative hypotheses as follows:

$$H_0 : \alpha = 1 \text{ and } H_1 : \alpha < 1.$$

1.3 Autoregressive and Moving Average Models

Next we are going to introduce some popular linear models in time series analysis, namely the autoregressive (AR) and the moving average (MA). These models are widely used in practice due to their simplicity and effectiveness in capturing the dynamics of time series data. By combining these models, we can create the autoregressive moving average (ARMA) model, which provides a more comprehensive framework for analyzing time-dependent data. The following definitions are based on [1].

Definition 1.10. A moving average model of order q, denoted by MA(q), is defined as

$$X_t = \beta_0\varepsilon_t + \beta_1\varepsilon_{t-1} + \cdots + \beta_q\varepsilon_{t-q} = \sum_{j=0}^{q} \beta_j\varepsilon_{t-j}, \tag{18}$$

where $\beta_0, \ldots, \beta_q$ are constants and ε_t is white noise.

We can rewrite the MA(q) model in terms of the lag operator L as $X_t = \beta(L)\epsilon_t$ with $\beta(L) = \beta_0 + \beta_1 L + \cdots + \beta_q L^q$. Since the MA($q$) model is a linear combination of white noise, it is always stationary since $\mathbb{E}[X_t] = 0$ and the autocovariance function

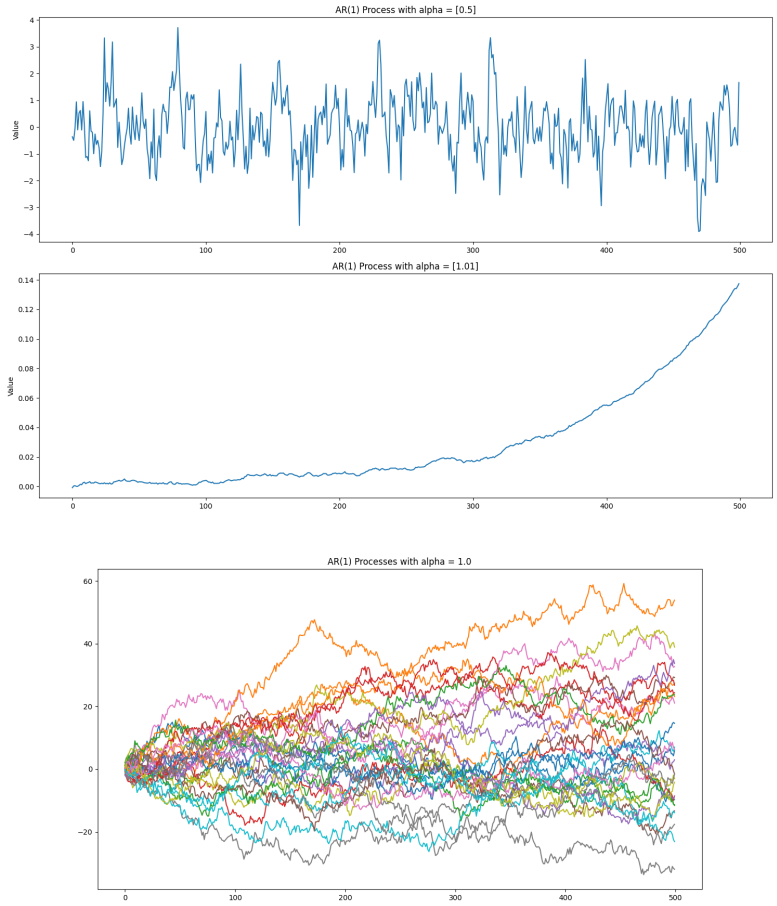

Figure 2: Examples of AR(1) processes for different values of α

is time-independent. The latter can be seen by computing the autocovariance function which is given by

$$\begin{aligned}
\gamma(t,\tau) &= \text{Cov}\left[X_t, X_{t-\tau}\right] \\
&= \text{Cov}\left[\sum_{j=0}^{q}\beta_j\varepsilon_{t-j}, \sum_{k=0}^{q}\beta_k\varepsilon_{t-(\tau+k)}\right] \\
&= \sum_{j=0}^{q}\sum_{k=0}^{q}\beta_j\beta_k\text{Cov}\left[\varepsilon_{t-j}, \varepsilon_{t-(\tau+k)}\right] \\
&= \sum_{j=0}^{q-\tau}\beta_j\beta_{j-\tau}\sigma^2, \text{ for } \tau \leq q
\end{aligned}$$

which is also time independent.

Definition 1.11. An autoregressive model of order p, denoted by AR(p), is defined as

$$X_t = c + \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \varepsilon_t, \tag{19}$$

where $c, \alpha_1, \ldots, \alpha_p$ are constants and ε_t is white noise.

The AR(p) model can also be rewritten in terms of the lag operator L as $\alpha(L)X_t = c + \epsilon_t$ with the lag polynomial $\alpha(L) = 1 - \alpha_1 L - \cdots - \alpha_p L^p$. As previously discussed in Section 1.2, we can substitute the operator L with a complex number z. From the fundamental theorem of algebra, we know that $\alpha(z)$ has p roots. Therefore, the polynomial can be decomposed into its linear factors.

$$\alpha(z) = (1 - \alpha_1 z)(1 - \alpha_2 z) \cdots (1 - \alpha_p z). \tag{20}$$

Now we can make use of the theory we developed in the section before and finally generalize the unit root test for $AR(p)$ processes. Furthermore, we know that the space of bounded operators $\mathfrak{L}(\ell^{\mathbf{1}})$ form a Banach algebra (see [5], page 506) and thus we can use the following property

Lemma 1.12. *Let $A_j \in \mathfrak{L}(\ell^{\mathbf{1}})$ for $j = 1, \ldots, p$ then $A = A_1 A_2 \cdots A_p$ is invertible if and only if A_j is invertible $\forall\, j \leq p$.*

Proof. This follows directly form the properties of a Banach algebra. □

Combining 1.12 with (20) and the results from section 1.2, we can see that the $AR(p)$ process is stationary if and only if $|\alpha_i| < 1$ for all $i \leq p$. Now we know there exists $\alpha^{-1}(L)$ such that $\alpha(L)\alpha^{-1}(L) = 1$ and

$$\alpha^{-1}(L) = a_0 + a_1 L + \cdots = \sum_{i=0}^{\infty} a_i L^i,$$

if all the roots of the polynomial lie outside the unit circle, i.e. $\alpha(z) \neq 0$ for $|z| \leq 1$. Multpliying $\alpha(L)X_t = c + \varepsilon_t$ with $\alpha^{-1}(L)$ yields $X_t = \alpha^{-1}(1)c + \alpha^{-1}(L)\varepsilon_t$. Hence, the $AR(p)$ can be written as a $MA(\infty)$ process,

$$X_t = \sum_{i=0}^{\infty} a_i c + \sum_{i=0}^{\infty} a_i L^i \varepsilon_t$$

which is stationary.

1.4 The Yule-Walker Equations

The Yule-Walker equations play an important role in the analysis of autoregressive processes. They provide a simple method to calculate or "invert" the autocovariance function when the AR parameters are known. Specifically, they establish a relationship between the AR model parameters and the autocovariance at different lags. This relationship is instrumental in calculating the autocorrelation function (ACF), a core element in time

series analysis that captures the dependencies in data. Moreover, the Yule-Walker equations are widely used to estimate the AR parameters directly from the autocorrelation or autocovariance values observed in empirical data. Thus, they provide a convenient method for moving between model parameters and autocorrelation values. Before we derive the Yule-Walker equations, we will define the autocorrelation function. For the following I will rely on [1] and [6].

Definition 1.13. The autocorrelation function (ACF) of a stochastic process $(X_t)_{t\in\mathbb{Z}}$ is defined as

$$\rho(t,\tau) = \frac{\gamma(t,\tau)}{\sqrt{\gamma(t,0)\gamma(t-\tau,0)}}, \tag{21}$$

for all $t, \tau \in \mathbb{Z}$. For stationary processes we can write

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)}, \tag{22}$$

beacause these kind of processes are time independent. This function basically measures the linear relationship between X_t and $X_{t-\tau}$, as it represents the correlation between these two variables.

The autocorrelation function (ACF) can, at least theoretically, be used to determine the order of a MA process. Specifically, the ACF becomes zero for lags greater than the order of the MA process.

Lemma 1.14. *The autocorrelation function $\rho(\tau)$ of a MA(q) process is zero for $\tau > q$.*

Proof. We already know that MA processes are stationary and $\gamma(\tau) = \sum_{j=0}^{q-\tau} \beta_j \beta_{j-\tau} \sigma^2$, hence

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \frac{\sum_{j=0}^{q-\tau} \beta_j \beta_{j-\tau} \sigma^2}{\mathrm{Var}[\sum_{j=0}^{q} \beta_j \varepsilon_{t-j}]} = \frac{\sum_{j=0}^{q-\tau} \beta_j \beta_{j-\tau} \sigma^2}{\sum_{j=0}^{q} \beta_j^2 \sigma^2} = 0 \quad \text{for } \tau > q. \tag{23}$$

□

In Figure 3, the ACF for MA(1) and MA(3) processes is shown. These plots were created using Python and the 'statsmodels' library. The ACF for the MA(1) process shows a sharp decline after the first lag, reaching zero for lags greater than one. This pattern is consistent with the theoretical result that the ACF is zero for lags greater than the order of the MA process. Similarly, the ACF for the MA(3) process drops to zero after the third lag, reflecting the order of the MA process. These examples demonstrate how the ACF can help identify the order of an MA process. The blue bar around the x-axis represent the 95% confidence intervals for the autocorrelations at each lag. Lags where the correlation values extend beyond the blue bars indicate statistically significant correlations, meaning the correlation is likely not due to random noise. Lags where the correlation values lie within the blue bars suggest that the correlation is not significant, implying that the observed correlation is likely due to random variation in the data.

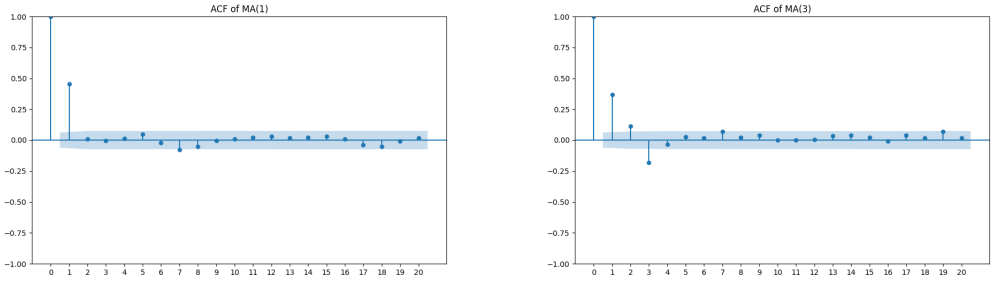

Figure 3: Visualization of ACF for MA(1) and MA(3) processes

Now we are ready for the Yule-Walker equations for a stationary AR(p) process where $c = 0$. First lets multiply the equation by $X_{t-\tau}$ and take the expacted value, which yields

$$\mathbb{E}\left[X_t X_{t-\tau}\right] = \alpha_1 \mathbb{E}\left[X_{t-1} X_{t-\tau}\right] + \cdots + \alpha_p \mathbb{E}\left[X_{t-p} X_{t-\tau}\right] + \mathbb{E}\left[\epsilon_t X_{t-\tau}\right]. \tag{24}$$

and noting that $\mathbb{E}[\epsilon_t X_{t-\tau}] = 0$ for $\tau = 1, \ldots, p$ (since ϵ_t is uncorrelated with past values of X_t), we simplify the equation to:

$$\gamma(\tau) = \alpha_1 \gamma(\tau - 1) + \alpha_2 \gamma(\tau - 2) + \cdots + \alpha_p \gamma(\tau - p).$$

Here, $\gamma(\tau) = \mathbb{E}[X_t X_{t-\tau}]$ is the autocovariance function, and this holds for each lag $\tau = 1, \ldots, p$. Now, rewrite this system of equations for all lags:

$$\begin{aligned}
\gamma(1) &= \alpha_1 \gamma(0) + \alpha_2 \gamma(1) + \cdots + \alpha_p \gamma(p-1), \\
\gamma(2) &= \alpha_1 \gamma(1) + \alpha_2 \gamma(0) + \cdots + \alpha_p \gamma(p-2), \\
&\vdots \\
\gamma(p) &= \alpha_1 \gamma(p-1) + \alpha_2 \gamma(p-2) + \cdots + \alpha_p \gamma(0).
\end{aligned}$$

To normalize the equations, we divide by $\gamma(0)$, the variance of X_t. Since $\gamma(0) > 0$, division is valid; $\gamma(0) = 0$ would imply that the process has no variability, reducing it to a constant process. This yields:

$$\rho(\tau) = \alpha_1 \rho(\tau - 1) + \alpha_2 \rho(\tau - 2) + \cdots + \alpha_p \rho(\tau - p),$$

where $\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)}$ is the autocorrelation function. In matrix form, this becomes:

$$\begin{pmatrix} 1 & \rho(1) & \ldots & \rho(p-1) \\ \rho(1) & 1 & \ldots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \ldots & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{pmatrix}.$$

Thus, the Yule-Walker equations for an AR(p) process can be written in their final matrix form:

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b},$$

where $\mathbf{A}$ is the autocorrelation matrix, $\boldsymbol{\alpha}$ is the vector of AR coefficients, and $\mathbf{b}$ is the vector of autocorrelations at lags $1, \ldots, p$.

Consequently, the structure of an autoregressive process of order p can be described in two equivalent ways: through a parametric representation using the parameters $\alpha_1, \ldots, \alpha_p$, or through a non-parametric representation involving the first p autocorrelation coefficients $\rho(1), \ldots, \rho(p)$. Both representations convey the same information, and the choice of representation depends on the context. For autoregressive processes of finite and known order, the parametric representation is typically preferred.

1.5 Partial Autocorrelation Function

The partial autocorrelation function (PACF) is an essential tool in time series analysis because it helps distinguish direct relationships in a series from indirect effects through intervening lags. While the autocorrelation function (ACF) captures the correlation of observations with all previous lags, the PACF isolates the "pure" or direct correlation between an observation and a specific lag, removing the influence of intermediate terms. This distinction is especially valuable for identifying the appropriate order of an autoregressive (AR) model. By examining the PACF, one can determine the number of significant lags needed to explain the series, as the partial autocorrelations will typically drop off after the true AR order. Thus, the PACF provides crucial insights into the underlying structure of a time series, enabling more accurate model selection and deeper understanding of data dependencies. For the following I will rely on [1] and [6].

The partial autocorrelation coefficient can be introduced using the following notation:

$$X_t = c + \phi_{p1}X_{t-1} + \phi_{p2}X_{t-2} + \cdots + \phi_{pp}X_{t-p} + \varepsilon_t,$$

where ϕ_{pi} denotes the coefficient of the variable at lag i in a process of order p. This formulation sets the groundwork for defining the partial autocorrelation.

Definition 1.15. The partial autocorrelation function of a stationary stochastic process $(X_t)_{t\in\mathbb{Z}}$ is defined as

$$\phi_{pp} = Corr(X_t - \mathbb{E}\left[X_t|X_{t-1}, \ldots, X_{t-p+1}\right], X_{t-p} - \mathbb{E}\left[X_{t-p}|X_{t-1}, \ldots, X_{t-p+1}\right]), \quad (25)$$

for all $t \in \mathbb{Z}$. The partial autocorrelation function captures the direct relationship between two variables, X_t and $X_{t-\tau}$, while controlling for the influence of intermediate terms. To see that the PACF is well defined and independent of t, we can rewrite $\mathbb{E}\left[Y|Z\right] = \Sigma_{YZ}\Sigma_{ZZ}^{-1}$ with $\Sigma_{ZZ} = \mathrm{Var}[Z]$ as the covariance matrix and $\Sigma_{YZ} = \mathrm{Cov}\left[Y, Z\right]$ as the covariances between Y and Z. Due to the stationarity of the process $(X_t)_{t\in\mathbb{Z}}$, the PACF is time independent.

We can also define the PACF with the help of the Yule-Walker equations. Consider the following system of equations

$$\begin{pmatrix} 1 & \rho(1) & \dots & \rho(p-1) \\ \rho(1) & 1 & \dots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \dots & 1 \end{pmatrix} \begin{pmatrix} \phi_{p1} \\ \phi_{p2} \\ \vdots \\ \phi_{pp} \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{pmatrix}. \tag{26}$$

These are the Yule Walker equations (see 1.4) for an AR(p) model. To get the PACF we are only interested in ϕ_{pp}. Luckily with Cramer's rule (see [7] chapter 7, page 240) we are getting excactly this element through

$$\phi_{pp} = \frac{\begin{vmatrix} 1 & \rho(1) & \dots & \rho(1) \\ \rho(1) & 1 & \dots & \rho(2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \dots & \rho(p) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \dots & \rho(p-1) \\ \rho(1) & 1 & \dots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \dots & 1 \end{vmatrix}} = \frac{\det J_p^{(p)}}{\det J_p} \tag{27}$$

where $J_p^{(p)}$ is the matrix J_p with the p-th column replaced by $(\rho(1), \dots, \rho(p))^\top$.

As observed for MA processes, the autocorrelation function (ACF) becomes zero for lags greater than the order of the process. For AR processes, we encounter an equivalent behavior,

Corollary 1.16 (see Corollary 2.6.1.2. in [9])**.** *The partial autocorrelation function ϕ_{kk} of a stationary AR(p) process is zero for $k > p$.*

This is a very useful property because it allows us to determine the order of the process by looking at the PACF. Let us look at a stationary AR(1) process. From (14) and $\text{Var}[X_t] = \frac{\sigma^2}{1-\alpha^2}$ we get that $\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \frac{\frac{\alpha^\tau \sigma^2}{1-\alpha^2}}{\frac{\sigma^2}{1-\alpha^2}} = \alpha^\tau$ and hence

$$\phi_{11} = \rho(1) = \alpha \neq 0$$

as we only consider non-degenerated AR processes, but

$$\phi_{22} = \frac{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & \rho(2) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{vmatrix}} = \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} = 0$$

because $\rho(2) = \rho(1)^2$.

Figure 4 illustrates the PACF for AR(2) and AR(4) processes. For the AR(2) process, the PACF shows a sharp decline after the second lag, becoming zero for lags beyond two. This behavior aligns with the theoretical property that the PACF vanishes for lags greater than the order of the AR process. Likewise, the PACF for the AR(4) process drops to zero after the fourth lag, consistent with the process order. These examples highlight how the PACF can be used to determine the order of an AR process.

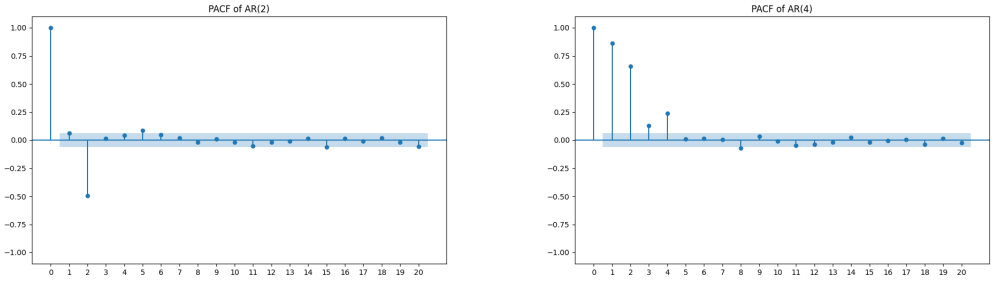

Figure 4: Visualization of PACF for AR(2) and AR(4) processes

We have now explored a theoretical approach to determining the order of a MA and a stationary AR process. But what if the ACF and PACF do not break off as expected? In such cases, we can turn to the autoregressive moving average (ARMA) model, which combines the properties of AR and MA processes. The ARMA model is particularly useful for capturing time series dynamics that cannot be adequately represented by a purely AR or MA process alone.

Definition 1.17. An autoregressive moving average model of order (p, q), denoted by ARMA(p, q), is defined as

$$X_t = c + \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \beta_0 \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q}, \tag{28}$$

where $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$ are constants, ε_t is white noise and c is a constant. It can also be written in terms of the moving average and the autoregressive lag polynomial, $\alpha(L)X_t = c + \beta(L)\varepsilon_t$.

Stationarity of the ARMA(p,q) model can be shown by the same argument as for the AR(p) model. If the roots of the characteristic equation of the autoregressive part are outside the unit circle, then $\alpha(L)$ has an inverse $\alpha^{-1}(L)$ and therefore, 28 can be written as $X_t = \alpha^{-1}(L)\beta(L)\varepsilon_t$, which is a MA$(\infty)$ process and hence stationary if $c = 0$.

2 Estimation of Linear Processes

The following section relies on cahpter 1.3 from [6]. For statistical inference with time series, certain assumptions are necessary because in practice we typically only observe a single realization of the process generating the time series. While a stochastic process

is theoretically described by a multidimensional distribution function, this approach is impractical in many situations. Instead, assumptions like ergodicity and stationarity are required to analyze time series data. Ergodicity ensures that sample moments derived from a finite time series converge to the moments as the sample size $N \to \infty$.

For a process where $\mathbb{E}[X_t] = \mu$ and $\mathrm{Var}[X_t] = \sigma^2$ are constant, ergodicity can be characterized as follows:

1. Mean ergodicity:

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\frac{1}{N}\sum_{j=1}^{T} x_j - \mu\right)^2\right] = 0,$$

which ensures the sample mean converges to the mean.

2. Variance ergodicity:

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\frac{1}{N}\sum_{j=1}^{N} (x_j - \mu)^2 - \sigma^2\right)^2\right] = 0,$$

indicating that the sample variance approaches the true variance. Here x_j denotes a realisations of X_j, i.e. $X_j(\omega) = x_j$, $\omega \in \Omega$.

These properties, while fundamental for dependent random variables, cannot be empirically verified and must therefore be assumed. Moreover, for a stochastic process to be ergodic, it must satisfy the condition of statistical equilibrium, which requires the process to be stationary. Due to this assumptions we can use the well known estimators in our case:

$$\hat{\mu} \coloneqq \frac{1}{N}\sum_{j=1}^{N} x_j, \qquad \text{(sample mean)}$$

$$\hat{\gamma}(0) = \hat{\sigma}^2 \coloneqq \frac{1}{N-1}\sum_{j=1}^{N} (x_j - \hat{\mu})^2, \qquad \text{(sample variance)}$$

$$\hat{\gamma}(\tau) \coloneqq \frac{1}{N}\sum_{j=1}^{N-\tau} (x_j - \hat{\mu})(x_{j+\tau} - \hat{\mu}), \quad \tau \leq N-1 \qquad \text{(sample autocovariance)}$$

$$\frac{\hat{\gamma}(\tau)}{\hat{\gamma}(0)} = \hat{\rho}(\tau) \coloneqq \frac{\frac{1}{N}\sum_{j=1}^{N-\tau} (x_j - \hat{\mu})(x_{j+\tau} - \hat{\mu})}{\frac{1}{N}\sum_{j=1}^{N} (x_j - \hat{\mu})^2}. \qquad \text{(sample autocorrelation)}$$

2.1 Estimation of AR(p) Models

The following subsection is based on chapter 12.7 from [1] and chapter 2.1.5 from [6]. Now we are going to estimate the parameters of an AR(p) process. For this purpose, we

use the Yule-Walker equations where we replace the theoretical autocorrelation function with the sample autocorrelation function. Therefore, we get

$$\begin{pmatrix} 1 & \hat{\rho}(1) & \cdots & \hat{\rho}(p-1) \\ \hat{\rho}(1) & 1 & \cdots & \hat{\rho}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\rho}(p-1) & \hat{\rho}(p-2) & \cdots & 1 \end{pmatrix} \begin{pmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_p \end{pmatrix} = \begin{pmatrix} \hat{\rho}(1) \\ \hat{\rho}(2) \\ \vdots \\ \hat{\rho}(p) \end{pmatrix}. \tag{29}$$

In short 29 can be written in matrix form $A\hat{\alpha} = \hat{\rho}$ and can be solved using the well known approach of linear least squares,

$$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^p}{\arg\min} \sum_{j=0}^{p} \left(\rho_j - \sum_{k=0}^{p} \alpha_k a_{jk} \right)^2, \text{ with } \hat{\alpha} = (A^\top A)^{-1} A^\top \rho.$$

This solution is well known and can be found in [8] on page 332 (remark 12.14).

We already have seen how to theoretically estimate the order of an AR(p) process, namely through the PACF. Now we are going to see how to estimate the order of the process in practice. This can be achieved using information criteria by estimating AR models incrementally for increasing orders $p = 1, 2, \ldots, p_{\max}$. The optimal order is identified as the one that minimizes the chosen criterion. Commonly used criteria for this purpose include the following:

1. Akaike Information Criterion (AIC):

$$\mathrm{AIC}(k) = -2 \log \left(\frac{1}{N-k} \sum_{j=k+1}^{N} \hat{\varepsilon}_j^2 \right) + k \frac{2}{N}, \tag{30}$$

2. Bayesian Information Criterion (BIC):

$$\mathrm{BIC}(k) = -2 \log \left(\frac{1}{N-k} \sum_{j=k+1}^{N} \hat{\varepsilon}_j^2 \right) + k \frac{\log(N)}{N}, \tag{31}$$

3. Hannan-Quinn Information Criterion (HQIC):

$$\mathrm{HQIC}(k) = -2 \log \left(\frac{1}{N-k} \sum_{j=k+1}^{N} \hat{\varepsilon}_j^2 \right) + k \frac{2 \log(\log(N))}{N}. \tag{32}$$

Where:

- $\hat{\varepsilon}$ are the estimated residuals.
- k is the number of parameters estimated in the model.
- N is the number of observations.

These criteria focus on different aspects of model selection, balancing fit and complexity in distinct ways. AIC prioritizes prediction accuracy by favoring models with a better fit, even at the cost of additional complexity. This can lead to overfitting, especially with small datasets. BIC, in contrast, applies a stricter penalty for model complexity, particularly as the sample size grows, making it more suitable for selecting simpler, more interpretable models. HQC strikes a balance between AIC and BIC, with a penalty that grows more moderately, making it a useful compromise in cases where neither extreme complexity nor excessive simplicity is ideal.

We are going to demonstrate the order estimation of an AR process using the methods described before. Suppose we have data points $(x_t)_{t=1}^N$ and assume that is was generated by an AR process. First we are going to choose an arbitrary order $p^\star$. The selection of the order can be guided by analyzing the PACF. The model is then given by

$$x_t = c + \sum_{j=1}^{p^\star} \alpha_j x_{t-j} + \varepsilon_t, \quad \text{and} \quad \varepsilon_t \sim N(0, \sigma^2).$$

Now we define the following ordinary least squares problem

$$\arg\min_{v \in \mathbb{R}^{p+1}} \|x - Av\|^2$$

where $x = (x_{p^\star+1}, \ldots, x_N)^\top$, $v = (c, \alpha_1, \ldots, \alpha_{p^\star})^\top$ and

$$A = \begin{pmatrix} 1 & x_{p^\star} & \cdots & x_1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N-1} & \cdots & x_{N-p^\star} \end{pmatrix}.$$

Assuming A has full rank we get the following solution for the LLS problem

$$\hat{v} = (A^\top A)^{-1} A^\top x.$$

The estimated parameters are then given by $\hat{c} = \hat{v}_1$ and $\hat{\alpha}_i = \hat{v}_{i+1}$ for $i = 1, \ldots, p^\star$. The estimated residuals are given by

$$\hat{\varepsilon}_t = x_t - \left(\hat{c} + \sum_{j=1}^{p^\star} \hat{\alpha}_j x_{t-j} \right),$$

for $t = p^\star+1, \ldots, N$. The information criteria can then be calculated using the estimated residuals. The order of the AR process is then chosen as the one that minimizes the information criterion. This will be illustrated in an application; see Chapter 3.

2.2 Estimation of MA(q) Models

The following subsection is based on chapter 5.4 from [3]. The estimation of MA(q) processes is more challenging than for AR(p) processes, which can already be seen for

$q = 1$. An MA(1) process is given by $X_t = c + \beta\varepsilon_{t-1} + \varepsilon_t$, where $\varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$. Suppose we have observations $(x_t)_{t=1}^N$ generated by this process. We now want to estimate the parameter $\theta = (c, \beta, \sigma^2)$.

Suppose we know that $\varepsilon_0 = 0$ then $X_1|\varepsilon_0 = c + \varepsilon_1 \sim N(c, \sigma^2)$, therefore $\varepsilon_1 = x_1 - c$. Moreover $X_2 = c + \beta\varepsilon_1 + \varepsilon_2$, so $\varepsilon_2 = x_2 - c - \beta\varepsilon_1$ with the conditional density function

$$f_{X_2|X_1,\varepsilon_0=0}(x_2|x_1, \varepsilon_0 = 0; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_2 - c - \beta\varepsilon_1)^2}{2\sigma^2}\right). \tag{33}$$

Through iteration we get

$$f_{X_t|X_{t-1},\ldots,X_1,\varepsilon_0=0}(x_t|x_{t-1}, \ldots, x_1, \varepsilon_0 = 0; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma^2}\right) \tag{34}$$

and therefore we have the joint conditional density function

$$f_{X_N,X_{t-1},\ldots,X_1|\varepsilon_0=0}(x_N, x_{t-1}, \ldots, x_1|\varepsilon_0 = 0; \theta) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_j^2}{2\sigma^2}\right). \tag{35}$$

The conditional log likelihood is

$$\ell(\theta) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \sum_{j=1}^{N} \frac{\varepsilon_j^2}{2\sigma^2}. \tag{36}$$

The non-linear nature of the log-likelihood function requires numerical optimization techniques for estimating maximum likelihood parameters, as no closed-form solution exists. Iterative methods, such as Newton-Raphson or gradient-based algorithms, are typically used to maximize $\ell(\theta)$. These algorithms are readily available in statistical software like R, Julia or Python, simplifying their practical implementation. However, we will not delve further into the estimation of moving average models, as this example is solely intended to illustrate the challenges associated with such estimations.

3 Applications of the Theory

In this chapter, we apply the previously developed theoretical framework to real-world time series data. The implementation will be carried out in Python. For this I used statsmodels, which is a Python library that offers a wide range of classes and functions for estimating statistical models and performing statistical tests. Furthermore I used numpy and matplotlib, which are well known libraries for numerical computations and plotting, respectively. Also I used yfinance which is a library that simplifies downloading and handling financial data from Yahoo Finance. It provides historical market data, including stock prices, indices, and currency exchange rates, making it a good tool for time series analysis and financial research.

To address the objectives of my thesis, I employed two different approaches. The first approach involved generating synthetic data, allowing me to better illustrate how the

theoretical concepts work in a controlled setting. The second approach used real-world data by downloading the Nikkei 225 time series. This was intended to demonstrate the practical challenges of applying the theory to real data, which often does not perform as well as expected.

Initially, I focused solely on the real-world data approach, but the results were less effective in demonstrating the theoretical concepts due to the noisy and complex nature of the data. As a result, I shifted to self-generated data, which provided clearer insights into the workings of the theory. This dual approach highlights the contrast between theoretical simplicity and practical complexity in time series analysis.

3.1 Analysis with Synthetic Data

3.1.1 Motivation

The use of synthetic data in this analysis was driven by the need for a controlled environment where the true parameters of the time series are known. This allows for precise comparisons between theoretical expectations and actual results, making it easier to validate the models and methods applied. By generating the data myself, I could design processes that align closely with the theoretical framework, ensuring that the outcomes clearly demonstrate the key concepts.

3.1.2 Generating Synthetic Data

To generate the data for an AR(2) process, I used a custom Python function with parameters $\alpha_1 = 0.6$, $\alpha_2 = 0.35$ and $\sigma^2 = 2$. Below is the Python function used for the generation:

```
def manual_ar_process(ar_coeffs, noise_var, n):
    noise = np.random.normal(0, np.sqrt(noise_var), n)
    p = len(ar_coeffs)
    ar_series = np.zeros(n)

    for t in range(p, n):
        ar_series[t] = sum(ar_coeffs[i] * ar_series[t - i - 1] for
            i in range(p)) + noise[t]

    return ar_series
```

The data generation process involves simulating values for an AR(2) process using the specified parameters and a predefined number of time steps. A plot of the time series is shown below:

Next, I split it into training and test sets. This allows me to fit the model on the training data and later evaluate its performance on the test set. This separation ensures that the models predictive capabilities can be assessed independently of the data it was trained on.

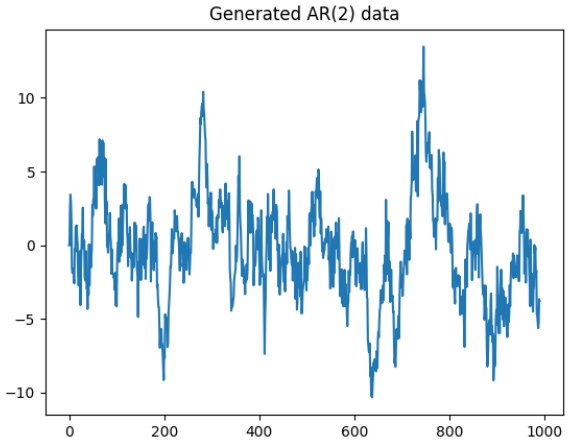

Figure 5: Visualization of generated AR(2) data.

3.1.3 Application of the Theory

Given the chosen parameters for the AR(2) process, the generated data is expected to be stationary. To verify the stationarity of the generated AR(2) process, I calculated the unit roots of the process. Stationarity in autoregressive models is determined by the location of these roots relative to the unit circle in the complex plane. Specifically, an AR process is stationary if all the roots of its characteristic equation lie outside the unit circle. Below is the Python function used to calculate the unit roots of the AR process:

```
def calculate_unit_roots(ar_coeffs):

    poly_coeffs = [1] + [-coeff for coeff in ar_coeffs]

    # Find the roots of the polynomial
    roots = np.roots(poly_coeffs)

    unit_roots = 1 / roots

    return roots, unit_roots

# Calculate roots and unit roots
roots, unit_roots = calculate_unit_roots(ar_coeffs)

# Check for stationarity: All roots should lie outside the unit
    circle
is_stationary = all(abs(root) > 1 for root in unit_roots)
if is_stationary:
    print("The process is stationary")
```

```
    'The process is stationary'
```

The unit roots of the process are 1.03807131 and −2.75235702. Since both roots lie outside the unit circle, the process is stationary. I also implemented a visualization function to plot the unit roots on the complex plane. This provides a clear visual representation of the stationarity of the AR process. The plot is shown below:

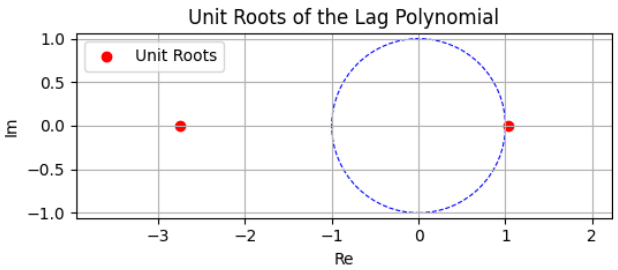

Figure 6: Unit roots of AR(2) process

Now we want to find the order of the process. Of cousre we know that it is 2 but we want to show how to estimate it. By analyzing the PACF plot, we can identify the lag beyond which the correlations are not significant, indicating the order of the AR process. For this I used the already implemented function `plot_pacf` from the statsmodels library. The plot is shown below 7.

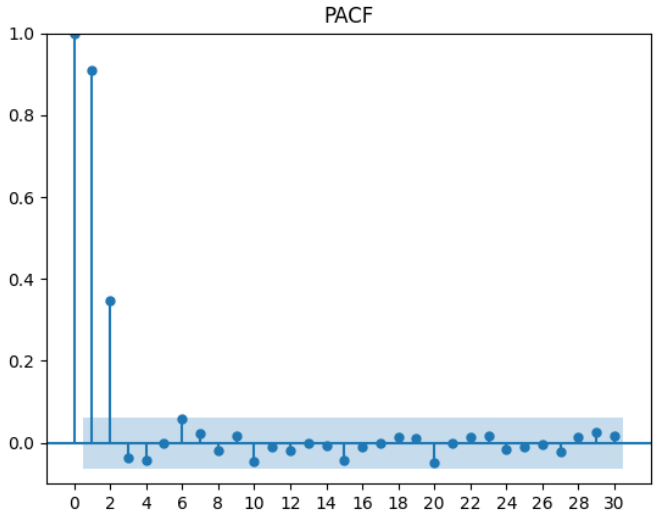

Figure 7: PACF of AR(2) process

I also plotted the ACF, where we can see that the values are decreasing. This is another indicator that the process is stationary. This can be seen in 8.

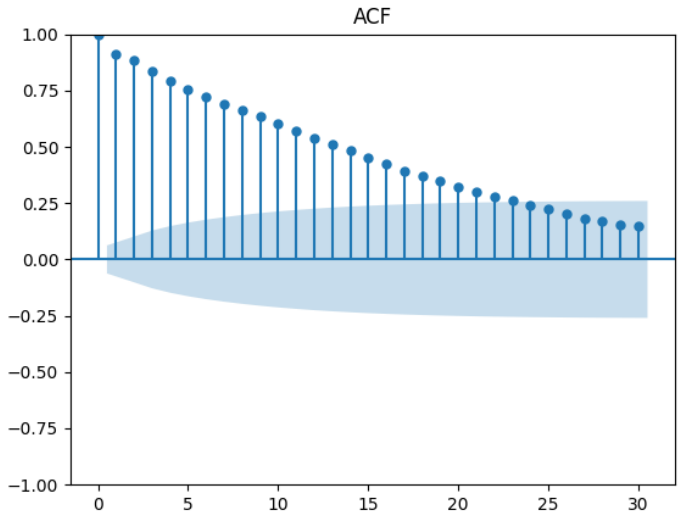

Figure 8: ACF of AR(2) process

In the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots, the blue shaded area around the x-axis represents the 95% confidence interval. This interval indicates the range within which we expect the true correlation to lie if the series is purely random (i.e., white noise). Any autocorrelation or partial autocorrelation value that lies outside this shaded area is considered statistically significant, implying that the correlation at that specific lag is unlikely to be due to random chance.

As we expected, in 7, we observe that the PACF values are significant up to lag 2. Therefore, we can conclude that the order of the AR process is 2, which aligns with the parameters used to generate the data. I also calculated the AIC values and found that the AIC is minimized at order 2, further confirming the correct order of the process. The AIC values for different orders are shown in 9.

In the next step we will estimate the parameters of the process. I will do this using the Yule-Walker equations and the implemented function `ARIMA.fit()` from the statsmodels library. I implemented the function to estimate the parameters of the AR process using the Yule-Walker equations.

```
1 def yule_walker(data, order):
2     # Compute sample autocovariances
3     autocovariances = [np.cov(data[:-k], data[k:])[0, 1] if k > 0
          else np.var(data, ddof=0)
4                        for k in range(order + 1)]
5
6     # Create the autocovariance matrix R
7     R = np.array([[autocovariances[abs(i - j)] for j in range(
          order)] for i in range(order)])
8
```

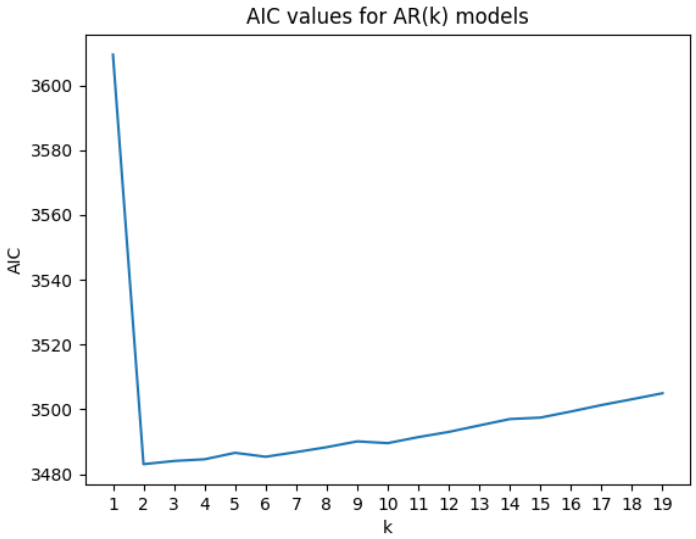

Figure 9: AIC values for different orders

```
    # Create the autocovariance vector r
    r = np.array(autocovariances[1:order + 1])

    ar_params = np.linalg.solve(R, r)

    # Compute noise variance
    noise_variance = autocovariances[0] - np.dot(ar_params, r)

    return ar_params, noise_variance
```

and the built in function from the statsmodels library:

```
model = ARIMA(train_data, order=(2, 0, 0)).fit()
model.summary()
```

Model	Parameter 1	Parameter 2	Sigma2
AR(2)	0.5917	0.3486	1.9548

Table 1: Summary from statsmodels ARIMA fit

Model	Parameter 1	Parameter 2	Sigma2
AR(2)	0.5918	0.3501	1.9241

Table 2: Summary from Yule Walker estimation

The estimated parameters from the Yule-Walker equations are $\alpha_1 = 0.5918$, $\alpha_2 = 0.3501$, and $\text{Var}[\sigma^2] = 1.9241$. These values are close to the true parameters used to generate the data, which were $\alpha_1 = 0.6$, $\alpha_2 = 0.35$, and $\text{Var}[\sigma^2] = 2$. This demonstrates the effectiveness of the Yule-Walker method in estimating the parameters of an AR process. The results are also consistent with the values obtained from the statsmodels library, as shown in Table 1.

Last but not least, I used the estimated parameters to forecast future values of the time series. I implemented a function to generate forecasts based on the AR(2) model. The plot 10 below shows the test data and the forecasted values using the estimated parameters.

I also used the built in function from the statsmodels library to forecast the values. The plot 11 below shows the test data and the forecasted values using the estimated parameters. Additionally I used the last two values of the training data as the initial values for the forecast.

3.1.4 Conclusion

The use of synthetic data has proven to be excellent for understanding and demonstrating the theoretical concepts of time series analysis. By generating data with known parameters and ensuring stationarity, I was able to directly observe how the theoretical models and methods perform under ideal conditions. This approach allowed for a clear validation of the theoretical framework and highlighted the strengths and limitations of the tools used in the analysis.

However, while synthetic data effectively demonstrates key properties such as stationarity and parameter estimation accuracy, it does not capture the complexities and unpredictability of real-world time series. To address these limitations, the next chapter will focus on applying the theoretical framework to a real-world time series, providing insights into the challenges and nuances of working with practical data.

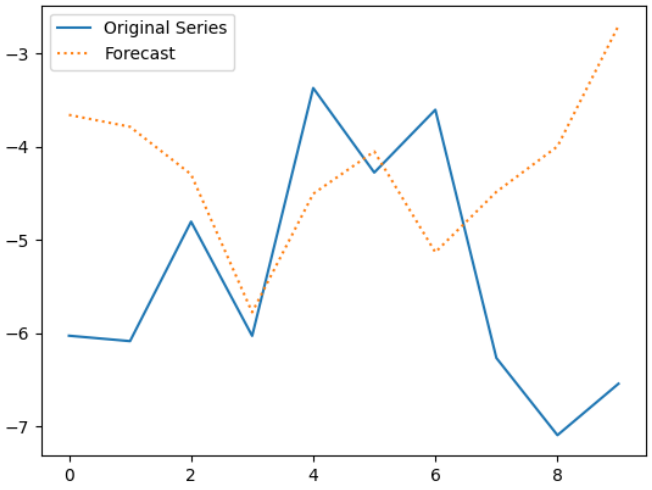

Figure 10: Forecasting with Yule-Walker estimation

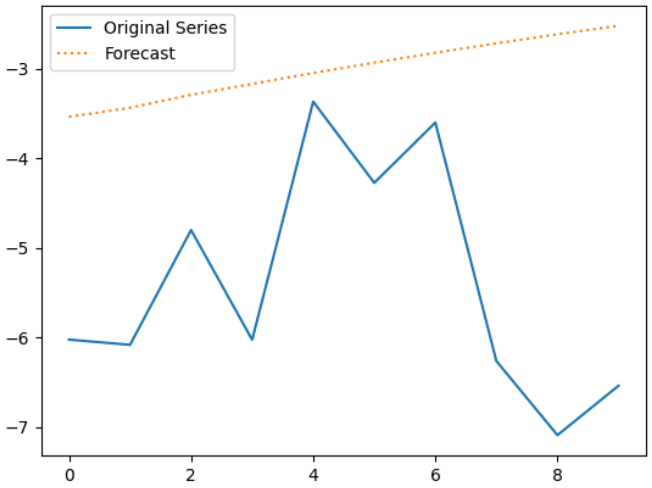

Figure 11: Forecasting with statsmodels ARIMA fit

3.2 Analysis of the Nikkei 225 Index

3.2.1 Data Description

The Nikkei 225 is a stock market index for the Tokyo Stock Exchange (TSE), which is one of the most important stock exchanges in the world. The index consists of the 225

largest companies listed on the TSE, representing a wide range of industries, including technology, automotive, and finance. The Nikkei 225 is a price-weighted index, meaning that the weight of each stock in the index is proportional to its price per share. The index is calculated by taking the sum of the prices of the 225 constituent stocks and dividing by a divisor to adjust for stock splits and other corporate actions.

For this analysis, I used the python package `yfinance` to download historical data for the Nikkei 225 index. The data includes the open, high, low, and close prices for each trading day and covers the period from December 30, 2019, to February 5, 2024. Table 3 shows a sample of the data.

Date	Open	High	Low	Close
2019-12-30	23770.929688	23782.490234	23656.619141	23656.619141
2020-01-06	23319.759766	23365.359375	23148.529297	23204.859375
2020-01-07	23320.119141	23577.439453	23299.919922	23575.720703
2020-01-08	23217.490234	23303.210938	22951.179688	23204.759766
2020-01-09	23530.289062	23767.089844	23506.150391	23739.869141

Table 3: Sample data from the Nikkei 225 index

The code snippet below shows how the data was downloaded and preprocessed for the analysis. The data was cleaned by removing missing values and unnecessary columns, such as the adjusted close price and volume. It was also limited to the first 1000 observations for the analysis.

```
import yfinance as yf

data: pd.DataFrame =  yf.download("^N225", period='5y')
data.dropna(inplace=True)
data.drop(columns=['Adj Close', 'Volume'], inplace=True)
data = data.iloc[:1000]
```

3.2.2 Application of the Theory

The application of the theoretical framework to the Nikkei 225 time series revealed several practical challenges that were absent in the controlled environment of synthetic data. First we only considered the closing prices of the Nikkei 225 index as we only covered single variable time series in the theoretical part. The data was then split into a training set and a test set. In 12 we can see the closing prices of the Nikkei 225 index over time.

Looking at 12, it is evident that the time series is most likely not stationary. In the previous chapter, we knew the parameters of the AR process and could calculate the unit roots to verify stationarity directly. However, with real-world data, this approach is not feasible. Instead, we will use the Dickey-Fuller test to determine whether the series

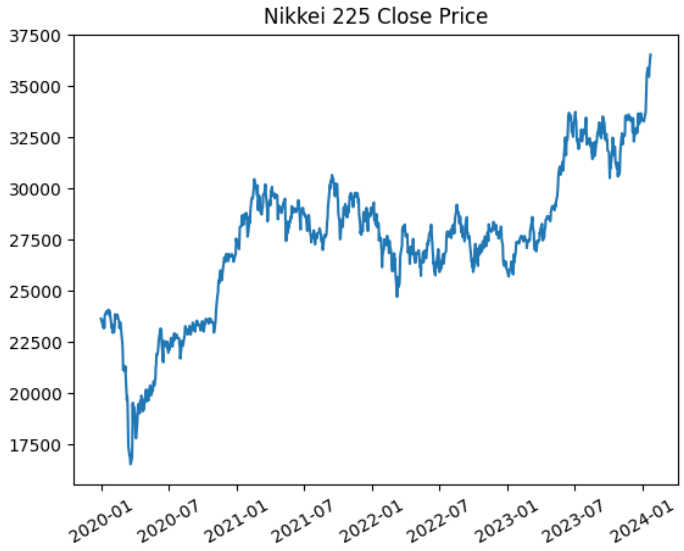

Figure 12: Closing prices of the Nikkei 225 index

	Values
Test Statistic	-0.690621
p-value	0.849164
Critical Value (1%)	-3.436999
Critical Value (5%)	-2.864476
Critical Value (10%)	-2.568333

Table 4: Results of the Dickey-Fuller test for the Nikkei 225 closing prices

is stationary. This following code snippet shows how the Dickey-Fuller test was applied to the closing prices of the Nikkei 225 index.

```
from statsmodels.tsa.stattools import adfuller
adf_test(train_data)
```

The results of the Dickey-Fuller test are shown in Table 4. The test statistic is -0.690621, and the p-value is 0.849164. The critical values at the 1%, 5%, and 10% levels are -3.436999, -2.864476, and -2.568333, respectively. Since the test statistic is greater than the critical values, we fail to reject the null hypothesis that the series is non-stationary. This result is consistent with the visual inspection of the time series.

A common technique discussed in Chapter 1.1 is to analyze the differenced time series, which can also be interpreted as the daily returns of the index. The differenced time series is shown in 13, where the transformation appears to address the initial non-stationarity. To confirm this, the Dickey-Fuller test was applied again, with the results summarized in 5.

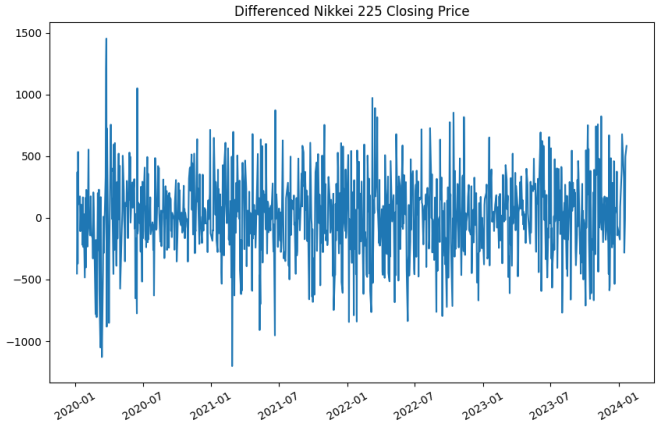

Figure 13: Differenced Nikkei 225 closing prices

	Values
Test Statistic	-18.034672
p-value	0.000000
Critical Value (1%)	-3.436999
Critical Value (5%)	-2.864476
Critical Value (10%)	-2.568333

Table 5: Results of the Dickey-Fuller test for the differenced Nikkei 225 closing prices

Since the test statistic is less than the critical values, we reject the null hypothesis that the series is non-stationary. This result confirms that the differenced time series is stationary.

Next, I plotted the ACF and PACF 14 for the differenced time series and encountered another problem. Unlike the synthetic data, these plots do not exhibit significant spikes or clear break-offs. This lack of clear structure may arise due to several factors:

- Noise: Real-world time series often contain random noise or external influences, which can obscure the underlying patterns visible in idealized synthetic data.
- Model Misfit: The data may not align well with the assumptions of AR or MA models, requiring either more complex models or additional preprocessing steps.
- Short-Term Dependencies: Real-world data may have weaker autocorrelations that are difficult to detect, particularly when the signal-to-noise ratio is low.

Given the challenges observed in the ACF and PACF plots, it is still possible to fit AR and MA models to the differenced time series. By experimenting with different orders, we can determine which models are most suitable for capturing the underlying

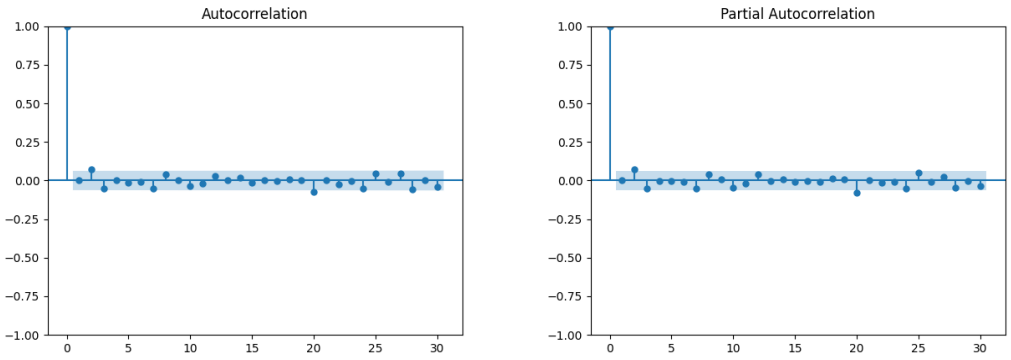

Figure 14: Visualization of ACF and PACF for the differenced process

structure of the data, despite the lack of clear patterns in the autocorrelation plots. To explore the suitability of AR and MA models, I fitted models for orders $p, q = 1, \ldots, 19$ and calculated their AIC values. The results are visualized in 15, providing a comparison of model performance across different orders. The optimal model order can be selected based on the AIC values, with lower values indicating better fit. In this case, the AIC values suggest that an ARMA(3, 3) model may be the most appropriate for the differenced Nikkei 225 time series.

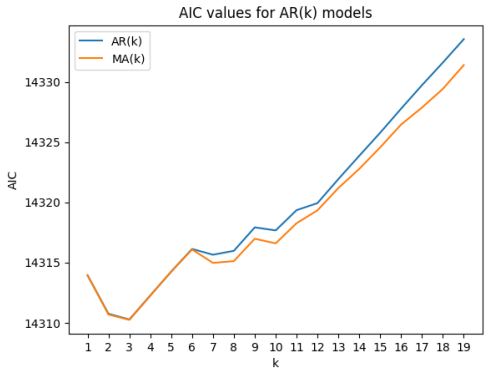

Figure 15: AIC values for AR and MA models of different orders

After fitting the ARMA(3, 3) model to the differenced time series, I used it to forecast the next 10 days of closing prices. The forecasted values, shown in 16, are plotted alongside the actual closing prices for comparison. While the forecast appears to align with the general trend of the differenced time series, a closer examination reveals significant limitations. Given the behavior of the differenced series, which fluctuates around zero, the forecast essentially reflects this mean-reverting nature without capturing any meaningful patterns or variations. In hindsight, even a simple heuristic or guess could have produced a similar forecast, highlighting the inadequacy of the model in providing

predictive insights for this data.

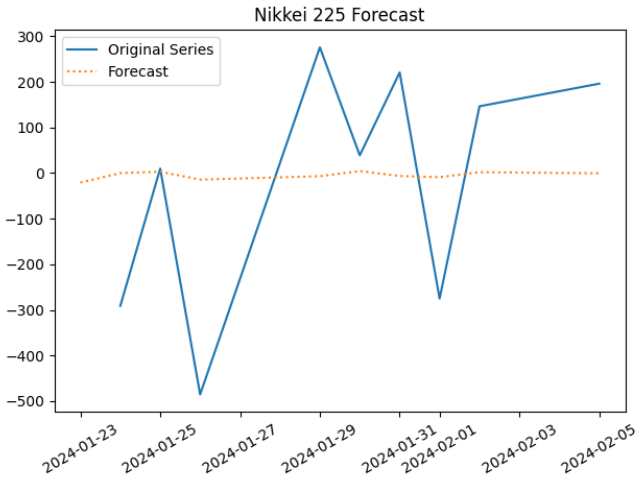

Figure 16: Forecasted and actual closing prices of the Nikkei 225 index

To further explore model estimation, I applied the Yule-Walker equations to estimate the AR parameters of the process. This method yielded a visually more pleasing result 17, as the fitted model appeared to capture some structure in the data. However, it is difficult to determine if this approach is genuinely superior, as Yule-Walker only estimates AR parameters and does not account for the MA components that may be present in the data. This highlights the complexity and uncertainty involved in modeling real-world time series.

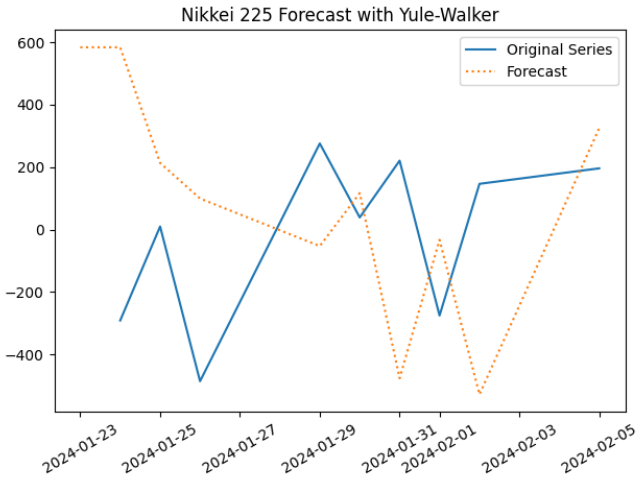

Figure 17: Forecasted and actual closing prices of the Nikkei 225 index using Yule-Walker

3.2.3 Conclusion

The analysis of the Nikkei 225 time series highlights the challenges of applying simple linear models, such as AR and MA, to real-world financial data. While these models are useful for understanding fundamental time series concepts, their performance in forecasting and capturing the complexities of financial data is limited. The noise, outliers, and lack of clear patterns in the data make it difficult for linear models to provide accurate and meaningful predictions.

This study demonstrates the need for more advanced techniques, such as ARCH and GARCH models for handling volatility, or machine learning approaches like neural networks, which can better capture the dynamics of financial time series. Incorporating these advanced methods into future analyses could significantly improve forecasting reliability and offer deeper insights into financial markets.

References

[1] Jürgen Franke, Wolfgang Härdle, Christian M. Hafner. *Statistics of Financial Markets*. Springer, 2008.

[2] Wolters, Jürgen; Hassler, Uwe. *Unit root testing*. Diskussionsbeiträge des Fachbereichs Wirtschaftswissenschaft der Freien Universität Berlin, No. 2005/23, http://hdl.handle.net/10419/28021

[3] James Douglas Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

[4] Bryne P. Rynne, Martin A. Youngson. *Linear Functional Analysis*. Springer, 2008.

[5] Dirk Werner. *Funktionalanalysis*. Springer, 2018.

[6] Gebhard Kirchgässner, Jürgen Wolters. *Introduction to modern time series analysis*. Springer, 2009.

[7] Christian Karpfinger, Hellmuth Stachel. *Lineare Algebra*. Springer, 2014.

[8] Hans-Otto Georgii. *Stochastik*. De Gruyter, 2009.

[9] Wayne A. Fuller. *Introduction to Statistical Time Series*. Wiley, 1996.