Pontifícia Universidade Católica
do Rio de Janeiro

Luiz Fernando Cunha Duarte

SARIMAX.jl: Open-Source Time Series Modeling in Julia through Advanced Optimization

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção.

Advisor: Prof. Davi Michel Valladão

Rio de Janeiro
August 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Luiz Fernando Cunha Duarte

SARIMAX.jl: Open-Source Time Series Modeling in Julia through Advanced Optimization

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção. Approved by the Examination Committee:

Prof. Davi Michel Valladão
Advisor
Departamento de Engenharia de Produção – PUC-Rio

Prof. Fernando Luiz Cyrino Oliveira
Departamento de Engenharia Industrial PUC-Rio

Prof. Alexandre Street de Aguiar
Departamento de Engenharia Industrial PUC-Rio

Dr. Joaquim Masset Lacombe Dias Garcia
PSR Power Systems Research

Rio de Janeiro, August 13th, 2024

Luiz Fernando Cunha Duarte

Graduated in Computer Engineering by the PUC-Rio. Researcher at Laboratory of Applied Mathematical Programming and Statistics (LAMPS). First author of the paper "Mixed Integer Optimization for SARIMA Specification and Estimation" presented in SBPO 2023.

Bibliographic data

Duarte, Luiz Fernando Cunha

SARIMAX.jl: Open-Source Time Series Modeling in Julia through Advanced Optimization / Luiz Fernando Cunha Duarte; advisor: Davi Michel Valladão. – 2024.

68 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia de Produção, 2024.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Séries Temporais. 3. Programação inteira mista. 4. Estimação. 5. Especificação automática. 6. ARIMA. I. Valladão, Davi Michel. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia de Produção. III. Título.

CDD: 658.5

To my parents and my aunts, for their support
and encouragement.

Acknowledgments

To my adviser Professor Davi Valladão for the stimulus and partnership to carry out this work.

To CNPq and PUC-Rio, for the aids granted, without which this work does not could have been accomplished.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Duarte, Luiz Fernando Cunha; Valladão, Davi Michel (Advisor). **SARIMAX.jl: Open-Source Time Series Modeling in Julia through Advanced Optimization**. Rio de Janeiro, 2024. 68p. Dissertação de Mestrado – Departamento de Engenharia de Produção, Pontifícia Universidade Católica do Rio de Janeiro.

This dissertation introduces SARIMAX.jl, a Julia package designed for time series estimation. The primary contribution of this work is the separation of model formulation from the estimation process, which allows for the selection of the most appropriate estimation method for each specific situation. SARIMAX.jl employs advanced optimization techniques to enhance stability, robustness, and accuracy in modeling SARIMA processes. The package also offers flexibility by allowing users to incorporate regularization and switch objective functions. Through a comparative study, SARIMAX.jl demonstrates superior performance across various in-sample metrics and competitive performance when compared to the *R forecast* package in the M4 competition monthly series, establishing it as a reliable open-source option for time series modeling. Additionally, this dissertation proposes a mixed-integer optimization approach for the specification and estimation of a specific subset of SARIMA models, known as seasonal autoregressive integrated (SARI) models. This approach guarantees global optimality in parameter estimation and the specification of the integration order and autoregressive part.

Keywords

Time Series; Mixed integer optimization; Estimation; Automatic specification; ARIMA.

Resumo

Duarte, Luiz Fernando Cunha; Valladão, Davi Michel. **SARIMAX.jl: Modelagem de séries temporais open-source em Julia usando otimização avançada**. Rio de Janeiro, 2024. 68p. Dissertação de Mestrado – Departamento de Engenharia de Produção, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação apresenta o SARIMAX.jl, um pacote em Julia projetado para estimação de séries temporais. A principal contribuição deste trabalho é a dissociação da formulação do modelo do processo de estimação, permitindo a seleção do método de estimação mais apropriado para cada situação específica. O SARIMAX.jl emprega técnicas avançadas de otimização para aprimorar a estabilidade, robustez e precisão na modelagem de processos SARIMA. O pacote também oferece flexibilidade ao permitir que os usuários incorporem regularização e alterem as funções objetivo. Por meio de um estudo comparativo, o SARIMAX.jl demonstra um desempenho superior em várias métricas de amostra e um desempenho competitivo em comparação com o pacote *R forecast* nas séries mensais da competição M4, estabelecendo-se como uma opção confiável e de código aberto para modelagem de séries temporais. Além disso, esta dissertação propõe uma abordagem de otimização inteira mista para a especificação e estimação de um subconjunto específico de modelos SARIMA, conhecidos como modelos autorregressivos integrados sazonais (SARI). Esta abordagem garante a optimalidade global na estimação de parâmetros e na especificação da ordem de integração e da parte autorregressiva.

Palavras-chave

Séries Temporais; Programação inteira mista; Estimação; Especificação automática; ARIMA.

Table of contents

List of figures

List of tables

List of algorithms

List of codes

List of Abreviations

AIC – Akaike Information Criterion

AICc – Akaike Information Criterion corrected

AR – Autoregressive

ARIMA – Autoregressive Integrated Moving Average

ARMA – Autoregressive Moving Average

BIC – Bayesian Information Criterion

CART – Classification and Regression Trees

CSS – Conditional Sum of Squares

MLE – Maximum Likelihood Estimation

MSE – Mean Squared Errors

MIO – Mixed Integer Optimization

MIP – Mixed Integer Programming

MA – Moving Average

OLS – Ordinary Least Squares

RNN – Recurrent Neural Networks

SARI – Seasonal Autoregressive Integrated

SARIMA – Seasonal Autoregressive Integrated Moving Average

SSE – Sum of Squared Errors

Memento homo, quia pulvis es et in pulverem reverteris

Genesis, *3:19.*

1 Introduction

The integration of time series modeling with new optimization techniques represents a significant advancement by bridging two areas that have historically been separated due to computational limitations. Previous methodologies, though effective, were constrained by rigid structures where model formulation was closely tied to the estimation process. This close association made it difficult to customize or vary the formulation to suit different needs.

By formulating the SARIMA model within an optimization framework, the model formulation is separated from the estimation process, which is then handled by optimization solvers. This separation provides greater flexibility, allowing for more customization in the model formulation.

This paradigm shift is evident in the formulation of the Autoregressive Integrated Moving Average (ARIMA) model as an optimization problem. Departing from traditional approaches resolves lingering issues and enhances flexibility and adaptability in time series modeling.

The ARIMA model, within the optimization framework, offers precise estimations and customization options. This approach removes previous barriers by dissociating the problem formulation from the estimation process and allows for domain-specific constraints, parameter regularization, variable selection, and integration of advanced features.

Additionally, this work introduces a novel method for optimal specification of a subset within the ARIMA family—Seasonal Autoregressive Integrated (SARI) models. This advancement moves away from conventional heuristics, aiming for precise, data-driven model specifications to improve forecasting accuracy.

Utilizing the JuMP framework in the Julia programming language is central to this effort. It empowers the estimation process while remaining flexible with solver selection, enhancing problem-solving efficiency without extensive reformulation.

The following chapters provide a detailed exploration of this fusion of time series modeling and optimization. The ARIMA model, cast as an optimization problem, serves as a catalyst for innovation in time series analysis.

2
Relevant Literature

Time series data, characterized by observations collected over regular intervals, permeate various domains, offering insights into the dynamics of phenomena ranging from economics and finance to climate and public health. In the realm of time series analysis, the introduction of the Autoregressive Integrated Moving Average (ARIMA) models in the 1970s marked a pivotal moment. Developed by George Box and Gwilym Jenkins as part of the Box-Jenkins methodology (BOX; JENKINS, 1970), ARIMA models, an extension of the Autoregressive Moving Average (ARMA) family, emerged as a powerful and versatile tool for the analysis and forecasting of time series.

The introduction of ARIMA models unfolded in an era strikingly distinct from today's computational landscape. At its inception, computational capabilities were in their nascent stages, and the field of optimization was only beginning to take shape. These circumstances presented formidable challenges in the modeling and analysis of time series data. However, the seminal work of George Box and Gwilym Jenkins laid the groundwork for further investigation and refinement, with the alluring prospect of leveraging modern computational tools for optimization. Over the ensuing decades, the evolution of computational resources has been nothing short of remarkable, marked by exponential growth. Simultaneously, the field of optimization has experienced substantial maturation.

The remarkable evolution of computational power bears profound implications for fields that have traditionally operated independently of optimization methodologies. As an illustration, Bertsimas (BERTSIMAS; DUNN; PASCHALIDIS, 2017) has calculated a striking speedup factor of 800 billion in the performance of Mixed Integer Optimization (MIP) solvers. This dramatic leap in computational efficiency has motivated the exploration of an optimization-based approach to Classification and Regression Trees (CART), bridging the gap between optimization and machine learning. It's worth noting that the estimation and specification of such tree models were initially introduced by Breiman (BREIMAN, 1984) as heuristics, a practical compromise necessary by the limited computational power available at the time.

These computational advancements are particularly advantageous for statistical modeling, especially within the domain of time series analysis. The fusion of contemporary computational capabilities with established statistical modeling techniques opens new horizons for data-driven decision-making. The convergence of these previously distinct domains not only promises enhanced comprehension of temporal data patterns but also unleashes the potential of sophisticated modeling approaches, all within the evolving framework of computational technologies.

2.1 Base Literature

Since its inception, the estimation of Autoregressive Integrated Moving Average (ARIMA) models has presented a significant challenge in time series analysis. This challenge arises primarily from the inherent non-linearity introduced by the Moving Average (MA) component. The MA component relies on past estimation errors multiplied by model-dependent coefficients, resulting in a non-linear relationship within the model.

The non-linear nature of the Moving Average (MA) component in Autoregressive Integrated Moving Average (ARIMA) models introduces complexity into the parameter estimation process. As a result, ARIMA model estimation remains technically intricate and mathematically demanding, often necessitating advanced statistical and computational methods to ensure the reliability and accuracy of the results.

During its early stages, ARIMA model estimation primarily relied on Maximum Likelihood Estimation (MLE)(FISHER, 1922), a rigorous statistical approach. MLE aims to identify model parameters that maximize the likelihood of observed data, thereby aligning with the structure of the model. This methodology involves formulating a likelihood function, followed by the application of numerical optimization techniques for parameter estimation. While MLE offers a principled approach, its computational intensity and the significant manual intervention required limited its practicality to some extent.

In this context, another method emerged known as Conditional Sum of Squares (CSS)(BOX; JENKINS, 1970), which gained prominence in the early stages of ARIMA modeling. CSS estimation provided an alternative technique that significantly simplified the estimation process compared to MLE. CSS involves sequentially considering each parameter, thereby reducing the computational complexity inherent in the estimation process.

Moreover, to further enhance the estimation process, alternative methods came to the forefront. Notably, the utilization of Yule-Walker equations became a pivotal element in ARIMA model estimation. These equations introduced a systematic framework for the estimation of Autoregressive (AR) components, relying on autocorrelation functions derived from the observed time series data. By solving these equations, preliminary estimates for the AR terms were attained.

Therefore, during its initial phases, the ARIMA estimation process seamlessly integrated the foundational principles of the Box-Jenkins methodology with the mathematical rigor inherent in Maximum Likelihood Estimation. This approach presented a structured, data-driven framework for the modeling and forecasting of time series data, particularly when seasonality represents a pivotal consideration. Despite the initial computational challenges associated with MLE, subsequent developments, including the strategic application of Yule-Walker equations, the Durbin-Levinson algorithm, and the adoption of CSS, contributed significantly to the precision and efficiency of the estimation process, particularly concerning the autoregressive components within ARIMA models.

An alternative approach, grounded in the framework of State Space Models (SSM) as articulated by Harvey (HARVEY, 1990), represents a compelling perspective in time series analysis. While conventional Autoregressive Integrated Moving Average (ARIMA) models traditionally involve a stepwise process of differencing to eliminate trend and seasonal components, aiming to render the series stationary, the SSM approach dissects time series into separate trend, seasonal, and irregular components.

A noteworthy revelation in this context is the recognition that ARIMA models can be interpreted as a specific subset of State Space models (DURBIN; KOOPMAN, 2012). The strength of this realization lies in the capacity of State Space models to amalgamate these components within a unified framework. As a result, State Space models enable the seamless integration of trend, seasonality, and irregularity into a single, holistic representation, eliminating the need for stepwise transformations.

What distinguishes the State Space framework is its incorporation of advanced estimation techniques, including the Kalman filter(KALMAN, 1960), which renders adaptable and responsive to real-time data. Moreover, State Space models facilitate retrospective analysis through smoothing algorithms(JONG, 1989), allowing for the extraction of historical insights, a feature notably absent in ARIMA models. This convergence of modeling and estimation within the State Space framework presents a powerful advantage for a broad spectrum of applications, from economics and finance(ZENG; WU, 2013) to engineering and signal processing(SMITH; BROWN, 2003).

The specification of ARIMA models has also posed a persistent challenge for time series analysts. Originating from the foundational work of Box and Jenkins, the intricacies of determining the autoregressive, seasonal autoregressive, moving average, and seasonal moving average orders, alongside both seasonal and non-seasonal integration orders, remain subjects of considerable disagreement within the methodological discourse. This complexity arises from the imperative reliance on statistical tests and analytical tools, including autocorrelation functions and partial autocorrelation functions, further contributing to the intricacy of the model specification process.

Various endeavors have sought to address the intricacies associated with ARIMA model specification. In 1982, Hannan and Rissanen(HANNAN; RISSANEN, 1982) presented a methodology for determining the order of an Autoregressive Moving Average (ARMA) model, with a caveat that its applicability was confined to stationary time series. Gomez (GÓMEZ, 1998) subsequently expanded the Hannan-Rissanen method to encompass multiplicative ARIMA models.

Liu (LIU, 1989) contributed to the field by proposing a method for identifying seasonal Autoregressive Integrated Moving Average (ARIMA) models, leveraging a filtering approach alongside heuristic rules. Additionally, proprietary algorithms employed by commercial software have been instrumental in ARIMA specification, albeit lacking detailed documentation in the public domain literature. Ord and Lowe (ORD; LOWE, 1996) conducted a comprehensive review of the implementation of automatic ARIMA specification in various commercial software packages, including AutoBox 3.0, AUTOCAST II, FORECAST PRO, NCSS, and 4CAST/2.

Notably, Forecast Pro (GOODRICH, 2000) has gained prominence for its adept automatic ARIMA algorithm, notably employed in the M3-forecasting competition (MAKRIDAKIS; HIBON, 2000), further emphasizing the significance of automated methodologies in enhancing the efficiency and accuracy of ARIMA model specification.

Preeminently employed in the field, the method put forth by Hyndman and Khandakar (HYNDMAN; KHANDAKAR, 2008) stands as a prominent approach for ARIMA model specification. This method relies on statistical tests, specifically the Canova-Hansen test (CANOVA; HANSEN, 1995) for seasonality and the KPSS test (KWIATKOWSKI et al., 1992) for non-seasonality, to ascertain the appropriate integration orders. The determination of other hyperparameters involves a systematic stepwise exploration across the model space, with the overarching objective of minimizing the Akaike Information Criterion (AIC). Crucially, this methodology rigorously excludes model specifications that contravene the fundamental properties of causality and invertibility, ensuring the coherence and validity of the resulting ARIMA models.

In spite of the existing challenges, the ARIMA model is widely used and is pervasive across open-source libraries in diverse programming languages. Notably, the R language (R Core Team, 2021) stands out for its influential implementation, distinguished by its original exposition of the Hyndman-Khandakar method for the automatic specification of ARIMA hyperparameters. Given R's preeminence in the statistical community, the *pmdarima* library (SMITH et al., 2017–) was conceived to deliver a robust adaptation of the R *auto.forecast* method for the Python programming language (ROSSUM; DRAKE, 2009).

In Python, the StatsModels library (SEABOLD; PERKTOLD, 2010) contributes yet another noteworthy implementation of ARIMA models, leveraging the State Space Models framework to enhance functionality and precision. Extending the scope, Julia (BEZANSON et al., 2017) emerges as an additional significant programming language in this context. Renowned for its powerful optimization packages, Julia supports advanced optimization techniques in the field of time series analysis. Of particular relevance is the StateSpaceModels.jl package (SAAVEDRA; BODIN; SOUTO, 2019), which utilizes Julia's optimization frameworks to implement state space models. Within this package, an implementation of the ARIMA model and the Hyndman-Khandakar method for automatic specification further enriches the landscape of time series modeling. This multilingual array of ARIMA implementations underscores the versatility and adaptability of the model across different programming environments, catering to the diverse needs of practitioners and researchers in the field.

2.2 Recent Literature and applications

One important factor to acknowledge, as mentioned by Robert Hyndman in his comments on the M3 competition, is that "there is virtually no improvement in forecasting accuracy using ARIMA models (labeled B-J automatic). This is interesting, but has been widely known since at least the time of the first M-competition" (HYNDMAN, 2001). Although it is reassuring that previous findings continue to hold, it is concerning that knowledge about these models has not significantly advanced in recent years. This statement, made in 2001, reflects a scenario that has seemingly remained static.

ARIMA models remain a powerful tool for time series modeling and forecasting and continue to be relevant across various fields. For instance, in epidemiology, ARIMA models have been used to predict the number of Dengue cases with great precision (MARTINEZ; SILVA; FABBRO, 2011). Another application in epidemiology can be found in (NOBRE et al., 2001), where ARIMA models are compared with Dynamic Linear Models for estimating the occurrence of two notifiable diseases.

ARIMA models are also extensively used in meteorology. Forecasting the mean temperature of a specific region is crucial for various activities, such as agriculture and energy consumption. For instance, a study by Chen (2018) utilizes ARIMA models to analyze the monthly mean temperature in Nanjing, China, from 1951 to 2017 (CHEN et al., 2018). Another application involves forecasting runoffs, which can inform actions to mitigate the effects of such events. A long-term study using ARIMA and SARIMA models analyzed this phenomenon in the United States (VALIPOUR, 2015). Additionally, SARIMA models are employed to forecast monthly, weekly, and daily monsoon rainfall time series (DABRAL; MURRY, 2017). These forecasts are critical for water resource management, irrigation scheduling, agricultural management, and reservoir operation. The article by Dabral et al. also compares SARIMA models with other techniques such as Artificial Neural Networks (ANN) and Fuzzy techniques, highlighting the advantages of SARIMA models in terms of interpretability and ease of modeling.

In the field of economics, ARIMA models are widely used due to their ability to forecast commodities, which is crucial for predicting the economic performance of many nations. For instance, a study by Divisekara et al. (2020) utilized SARIMA models to predict the price of lentils in Canada (DIVISEKARA; JAYASINGHE; KUMARI, 2020). Another application can be found in the forecasting of sugarcane production in India (KUMAR; ANAND, 2014). However, it is important to note that some economic time series, particularly those with high volatility, may not be suitable for ARIMA modeling (PETRICĂ; STANCU; TINDECHE, 2016). Despite this limitation, many aggregated economic series with relatively low sampling frequencies can be approximated by an ARIMA(0,1,1) process. This is demonstrated in a study that highlights the importance of ARIMA models in the economic field (ROSSANA; SEATER, 1995).

In the energy field, time series forecasts are crucial for the planning of energy production, the use of resources such as water in reservoirs, the expansion of transmission lines, and the installation of new power plants. For instance, a study by (VAGROPOULOS et al., 2016) compares four practical methods for electricity generation forecasting of grid-connected photovoltaic (PV) plants: SARIMA modeling, SARIMAX modeling (SARIMA with an exogenous factor), modified SARIMA modeling, and ANN-based modeling. These models were applied to the intraday forecast of a PV plant in Greece using real-world data. Another study (ATIQUE et al., 2019) applies ARIMA models to forecast total daily solar energy generation, which is vital for energy generation planning due to the intraday seasonality of solar generation. This seasonality can cause difficulties in meeting demand due to timing imbalances between peak demand and solar power generation. Additionally, (de Oliveira; Cyrino Oliveira, 2018) uses bagging ARIMA and other methods for mid- to long-term electric energy consumption forecasting, achieving substantial improvements in the forecast accuracy of energy demand for end-use services in both developed and developing countries.

Finally, one of the primary current uses of ARIMA models is as a benchmark for new models, particularly machine learning models, due to their high performance in time series modeling and forecasting. For instance, (SIAMI-NAMINI; NAMIN, 2018) uses ARIMA models as a benchmark for LSTM (Long Short-Term Memory) neural networks in economic and financial time series. Another study by (HEWAMALAGE; BERGMEIR; BANDARA, 2021) compares ARIMA with a new type of neural network, Recurrent Neural Networks (RNN), which have become competitive forecasting methods, as notably shown in the winning method of the recent M4 competition (MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2020). Furthermore, (NING; KAZEMI; TAHMASEBI, 2022) compared ARIMA, LSTM, and Prophet (TAYLOR; LETHAM, 2018) models for oil production forecasting and found that both ARIMA and LSTM performed better than Prophet, with ARIMA demonstrating robustness in predicting the oil rate of wells across unconventional reservoirs.

Hybrid approaches that leverage the strengths of both ARIMA models and machine learning techniques have also been explored. For example, (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019) proposes an intelligent hybridization of ARIMA models, Multi-Layer Perceptron (MLP), and Support Vector Regression (SVR). This new methodology was applied to six well-known real-world complex time series, achieving better performance than single and other hybrid models. Similarly, (PHAN; NGUYEN, 2020) proposed a hybrid approach that combines ARIMA models with statistical machine learning methods, applying the model to forecast the water level of the Red River.

In summary, ARIMA and SARIMA models continue to be invaluable tools across various fields for time series modeling and forecasting. Despite the observation by Robert Hyndman that there has been little improvement in forecasting accuracy with ARIMA models since the first M-competition, their relevance and application remain significant.

3
Theoretical Background

3.1
Autoregressive models

The autoregressive (AR) term in time series modeling aims to express the current value of the series, denoted as y_t, as a linear combination of its previous k observations. This means that to express the current value, the model uses a weighted sum of the past k values of the series.

In simpler terms, an AR model expresses the current value based on a combination of its past values, where the number of past values used is k. This approach helps to capture the pattern and dependencies in the time series data over time.

Mathematically, this can be represented as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_k y_{t-k} + \epsilon_t \tag{3-1}$$

Here's the revised paragraph with the coefficients ϕ_i mentioned first, followed by the definition of $\mathbf{\Phi}$:

In this context, ϕ_i are the coefficients associated with the autoregressive terms, and the vector of these coefficients will be represented as $\mathbf{\Phi}$. Additionally, c represents the intercept, and ϵ_t denotes the error term. In an autoregressive (AR) model, the objective is to determine the values of these coefficients that minimize the in-sample error. This is specifically achieved by minimizing the sum of squared errors (SSE), represented as:

$$f(\mathbf{\Phi}) = \epsilon_t^2 = (y_t - c - \sum_{i=1}^{p} \phi_i y_{t-i})^2 \tag{3-2}$$

The function $f(\mathbf{\Phi})$ is quadratic and convex, allowing for the estimation of the optimal coefficients through minimization. This process, known as ordinary least squares (OLS) estimation, identifies the coefficients that minimize the SSE, thereby providing the best fit for the model.

One important property that sometimes is enforced in AR models, is that the characteristic polynomial, formed by the model's coefficients, needs to be stationary. The polynomial is described as

$$\mathcal{P}(p) = 1 - \sum_{i=1}^{p} \phi_i z^i \tag{3-3}$$

If all the inverse roots of the polynomial lie inside the unit circle, the model is considered stationary. This property is crucial for ensuring that the model does not diverge. A stationary model can be represented by its past errors, depending on the coefficients.

3.2
Maximum Likelihood estimation

Another prevalent formulation of Autoregressive (AR) models involves maximizing the likelihood function. It can be demonstrated that maximizing the likelihood of a model is equivalent to minimizing squared errors. Consider a fundamental linear regression model. The likelihood function for an AR model with normally distributed errors is given by:

$$L(c, \mathbf{\Phi}, \sigma^2|\mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - c - \sum_{j=1}^{p} \phi_j y_{i-j})^2\right)$$

Here, σ^2 represents the variance of the errors, and it is assumed that the mean of the errors is 0. Under these assumptions, the log-likelihood function can be expressed as:

$$\ell(c, \mathbf{\Phi}, \sigma^2|\mathbf{y}) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - c - \sum_{j=1}^{p} \phi_j y_{i-j})^2$$

Maximizing the likelihood is equivalent to maximizing the log-likelihood. To simplify the optimization problem, it is customary to minimize the negative log-likelihood. Consequently, the problem transforms into:

$$\underset{c, \mathbf{\Phi}, \sigma^2}{\text{minimize}} \quad -\ell(\beta, \sigma^2|\mathbf{y})$$

Now, focus on the term $\sum_{i=1}^{n} (y_i - c - \sum_{j=1}^{p} \phi_j y_{i-j})^2$, since the rest can be considered a constant and therefore disregarded from the optimization. This term is proportional to the sum of squared errors (SSE) in linear regression. Therefore, minimizing the negative log-likelihood is equivalent to minimizing the sum of squared errors. This relationship holds for models with normally distributed errors, extending to AR models.

3.3 Moving average models

The Moving Average (MA) models instead of expressing the current term y'_t as a linear combination of the k past terms, it endeavors to elucidate the current term as a linear combination of the k preceding forecast errors ϵ_{t-k}:

$$y'_t = c + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{3-4}$$

Here, $\mathbf{\Theta}$ represents a vector of coefficients θ_i associated with the past errors, and ϵ_t denotes the error term. Unlike Autoregressive (AR) models that employ past observations of the time series, MA models utilize previous forecast errors, necessitating simultaneous estimation of both coefficients and errors.

This makes the use of OLS estimation infeasible, leading to the adoption of nonlinear least squares. However, this entails a trade-off, with the loss of some desirable properties, such as global optimality. So, the estimated coefficients do not have the guarantee that they are the best ones possible, at least using the standard estimation techniques discussed in 2.1.

An important property of MA models is called invertibility. A $MA(q)$ model is called invertible if it can be written as an $AR(\infty)$. This property is desirable once the error at a certain time can be expressed by a linear function of the model's present and past observations. This enables the computation of confidence intervals for ARIMA prediction.

3.4 Autoregressive Integrated Moving Average models

Autoregressive Integrated Moving Average (ARIMA) models combine autoregressive and moving average components with an additional integration component to model and forecast time series data.The integration term refers to the number of differencing steps needed to make the series stationary, addressing the model's requirement for stationarity. Thus, the integration term represents the differencing necessary to achieve a stationary series.

ARIMA models are typically denoted as ARIMA(p,d,q), where p represents the degree of the autoregressive component, d the degree of differencing (integration), and q the degree of the moving average component. Since the original series is differenced, it is represented as y'_t in the model's notation.

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{3-5}$$

The inclusion of the Moving Average (MA) component in ARIMA models introduces significant estimation challenges, as standard methods do not ensure optimality. Two principal methods for estimating these models are detailed in Section 2.1. The most prominent approach employs the State Space Models framework for estimation.

3.5 Seasonal Autoregressive Integrated Moving Average models

The Seasonal Autoregressive Integrated Moving Average (SARIMA) models extend ARIMA models to accommodate time series with seasonal patterns. ARIMA models, exemplified by ARIMA(p,d,q), are not sparse; the number of parameters to be estimated, excluding the intercept, equals $p + q$. Consequently, higher AR and MA orders increase the number of parameters, making the estimation process more prone to numerical instability.

In many monthly time series, data from the same month in different years exhibit relationships, suggesting the need for an AR model of order 12 to capture this seasonal pattern. To avoid estimating all 12 coefficients, SARIMA models incorporate seasonal autoregressive and moving average components. The mathematical formulation of SARIMA models is expressed as:

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{P} \Phi_i y'_{t-si} + \sum_{i=1}^{Q} \Theta_i \epsilon_{t-si} + \epsilon_t \tag{3-6}$$

Here, Φ_i are the coefficients of the seasonal AR process, Θ_i are the coefficients of the seasonal moving average process, and s is the seasonal period. The estimation process for SARIMA models remains consistent with that of ARIMA models.

3.6 Hyndman-Khandakar algorithm for ARIMA specification

The Hyndman-Khandakar algorithm, proposed by Hyndman and Khandakar, uses the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test to determine the differencing parameter (d) in ARIMA model specification. The algorithm begins by testing four candidate models: ARIMA(0, d, 0), ARIMA(2, d, 2), ARIMA(0, d, 1), and ARIMA(1, d, 0). For these models, a constant term is considered unless $d \geq 2$. If $d < 2$, an additional model, ARIMA(0, d, 0) without a constant term, is also tested. The best model is selected based on the corrected Akaike Information Criterion (AICc).

Once a model is chosen, it is tested to ensure it is both stationary and invertible. If the model fails these tests, it is discarded, and the next best model, according to the AICc criterion, is tested. This process continues until a model that meets both the stationarity and invertibility conditions is found. The algorithm then explores different variations by incrementing or decrementing p and/or q by 1 and by including or excluding the constant term.

The process can be broken down into the following steps:

- The algorithm starts by testing four candidate models:
  - ARIMA(0, d, 0)
  - ARIMA(2, d, 2)
  - ARIMA(0, d, 1)
  - ARIMA(1, d, 0)
- For all these models, a constant term is considered unless $d = 2$.
- If $d < 2$, an additional model, ARIMA(0, d, 0) without a constant term, is also tested.
- The best model is chosen based on the Akaike Information Criterion corrected (AICc).
- The chosen model is then tested to ensure that it is both stationary and invertible. If the model fails these tests, it is discarded, and the next best model, according to the AICc criterion, is tested. This process continues until a model that meets both the stationarity and invertibility conditions is found.
- The algorithm then explores different variations of the chosen model by:
  - Incrementing or decrementing p by 1.
  - Incrementing or decrementing q by 1.
  - Including or excluding the constant term.

This thorough examination helps find the optimal ARIMA model configuration.

Figure 3.1, extracted from the "Forecasting: Principles and Practice" book (HYNDMAN; ATHANASOPOULOS, 2021), illustrates iterations of the Hyndman-Khandakar algorithm. The orders of the AR and MA components at each step are depicted, with the best-performing model of each iteration circled in black.

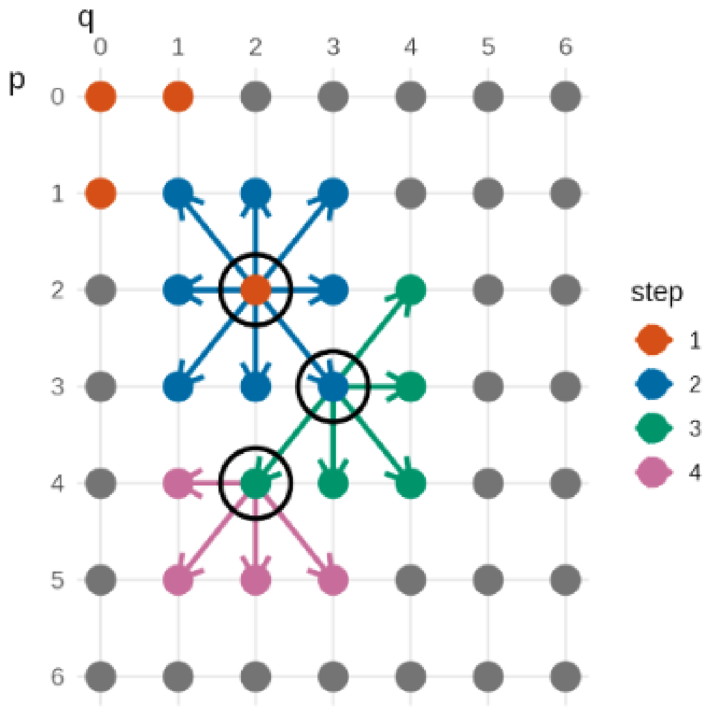

Figure 3.1: An example of Hyndman-Khandakar algorithm iterations.

While the example above is illustrative, it underscores a major challenge faced by the Hyndman-Khandakar algorithm—the high dimensionality of the hyperparameters space. Figure 3.1 displays only the dimensions of AR and MA, excluding differentiation and model constant for interpretability. The figure emphasizes the algorithm's exploration in a vast hyperparameters space, which makes the specification process a large combinatorial problem.

The same logic can be applied to SARIMA models. When dealing with SARIMA models, the process begins by determining the non-seasonal integration order (d). Once the series has been differenced accordingly, the Canova-Hansen test is applied to ascertain the seasonal integration order. Subsequently, the procedure mirrors that of non-seasonal ARIMA models but includes initial models with corresponding autoregressive orders for the seasonal component. Additionally, the seasonal parameters (P and Q) are varied independently of the non-seasonal parameters.

4
Proposed methodology

This chapter aims to elucidate the methodological contributions presented in this dissertation, comprising two distinct sections: Estimation and Specification. The primary contribution of this work is the creation of an abstraction layer in the estimation and specification process of ARIMA models. This abstraction dissociates the model formulation from the methods used for its estimation and specification, allowing flexibility in model formulation without increasing the complexity of the estimation and specification process.

4.1
Estimation under an optimization lens

This section systematically explores the complexities of modeling Autoregressive (AR) and Moving Average (MA) components, culminating in the presentation of a new way of estimating ARIMA models. An essential consideration is that, when considering ARIMA models, the differentiation of the original time series is conducted prior to the estimation process using traditional statistical tests to determine the integration orders. Hence, the model assumes that the inputted time series are stationary and will be referred to as y'.

4.1.1
Autoregressive (AR) Modeling

Considering the formulation presented in Equation 3-1 in Section 3.1, the estimation process of autoregressive (AR) models can be expressed using the optimization framework as follows:

$$\underset{c,\mathbf{\Phi},\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-1}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \epsilon_t, \quad \forall t \in 1, \cdots, T \tag{4-2}$$

This formulation explicitly outlines the objective of minimizing squared errors within the dynamic constraints of the AR model, as expressed in Equation 4-2. In this formulation, the c, $\mathbf{\Phi}$ and ϵ are variables that the optimization model aim to find the optimal values.

Notably, for an AR(p) model, where p denotes the number of preceding observations utilized, the initial p values of the time series are treated as the initial values of the model.

4.1.2 Moving Average (MA) Modeling

Considering the formulation presented in Equation 3-4 in Section 3.3, the estimation process of autoregressive (MA) models can be expressed using the optimization framework as follows:

$$\underset{c,\Theta,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-3}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \cdots, T \tag{4-4}$$

In this formulation, the c, $\mathbf{\Theta}$ and ϵ are variables that the optimization model aim to find the optimal values. While the objective function remains consistent, the constraint introduces a product of variables (θ_i and ϵ_i), inducing nonlinearity. Despite this, a wide range of optimization methods can provide estimations for these parameters with desired properties. Some optimization techniques present in Alpine optimizer (KIM; RICHARD; TAWARMALANI,) can deal with this non-linearity and guarantee global optimality.

An alternative modeling approach involves the following function:

$$f(\mathbf{\Theta}) = \underset{c,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-5}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \cdots, T \tag{4-6}$$

The f function accepts a vector of θ_i coefficients, $\mathbf{\Theta}$, and returns the Sum of Squared Errors (SSE) of the MA model using these coefficients. In f, the nonlinearity is absent, ensuring it yields the minimum SSE for the given coefficients. Utilizing a black-box optimization method enables the optimization of f to obtain coefficients that minimize the SSE of the function without the guarantee of global optimality.

4.1.3 ARIMA Modeling

Considering the previous formulations of both AR and MA models, this formulation of the ARIMA model aims to reproduce the dynamic of the model in the context of an optimization problem.

$$\underset{c,\Phi,\Theta,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-7}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \cdots, T \tag{4-8}$$

This formulation presupposes stationarity in the series, denoted by y', implying that both seasonal and non-seasonal differentiations have been previously conducted. That means that the integration order is determined before the estimation process.

The dynamic representation of the model is articulated in Equation 4-8. Within this equation, the term $\sum_{i=1}^{p} \phi_i y'_{t-i}$ signifies the autoregressive component, while the term $\sum_{i=1}^{q} \theta_i \epsilon_{t-i}$ denotes the moving average component.

It is crucial to underscore that in this context, the term ϵ_t represents the error at time t, embodying the difference between the observed value and the model's estimation.

It is imperative to notice that the vector ϵ is regarded as a variable within the optimization model. This innovation constitutes a distinctive feature of the proposed formulation, deviating from conventional ARIMA estimation methods that handle past errors recursively. In this approach, the optimization model concurrently estimates the errors alongside other model parameters. The overarching objective, defined by the minimization of the sum of squared errors, motivates the model to seek parameter values that not only capture the dynamics of the time series but also minimize the residuals, specifically ϵ_t.

This formulation introduces non-linearity into the dynamic constraint outlined in Equation 4-8. This non-linearity arises from the product between the θ coefficients and the ϵ variables. Consequently, certain desirable optimization properties, such as a global minimum, can be forfeited. However, it is crucial to note that this non-linear characteristic aligns with prevailing practices, as estimations from other packages also lack these particular properties. In this context, the non-linearity remains a valid and accepted feature of the formulation, as it does not deviate from established norms within the field of ARIMA modeling.

4.1.4
Alternative Approach to Address Non-Linearity

In order to address the non-linearity inherent in the preceding ARIMA formulation, the idea used in 4.1.2. To achieve this, the introduction of the function f is imperative, defined as follows:

$$f(\mathbf{\Theta}) = \underset{\phi_t,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-9}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in \{1, \cdots, T\} \tag{4-10}$$

It is evident that the non-linearity no longer prevails, as the $\mathbf{\Theta}$ coefficients are now parameters of the function and not variables within the optimization problem. Furthermore, with the specification of the coefficient vector,$\mathbf{\Theta}$, the estimation of the remaining parameters achieves optimality, transforming the problem into a linear form.

To optimize the moving average coefficients, a non-linear black box algorithm is employed with the objective of minimizing the output from $f(\mathbf{\Theta})$, specifically in this case, the sum of squared errors (SSE).

$$\underset{\mathbf{\Theta}}{\text{minimize}} \quad f(\mathbf{\Theta}) \tag{4-11}$$

This approach facilitates the meticulous adjustment of parameters, ensuring an optimal fit to the time series data given the moving average coefficients.

4.1.5
Adding Exogenous Variables

The incorporation of exogenous variables into the ARIMA model is a straightforward process using the optimization framework. Let $\mathbf{X}_{t\times m}$ denote the matrix containing the exogenous variables as its m columns. It is imperative to note that all explanatory variables must be stationary to adhere to the assumptions of the ARIMA model. Let β represent the vector of coefficients of the explanatory variables. Thus, the model can be formulated as follows:

$$\underset{c,\Phi,\Theta,\beta,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-12}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{m} \beta_i X'_{ti} + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in \{1, \cdots, T\} \tag{4-13}$$

The inclusion of the exogenous variables, represented by $\mathbf{X}$ and β, enhances the model's explanatory power by accounting for additional external factors.

4.1.6 SARIMA Modeling

Extending ARIMA models to handle time series with seasonal patterns, known as seasonal ARIMA (SARIMA), can also be tackled using optimization techniques. This setup involves adding a seasonal autoregressive component, where autoregressive terms are multiples of the seasonal length, along with a seasonal moving average component that follows the same pattern.

$$\underset{c,\phi_i,\theta_i,\Phi_i,\Theta_i,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-14}$$

$$\begin{aligned} \text{subject to} \quad y'_t = {} & c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \\ & + \sum_{i=1}^{P} \Phi_i y'_{t-si} + \sum_{i=1}^{Q} \Theta_i \epsilon'_{t-si} + \epsilon_t, \quad \forall t \in \{1, \cdots, T\} \end{aligned} \tag{4-15}$$

It is important to note that although the methodology creates distinct components to address the seasonal pattern, the tests to ensure the estimated coefficients meet stationarity and invertibility requirements should consider the AR and MA coefficients as a whole. This includes setting the lags not modeled between the non-seasonal and seasonal components to zero.

4.2 Specification under an optimization lens

As elucidated in Section 2.1, the process of specifying ARIMA models has garnered significant attention from researchers over the years. A primary challenge arises from its reliance on the outcomes and interpretation of statistical tests, introducing variability in model specifications. Consequently, there exists a critical gap in the literature — a lack of a methodology capable of deriving an optimal specification.

Another impediment in the specification phase is the inherent lack of sparsity in models. This issue primarily stems from constraints imposed by conventional ARIMA model estimation techniques, limitations that are circumvented by the optimization-based formulation presented in Section 4.1. For example, when specifying an AR(5) model, it implies the obligatory estimation of all the first five autoregressive coefficients. This condition dictates that higher AR orders lead to less sparse models. Recognizing this limitation, SARIMA models segregate seasonal and non-seasonal components, offering a marginally more sparse model compared to modeling with a high-order autoregressive component. Consequently, existing specification methods encounter challenges in modeling sparse ARIMA structures.

This section will underscore that the prevalent algorithm for ARIMA model specification, described in 3.6, relies on local search within the hyperparameter space where each greedy choice in an iteration lacks proven optimality.

Moreover, the methodology confines its search to hyperparameters that do not allow for the selection of sparse autoregressive and moving average components. This limitation results in a reduced hyperparameters space, overlooking the potential for sparse model configurations.

4.2.1 Another way of formulating ARIMA models

The content presented in this subsection and the subsequent subsection has been excerpted from a research paper developed as part of the master's program (DUARTE et al., 2023). The paper comprehensively discusses the proposed methodology and provides a comparative analysis against the Hyndman-Khandakar algorithm.

Consider the autoregressive formulation of type, where Δy represents the first different of series y,

$$\boldsymbol{\Delta} y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p-1} \phi_i \boldsymbol{\Delta} y_{t-i} + \epsilon_t, \tag{4-16}$$

as in the Augmented Dickey-Fuller (ADF) test (DICKEY; FULLER, 1979). Note that this is a regression model encompassing various components, including an intercept term α, a trend component βt, a differencing component γy_{t-1}, an autoregressive component $\sum_{i=1}^{p-1} \phi_i \boldsymbol{\Delta} y_{t-i}$, and an error term ϵ_t. Notably, the differencing component plays a crucial role in determining whether the original series or the differenced series is being modeled. This behavior is governed by the coefficient γ.

$$y_t - y_{t-1} = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p-1} \phi_i (y_{t-i} - y_{t-i-1}) + \epsilon_t \tag{4-17}$$

$$y_t = \alpha + \beta t + (1+\gamma) y_{t-1} + \sum_{i=1}^{p-1} \phi_i y_{t-i} - \sum_{i=1}^{p-1} \phi_i y_{t-i-1} + \epsilon_t \tag{4-18}$$

$$y_t = \alpha + \beta t + (1+\gamma+\phi_1) y_{t-1} + \sum_{j=2}^{p-1} (\phi_j - \phi_{j-1}) y_{t-j} - \phi_{p-1} y_{t-p} + \epsilon_t \tag{4-19}$$

When γ is zero, it indicates that the differenced series is being modelled. If γ is different from zero, it is possible to show that (4-16) can be reformulated into a traditional autoregressive model

$$y_t = \alpha + \beta t + \sum_{j=1}^{p-1} \theta_j y_{t-j} + \varepsilon_t.$$

Hence, by utilizing this formulation, it becomes feasible to estimate a regression model that incorporates differencing of the series and incorporates an autoregressive component akin to the ARIMA approach. It is worth noting that the MA component was omitted because it introduces a non-linearity in the model and the primary focus is to achieve optimality.

4.2.2 Optimal SARI specification

The formulation presented below is derived from the Augmented Dickey-Fuller (ADF) test, with a limitation of allowing only one automatic differencing order.

$$\underset{\alpha,\beta,\epsilon_t,\gamma,\phi}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{4-20}$$

$$\text{subject to} \quad \Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \phi_i \cdot \Delta y_{t-i} + \epsilon_t, \quad \forall t \in T \tag{4-21}$$

$$||\Psi||_0 \leq K, \quad \text{where } \mathbf{\Psi} = \{\alpha, \beta, \gamma, \Phi\} \tag{4-22}$$

The proposed formulation aims to minimize the sum of squared residuals while adhering to the dynamic nature of the model described in equation 4-21. An additional constraint is introduced to promote desirable sparsity in the model by imposing a limit on the 0-norm of the parameter vector Ψ, controlled by the hyperparameter K. For a more theoretical explanation of introducing L_0 regularization, refer to section 5.2.

Given the objective of estimating a fixed number of non-zero coefficients (denoted as K) from a larger set of possibilities, the formulation proposed above can be written using an integer optimization approach. This approach allows the control of the number of non-zero coefficients through the hyper-parameter named K. Consequently, the optimization model that focuses on capturing and estimating the K most relevant coefficients can be described as

$$
\begin{aligned}
\underset{\alpha,\beta,\gamma,\phi,I^{\phi},I^{\gamma}}{\text{minimize}} \quad & \sum_{t=1}^{T} \epsilon_t^2 && \text{(4-23)}\\
\text{subject to} \quad & \Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \phi_i \cdot \Delta y_{t-i} + \epsilon_t, \quad \forall t \in T && \text{(4-24)}\\
& -MI^{\phi_i} \leq \phi_i \leq MI^{\phi_i}, \quad \forall i \in \{1, \ldots, p\} && \text{(4-25)}\\
& -MI^{\gamma} \leq \gamma \leq MI^{\gamma} && \text{(4-26)}\\
& -MI^{\alpha} \leq \alpha \leq MI^{\alpha} && \text{(4-27)}\\
& -MI^{\beta} \leq \beta \leq MI^{\beta} && \text{(4-28)}\\
& I^{\alpha} + I^{\beta} + I^{\gamma} + \sum_{i=1}^{p} I_i^{\phi} \leq K, && \text{(4-29)}
\end{aligned}
$$

where $\boldsymbol{I}^{\phi} = \{I^{\phi_t} \mid t = i \in \{1, \ldots, p\}\}$ and $\boldsymbol{M}$ is a large number.

The vector I^{ϕ} and the variables $I^{\gamma}, I^{\alpha}, I^{\beta}$ in (4-25)–(4-29) are binary variables. I^{ϕ} is a binary variable vector that takes the value 0 when a coefficient is not considered (assumes zero value) and takes the value 1 when it is considered for non-zero estimation. Since this constraint(4-25) allows the model to disconsider some lags, the interpretation of the parameter p has changed from the number of previous lags used, to the range in which the lags can be selected. Thus, it is easy to notice that the family of autoregressive models considered by this formulation is significantly bigger than the one used by the traditional methods, which is a subset of it.

The values of the binary variables $I^{\gamma}, I^{\alpha}, I^{\beta}$ are also controlled by the model constraint in (4-29).

It is worth noting that when γ is equal to 0, the model is representing the first-order differenced series (Δy). As such, this methodology does not rely on statistical tests to determine whether differencing is necessary for the time series. Instead, the model estimates it as a parameter.

Given that the model selects only K parameters, an extension of this approach allows for the consideration of seasonality by using a value of p greater than the seasonal period. Let p' be the number of autoregressive lags considered in this approach, and let's assume a yearly seasonality. It can be observed that if $p' \geq 12$, the model could select lags 1 and 12, which is equivalent to choosing

$p = 1$ and $P = 1$ in the standard SARIMA approach. However, this extension does not include seasonal differentiation. This flexibility demonstrates that the traditional approach of modeling the autoregressive part is a subcase of the proposed approach.

Another important aspect of the proposed formulation is the use of the K as the only hyperparameter, which is one of its biggest strengths. Since it reduces the amount of hyperparameters of the traditional ones while enabling a wider range of models to specify.

Thus, a fundamental challenge in the proposed formulation is determining the optimal number of non-zero coefficients. To address this, a straightforward methodology has been developed based on the AICc (Akaike information criterion corrected). In this study, the assumption is made that the residuals of the estimated models are independent and normally distributed. This assumption allows for the approximation of the likelihood function using the estimated variance of the residuals.

To determine the appropriate value of K, the value is incrementally increased and the AICc values of the new models are compared with the previous ones. If the AICc value of the new model is smaller, indicating a better fit, the value of K is further increased. However, if the AICc value increases, the process is stopped and the previous model is considered as the final result.

4.3 An example using SARIMAX.jl

This chapter aims to offer the reader a succinct introduction to the SARIMAX.jl package, leveraging the Air Passengers dataset to showcase the package's efficacy and user-friendly design.

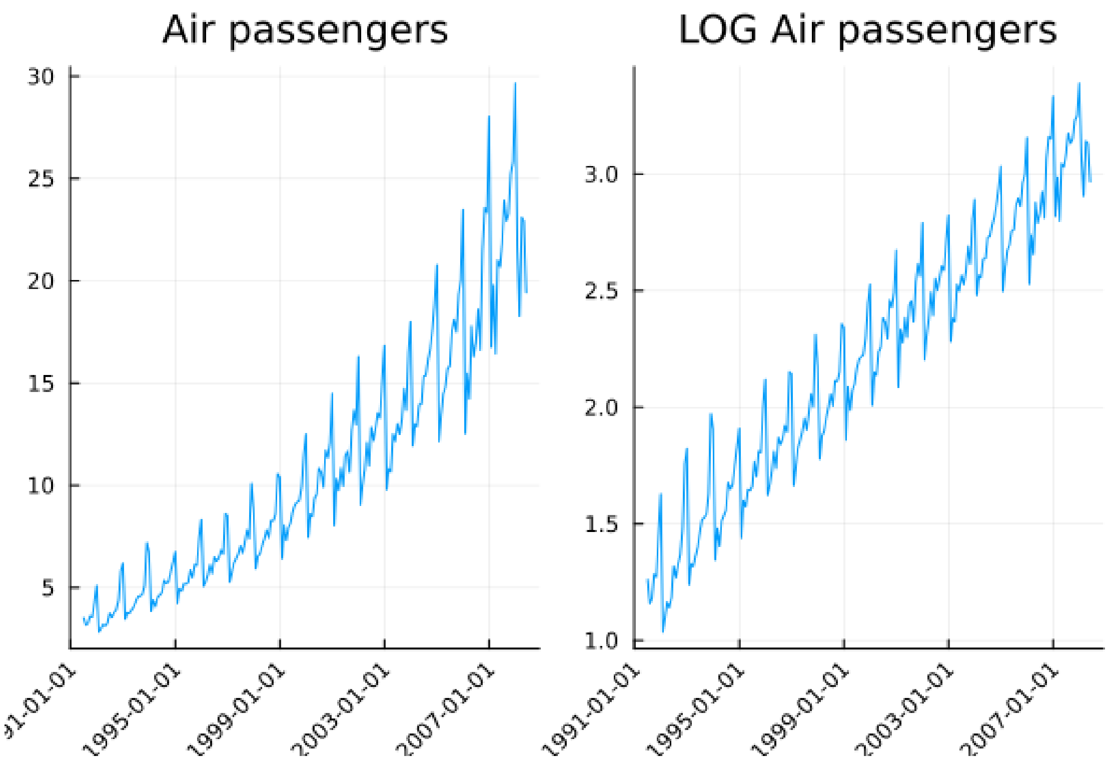

Figure 4.1: Air passengers time series in log scale.

The series depicted in Figure 4.1 has been transformed into a logarithmic scale, a necessary step to handle the escalating variance. The ensuing code snippet exemplifies the straightforward process of crafting a SARIMA model using the SARIMAX.jl package. Notably, the showcased model is a SARIMA(2,0,2)(0,1,2) with a seasonal length of 12 observations. Furthermore, users have the option to include both the constant and drift components, underscoring the package's flexibility and ease of use.

Code 1: Initialization

```
y = loadDataset(AIR_PASSENGERS)
y_log = log.(y)
modelo = SARIMA(
            y_log,
            2, 0, 2;
            seasonality=12,
            P=0, D=1, Q=2,
            allowMean=false,
            allowDrift=true
    )
```

The next step is to fit the model. This is the stage where the package distinguishes itself. Firstly, the default solver is Ipopt.jl (WäCHTER; BIEGLER, 2006), a non-linear solver based on the Interior Points method that yields optimal solutions in linear optimization problems. Since SARIMAX.jl is structured using JuMP.jl (LUBIN et al., 2023), the solver can be easily changed, once it is a parameter of the *fit* function, according to the user's preferences while conforming to the formulation's properties. For instance, if the MA part is being modeled, the solver must be capable of handling non-linear optimization problems.

Secondly, the user can choose the formulation of the problem. Options include the non-linear formulation with an objective function aiming to minimize the sum of squared errors, the non-linear formulation with an objective function aiming to maximize the likelihood, and the formulation in two stages presented in Section 4.1.4. Additionally, users can choose to silence the output of the solver or analyze the iterations.

Code 2: Fit

```
fit!(modelo; objectiveFunction="mse")
```

The final step in the forecasting process involves utilizing the `predict!` function to generate predictions for the time series. Additionally, the package allows for the simulation of scenarios based on both the mean and the standard deviation of the residuals.

Code 3: Predict and simulate

```
predict!(modelo; stepsAhead=24, displayConfidenceIntervals=true)
simulated = simulate(modelo, 24, 250)
```

The results, depicted in Figure 4.2, demonstrate the forecast of the time series and the confidence intervals of the forecast.

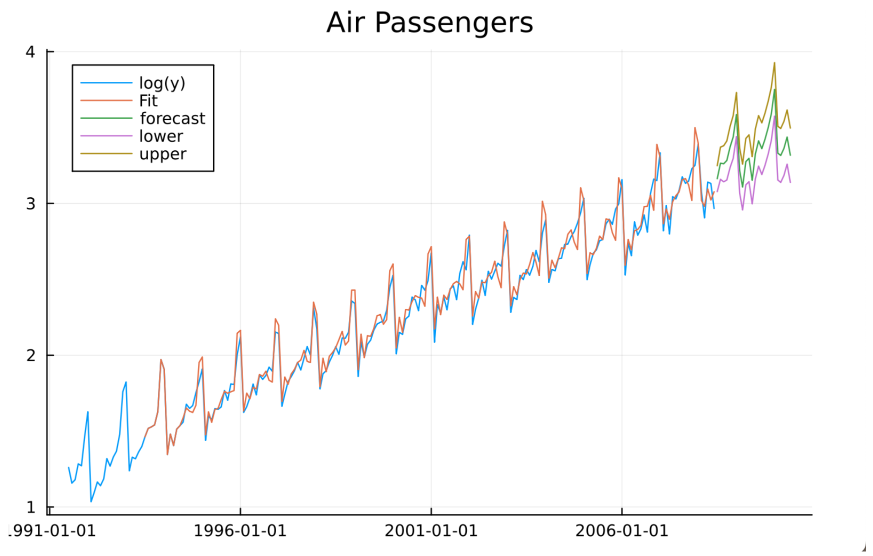

Figure 4.2: Fit in sample, forecast, and confidence intervals (97.5% and 2.5%)

In Figure 4.3, it is possible to notice the forecast together with the scenarios.

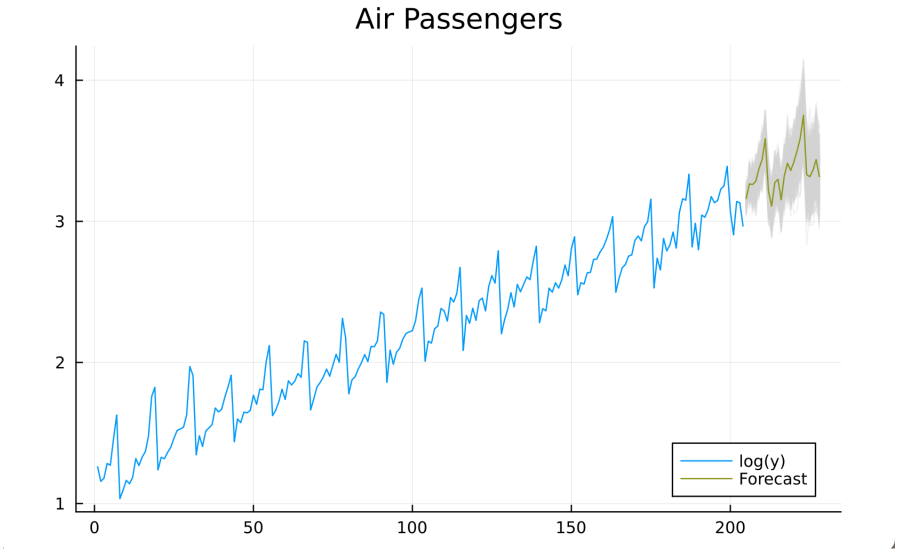

Figure 4.3: Forecast, and scenarios

Moreover, the package implements the `auto` function, uses the Hyndman-Khandakar(HYNDMAN; KHANDAKAR, 2008) algorithm for the automatic specification of ARIMA models with the estimation techniques of SARIMAX.jl. One advantage of combining both is that many models that have numerical error in the estimation process in other packages can be estimated using SARIMAX.jl, enhancing the algorithm's performance.

Code 4: Auto

```
modelo = auto(y_log; seasonality = 12, assertStationarity=true)
```

5
Advancing the state of the art

As discussed in Chapter 4, the ARIMA model is now being formulated within an optimization framework. This approach not only aims to enhance the precision of model estimation but also provides increased flexibility that was previously difficult to achieve. By framing the problem as an optimization task, it becomes easier to incorporate various constraints, regularization techniques, and advanced features, thus allowing users to choose the most suitable optimization method for their specific needs. This shift represents a development in time series modeling, offering the potential for more adaptable and sophisticated model specifications.

5.1
Objective function

A pivotal aspect of this modeling framework is its inherent adaptability, wherein the objective function of the model can be effortlessly modified without influencing the essential structure of the estimation process. Notably, the ubiquity of Least Squares estimation is attributed to its convex and differentiable objective function, specifically the sum of squared errors. However, this metric exhibits vulnerability to outliers, prompting the exploration of alternatives. An effective strategy for robustness against outliers involves replacing the sum of squared errors with the sum of absolute errors.

$$\underset{c,\phi_t,\theta_t\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} |\epsilon_t| \tag{5-1}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \ldots, T \tag{5-2}$$

This reformulation, utilizing the sum of absolute errors, transforms the optimization problem into a linear convex model. Such adaptability ensures the versatility of the ARIMA model, allowing for the incorporation of diverse optimization objectives to better suit the characteristics of the time series data under consideration.

5.2 L0 regularization

In order to introduce L0 regularization, consider representing an ARIMA model through a parameter vector denoted as $\boldsymbol{\Psi} = \{\phi, \theta\}$. Unlike other formulations, this approach lends itself to a concise expression through constraints, providing a distinctive perspective on modeling sparsity within the framework.

$$\underset{c,\phi_t,\theta_t\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{5-3}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \ldots, T \tag{5-4}$$

$$||\Psi||_0 \leq K \tag{5-5}$$

Here, the optimization problem introduces a sparsity-inducing constraint, where $||\Psi||_0$ represents the count of non-zero parameters in the vector $\boldsymbol{\Psi}$, and K signifies the predefined limit on the number of non-zero parameters. This constraint promotes a parsimonious representation of the ARIMA model, aligning with the principle of selecting a concise set of influential parameters while effectively capturing the temporal dynamics of the time series.

5.3 L1 regularization

Within the realm of regularization techniques, the p-norm of a vector μ is defined as

$$||\mu||_p = \left(\sum_{i=1}^{T} |\mu_i|^p \right)^{\frac{1}{lp}}$$

One of the prevalent regularization methods is the L1 regularization, commonly known as Lasso (TIBSHIRANI, 1996). Particularly advantageous in optimization models, the L1 regularization lends itself to a linear formulation and also guarantees sparse and robust properties (XU; CARAMANIS; MANNOR, 2010). Let $\boldsymbol{\Psi} = \{\phi, \theta\}$ represent the parameter vector. The inclusion of Lasso regularization in the model is articulated as:

$$\underset{c,\phi_t,\theta_t,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 \tag{5-6}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \ldots, T \tag{5-7}$$

$$\left(\sum_{i=1}^{N} |\Psi_i| \right) \leq \rho \tag{5-8}$$

Here, λ serves as a pre-specified parameter determining the extent of regularization. The model is more commonly expressed in the Lagrangian form, where the constraint 5-8 is introduced as a penalty term in the objective function:

$$\underset{c,\phi_t,\theta_t,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 + \lambda \left(\sum_{i=1}^{N} |\Psi_i| \right) \tag{5-9}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \ldots, T \tag{5-10}$$

$$\tag{5-11}$$

5.4 L2 regularization

Another widely adopted form of regularization is L2 regularization, commonly referred to as Ridge regression. This regularization technique is particularly useful when dealing with datasets exhibiting multicollinearity. The prevalent and widely applied formulation of Ridge regression is expressed in the Lagrangian form. In this formulation, a penalty term is introduced to the objective function, resembling the one discussed in section 5.3.

$$\underset{c,\phi_t,\theta_t,\epsilon_t}{\text{minimize}} \quad \sum_{t=1}^{T} \epsilon_t^2 + \lambda \sqrt{\left(\sum_{i=1}^{N} \Psi_i^2 \right)} \tag{5-12}$$

$$\text{subject to} \quad y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \quad \forall t \in 1, \ldots, T \tag{5-13}$$

6
Results

In this results chapter, the main goal is to substantiate certain claims made in Chapter 4. The first claim suggests that modeling ARIMA as an optimization problem to leverage various methodologies within the field is beneficial for the accuracy of the estimation process. The second claim, an extension of the first, posits that this approach enhances the stability and robustness of the estimation process, reducing susceptibility to numerical issues or divergence in estimation methods.

To validate this claims the proposed methodology was tested in two different experiments. The first aims to assess the model's fit capacity. The second experiment focused on the model's prediction capacity by analyzing its performance on M4 competition(MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2020) monthly series.

A third experiment detailed in this chapter was extracted from (DUARTE et al., 2023) and demonstrate the performance of the proposed specification method using monthly series of the M3 competition (MAKRIDAKIS; HIBON, 2000).

6.1
Estimation experiment

To investigate these claims, a controlled experiment was conducted. The initial step involved generating a dataset comprising 1000 ARMA processes, each with 100 observations, using Algorithm 1.

The generated dataset facilitated a comparative analysis of the in-sample fit for five ARIMA implementations: statsmodels, pmdarima, StateSpaceModels.jl, forecast (R language), and SARIMAX.jl (MSE objective function). Each model was fitted utilizing the autoregressive and moving average orders specified in the time series process generator, incorporating all available observations. Various statistics, including AIC, AICc, BIC, Log likelihood, and MSE, were extracted for each series. To ensure consistency in metric calculations across different implementations and methods, a standardized procedure was adopted. Different implementations and methods tend to diverge in some aspects of metric computation, making it essential to use a uniform approach for accurate comparison. The coefficients estimated by statsmodels, pmdarima, StateSpaceModels.jl, and forecast (R language) were used to create a model using SARIMAX.jl with fixed coefficients. The model was then fitted, and the

Algorithm 1: ARMA Process Generation

Initialize:
  Set seed of 12345
ARMA Process Generation Loop (1000 iterations):
  while #`process` < 1000 **do**
    Randomly select orders `p` and `q` from [0, 5]
    Generate random AR and MA coefficients from
    the interval [-1.0, 1.0]
    Check if AR process is stationary and MA is invertible
    The check is made by the analysis of the roots of the
    characteristic polynomial of both processes
    if Stationary **and** Invertible **then**
      Generate time series (`y`) using the generated coefficients
      Store ARMA process, coefficients, and orders
    end if
  end while

metrics were extracted. Consequently, all models were compared based on the metrics computed using the SARIMAX.jl framework.

Additionally, the occurrence of divergences or numerical errors during the estimation process was recorded for each ARIMA implementation. Consequently, for each series, the best model was determined based on each metric. In instances where a model diverged, i.e a numeric error occurred during the estimation process, no metric was stored, and an indication of an estimation problem was recorded.

Table 6.1 demonstrates that SARIMAX.jl (using the MSE objective function) had a superior performance than other models across all metrics, except for computation time, supporting the initial hypothesis. The row labeled "Tied Models" indicates the percentage of series where all models performed equally. Furthermore, Table 6.2 reinforces these findings, showing that SARIMAX.jl delivered superior results across all statistical measures, confirming a difference in performance metrics compared to the other models.

	AIC	**BIC**	**AICc**
statsmodels, pmdarima	0.9	0.9	0.9
StateSpaceModels.jl	1.0	1.0	1.0
Rforecast	0.9	0.9	0.9
SARIMAX.jl (MSE)	89.1	89.1	89.1
Tied Models	8.0	8.0	8.0
	MSE	**Log likelihood**	**Time**
statsmodels, pmdarima	0.8	0.7	0.0
StateSpaceModels.jl	0.8	0.9	20.7
Rforecast	1.2	1.2	64.1
SARIMAX.jl (MSE)	89.1	89.1	15.1
Tied Models	8.0	8.0	0.0

Table 6.1: Percentage of the cases the model was the best

AIC				
	mean	**median**	**min**	**max**
Statsmodels	13.261	9.716	-47.714	380.316
pmdarima	13.261	9.716	-47.714	380.316
StateSpaceModels.jl	13.533	9.112	-47.714	494.279
Rforecast	14.293	9.904	-47.714	395.883
Sarimax.jl	5.722	7.155	-118.463	61.913
BIC				
	mean	**median**	**min**	**max**
Statsmodels	23.593	19.091	-45.109	403.763
pmdarima	23.593	19.091	-45.109	403.763
StateSpaceModels.jl	23.494	18.399	-45.109	512.515
Rforecast	24.945	19.579	-45.109	421.935
Sarimax.jl	16.054	17.110	-102.832	77.544
AICc				
	mean	**median**	**min**	**max**
Statsmodels	13.762	10.135	-47.673	382.316
pmdarima	13.762	10.135	-47.673	382.316
StateSpaceModels.jl	14.003	9.508	-47.673	495.496
Rforecast	14.837	10.385	-47.673	398.355
Sarimax.jl	6.223	7.567	-117.559	62.816
MSE				
	mean	**median**	**min**	**max**
Statsmodels	1.086	0.988	0.573	34.302
pmdarima	1.086	0.988	0.573	34.302
StateSpaceModels.jl	1.337	0.986	0.573	114.312
Rforecast	1.105	0.991	0.573	38.834
Sarimax.jl	0.966	0.966	0.228	1.667
Loglikelihood				
	mean	**median**	**min**	**max**
Statsmodels	-139.806	-138.479	-302.899	-108.453
pmdarima	-139.806	-138.479	-302.899	-108.453
StateSpaceModels.jl	-140.305	-138.542	-368.564	-108.454
Rforecast	-140.134	-138.652	-308.844	-108.454
Sarimax.jl	-136.164	-137.098	-165.762	-64.903

Table 6.2: Statistics of selected in sample metrics

Exploring more the time performance of the models, it is important to notice that the procedure adopted to measure this metric, was based on the user's perspective. So, the time measured is elapsed time in the fit function(or equivalent functions), which is the time that a user would have to wait for the model's results to be calculated. This procedure tends to increase the time computed, since these functions have some pre-processing of data to build the entries for the solvers. In the case of SARIMAX.jl, it is known that the process of building the optimization JuMP model is significant, so the time spent on the solver is believed to be lower. In this metric, R forecast stood as the fastest model, one of the reasons of such performance is the use of C language for the estimation process, since C is known for its performance.

	Mean	**Median**	**Min**	**Max**
statsmodels	0.0861	0.0448	0.0057	0.7732
pmdarima	0.0880	0.0460	0.0058	0.9303
StateSpaceModels.jl	0.0369	0.0101	0.0003	0.7449
Rforecast	0.0098	0.0050	0.0010	0.1040
Sarimax.jl (MSE)	0.0188	0.0140	0.0034	0.8443

Table 6.3: Time statistics of the model's estimation process and the SARIMAX.jl fit

Table 6.4 reveals a notable trend where SARIMAX.jl (MSE objective function) exhibited no divergence in the estimation of any series, contrasting with some packages that had divergence in the estimation process. This observation lends support to the assertion that the optimization framework enhances the robustness of the estimation process.

Model	**Divergence (%)**
statsmodels	0.0
pmdarima	0.0
StateSpaceModels.jl	9.8
Rforecast	0.6
SARIMAX.jl (MSE)	0.0

Table 6.4: Percentage of the cases the model diverged.

Figure 6.5 depicts 4 examples of series (blue) with the specification of the ARMA process that generates it and the SARIMAX.jl fit in sample (red). In this series, both StateSpaceModels.jl and R forecast had problems in the estimation process. It seems that all series present a seasonal pattern with a high amplitude, which can be an explanation for the divergence in the estimation process, since this high amplitude can cause numerical instability. It is important that the fit in sample produced by SARIMAX.jl is really tied to the series data, which illustrates the model performance in this experiment.

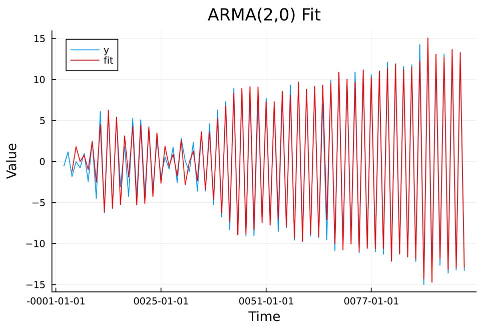

Figure 6.1: ARMA(2,0)

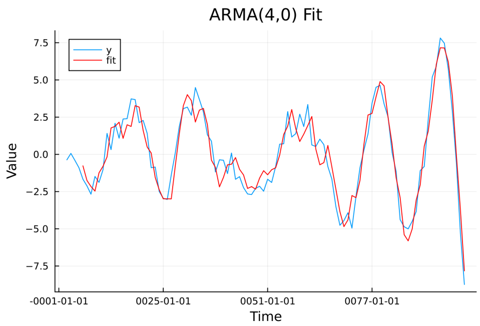

Figure 6.2: ARMA(4,0)

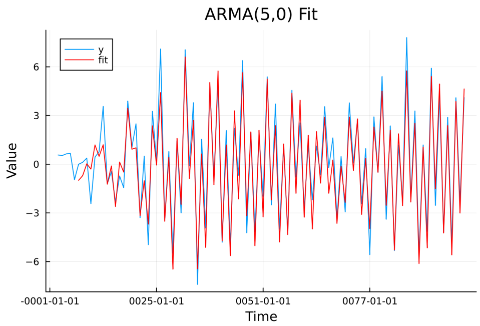

Figure 6.3: ARMA(5,0)

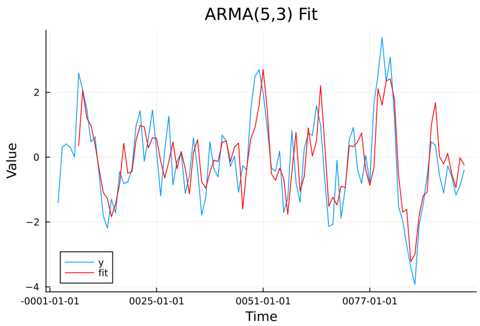

Figure 6.4: ARMA(5,3)

Figure 6.5: Example of series that R forecast diverged and Sarimax fit

6.2 Forecast Results

To evaluate the performance of SARIMAX.jl in forecasting, an experiment was designed utilizing the M4 competition dataset, which comprises 48,000 monthly time series. Due to the unknown underlying processes generating these series, the Hyndman-Khandakar algorithm (HYNDMAN; KHANDAKAR, 2008) for ARIMA specification was employed, as implemented in R forecast. It is important to note that this implementation differs slightly from the original algorithm described in the paper. Despite using the same specification, that considers seasonality and MA component, the chosen models for each series may vary due to differences in the estimation process, which affects the calculation of information criteria and model variance.

For benchmarking purposes, the R forecast *auto.forecast* model was selected, as it was submitted in the M4 competition and demonstrated strong performance on the monthly series. Also, the Naive method, that repeats the last seasonal observation, was used to assert the models' consistency. The forecast horizons were defined according to the M4 competition standards: the short horizon covers the first 6 months, the medium horizon spans from months 7 to 12, and the long horizon ranges from months 13 to 18. Three metrics were used for comparison: MASE, RMSE, and sMAPE. MASE and sMAPE were employed in the M4 competition, while RMSE was included due to its similarity to the SARIMAX.jl objective function. The following tables present statistics for each metric, including mean, median, minimum, and maximum values.

Table 6.5 provides the MASE statistics. It is notable that SARIMAX.jl exhibited competitive performance compared to the R package across all horizons. Particularly in the long horizon, the performance gap between the two models is more pronounced, especially in the maximum values, which explains the higher mean MASE in the total horizon. Additionally, the median MASE for SARIMAX.jl is slightly higher, contributing to this observed difference.

MASE				
	mean	**median**	**min**	**max**
Short Horizon				
Naive	0.744	0.559	0.000	51.979
SARIMAX.jl	0.632	0.478	0.000	51.590
R Forecast	0.624	0.476	0.000	51.590
Medium Horizon				
Naive	1.082	0.811	0.000	44.953
SARIMAX.jl	0.963	0.708	0.000	44.953
R Forecast	0.953	0.698	0.000	44.953
Long Horizon				
Naive	1.364	1.021	0.000	73.572
SARIMAX.jl	1.229	0.891	0.000	73.572
R Forecast	1.211	0.873	0.000	73.540
Total				
Naive	1.063	0.838	0.020	38.546
SARIMAX.jl	0.942	0.733	0.000	41.968
R Forecast	0.929	0.725	0.000	38.546

Table 6.5: Comparison of MASE for Different Models Across Horizons

Table 6.6 further demonstrates the competitiveness of SARIMAX.jl. Unlike the MASE metrics, the maximum values of sMAPE for SARIMAX.jl are actually lower than those for R forecast. However, the median behavior remains consistent, indicating that while SARIMAX.jl can handle extreme cases better in terms of sMAPE, its overall distribution of errors is similar to that of R forecast.

sMAPE				
	mean	**median**	**min**	**max**
Short Horizon				
Naive	11.050	4.777	0.000	188.324
SARIMAX.jl	9.785	4.116	0.000	171.435
R Forecast	9.590	4.116	0.000	184.856
Medium Horizon				
Naive	14.448	7.414	0.000	189.961
SARIMAX.jl	13.897	6.801	0.000	200.000
R Forecast	13.581	6.715	0.000	200.000
Long Horizon				
Naive	17.782	9.276	0.000	191.864
SARIMAX.jl	17.975	8.430	0.000	200.000
R Forecast	17.288	8.326	0.000	200.000
Total				
Naive	14.427	7.791	0.103	190.050
SARIMAX.jl	13.886	7.061	0.000	188.625
R Forecast	13.486	7.030	0.000	194.952

Table 6.6: Comparison of sMAPE for Different Models Across Horizons

Table 6.7 presents a similar behavior to Table 6.6 concerning the maximum and median values. The difference between the median values from both models is lower than in the other metrics, indicating the potential benefit of the objective function used in the estimation process.

RMSE				
	mean	**median**	**min**	**max**
Short Horizon				
Naive	550.986	208.281	0.000	67622.675
SARIMAX.jl	481.870	174.734	0.000	67702.962
R Forecast	471.771	173.342	0.000	67702.962
Medium Horizon				
Naive	710.911	293.602	0.000	65772.737
SARIMAX.jl	658.570	262.108	0.000	65772.737
R Forecast	647.552	259.986	0.000	69329.034
Long Horizon				
Naive	848.428	369.264	0.000	46937.367
SARIMAX.jl	801.048	324.380	0.000	46046.949
R Forecast	783.271	319.177	0.000	46117.079
Total				
Naive	750.327	329.521	1.532	39775.309
SARIMAX.jl	696.481	293.678	0.000	39394.129
R Forecast	682.509	288.122	0.000	40076.768

Table 6.7: Comparison of RMSE for Different Models Across Horizons

In conclusion, the comparative analysis of SARIMAX.jl and R forecast across the MASE, sMAPE, and RMSE metrics demonstrates the competitiveness of SARIMAX.jl in forecasting performance. Specifically, while the maximum values of sMAPE for SARIMAX.jl were lower than those of R forecast, the median values for both models showed consistent patterns across metrics. The RMSE results further highlight the advantage of SARIMAX.jl's objective function in the estimation process, as evidenced by the relatively smaller differences in median values. Despite some variations in maximum values, particularly in the long horizon for MASE, SARIMAX.jl's performance remains robust and comparable to the established R forecast model. This reinforces the potential of SARIMAX.jl as a reliable tool for time series forecasting in various contexts.

6.3 Opt SARI Results

To evaluate the accuracy of the proposed specification model, it was compared against the R language's *auto.forecast* function (HYNDMAN; ATHANASOPOULOS, 2021). The empirical study utilized 1428 monthly time series from the $M3$ competition, a dataset that encompasses a wide variety of time series categorized into six different groups, as outlined in Table 6.8. The proposed model was implemented using JuMP.jl (LUBIN et al., 2023), a mathematical programming framework in the Julia programming language. The results of the experiment demonstrated that, under the same conditions, the proposed model outperformed *auto.forecast* in cases where no MA component was specified.

For each series in the dataset, the data was divided into a training set and a test set. The last 24 observations were designated as the test set, while the remaining data were utilized to fit three models: SARIMA and two versions of the proposed model, referred to as Optimal SARI. Even though the proposed model does not deal with the seasonal differentiation, it was adopted this name to indicate the extension presented in the Section 4.2

Subsequently, each model was employed to generate forecasts for a 24-step ahead horizon. The accuracy of the forecasts was evaluated using four commonly used metrics: Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These metrics provided comprehensive measures of the forecast error for each combination of model and series.

It is noteworthy that the MASE metric relies on a naive model as a reference. To classify each series based on the presence or absence of the seasonal component, a combination of two seasonal tests was employed. Specifically, the p-value of the Kolmogorov-Smirnov (KS) test (KRUSKAL; WALLIS, 1952) and the p-value of the qs test, a variant of the Ljung-Box test (LJUNG; BOX, 1978), were utilized. If the p-value of the KS test is below 0.002 or the p-value of the qs test is below 0.01, the series was considered to have a seasonal component. For seasonal series, the seasonal naive model was utilized to compute the MASE metric, whereas for non-seasonal series, the simple naive model was used.

Category	**Number of series**	**Percentage**
Demographic	111	7.77%
Finance	145	10.15%
Industry	334	23.40%
Macro	312	21.85%
Micro	474	33.19%
Other	52	3.64%
Total	1428	100%

Table 6.8: Number of time series in each category.

Table 6.9 presents the percentage of series in which each model outperformed the others. It can be observed that the SARIMA model exhibited superior performance in at least 52.80% and 55.53% of the series, considering the quadratic and absolute error metrics, respectively. Additionally, Table 6.9 provides a comparison between the two versions of the proposed model. It is evident that, across all metrics, the model employing the quadratic error criterion achieved better forecast results in approximately 53% of the series.

Models	MAPE	MASE	MAE	RMSE
SARIMA	52.80%	58.47%	54.69%	55.11%
Auto ARIMA (quad. error)	47.20%	41.53%	45.31%	44.89%
SARIMA	55.53%	59.17%	55.88%	56.86%
Auto ARIMA (abs. error)	44.47%	40.83%	44.12%	43.14%
Auto ARIMA (quad. error)	53.50%	53.01%	53.01%	53.08%
Auto ARIMA (abs. error)	46.50%	46.99%	46.99%	46.92%

Table 6.9: Percentage of time series where a model showed a better forecast result in each metric.

While the analysis presented in Table 6.9 suggests that the proposed approach did not yield results comparable to SARIMA, it is important to note that the table does not consider the magnitude of the differences between the metrics associated with each model. Therefore, further analysis is necessary to understand the significance of the performance differences observed.

In order to assess the magnitude of these differences, it is crucial to compare the complete distributions of the error metrics. This comparison is illustrated in Figure 6.6 using box plots. The box plots reveal a similar behavior of the three methods for each of the four metrics, both in terms of median values and variability. However, in terms of the prevalence of outliers, it is evident that SARIMA tended to produce fewer upper outliers. It is worth noting that a logarithmic scale was employed in the figure for the purpose of facilitating the graphical analysis.

In addition to the insights gained from the graphical analysis, it is crucial to employ statistical tests to assess the significance of the observed differences. Due to the nature of the data, particularly the error metrics, the assumption of normality cannot be justified. Consequently, a Wilcoxon test (WILCOXON, 1945), a non-parametric test that compares the location of two distributions, was conducted. This test is commonly used as a median comparison test and serves as a non-parametric alternative to the traditional t-test.

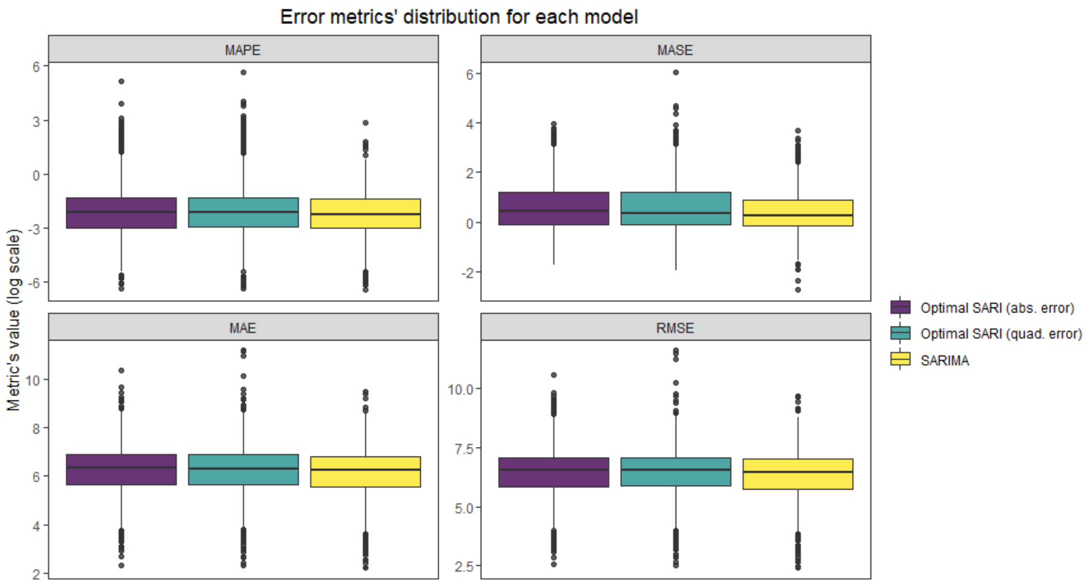

Figure 6.6: Boxplots comparing the distribution of each error metric (in log scale) for each model.

The bilateral version of the test was employed to compare the distributions of the three model combinations. Table 6.10 presents the results of each test in terms of p-values. The analysis reveals no compelling evidence to reject the null hypothesis of equality between the location of the distributions when comparing the two versions of the proposed model. Conversely, when comparing the proposed model to the SARIMA model using the MASE metric, there is clear evidence of a significant difference at conventional levels of significance. However, to reach the same conclusion for the other metrics, a significance level of at least 10% is required for the quadratic error model, and at least 5% for the absolute error model.

Models Compared	**MAPE**	**MASE**	**MAE**	**RMSE**
SARIMA vs **Auto ARIMA (quad. error)**	0.0585	< 0.01	0.0514	0.0607
SARIMA vs **Auto ARIMA (abs. error)**	0.0248	< 0.01	0.0286	0.0258
Auto ARIMA (quad. error) vs Auto ARIMA (abs. error)	0.7290	0.4195	0.7996	0.7220

Table 6.10: P-value of the bilateral Wilcoxon test, for all three model combinations.

Considering the dataset's division into six distinct categories of time series, it is crucial to evaluate the accuracy of the proposed models within each category, as time series within a category may possess unique characteristics. To examine the potential impact of time series category on model accuracy, separate analyses were conducted for each category, following the procedures described earlier.

Table 6.11 presents the percentage of time series within each category where a specific model yielded superior forecast results, indicated by a lower error metric value. It is worth noting that since these percentages are complementary between the two models considered, the results for the first model in each block are displayed. Upon examining Table 6.11, it becomes evident that the proposed model with a quadratic error objective function outperformed SARIMA in the Industry and Other categories across all metrics. Similarly, the comparison between SARIMA and the proposed model with an absolute error objective function revealed better accuracy for the latter in these two categories, except for the MASE metric in the Industry category. Notably, the results indicate that the two versions of the proposed model exhibited similar performance, with percentages hovering around 50%, except for the industrial category, where the model with a quadratic error objective function achieved better results in approximately 58% of the time series.

SARIMA vs Auto ARIMA (quadratic error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	60.36%	72.07%	62.16%	61.26%
Finance	53.10%	68.97%	53.34%	50.34%
Industry	46.40%	49.40%	45.81%	44.31%
Macro	58.33%	60.58%	56.09%	54.49%
Micro	53.38%	58.65%	59.92%	64.14%
Other	38.46%	44.23%	44.23%	46.15%
SARIMA vs Auto ARIMA (absolute error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	64.86%	72.07%	65.77%	65.77%
Finance	54.48%	66.21%	51.03%	52.41%
Industry	49.40%	52.69%	49.10%	47.90%
Macro	58.33%	61.22%	56.09%	56.41%
Micro	57.81%	58.44%	60.50%	63.92%
Other	40.38%	48.07%	48.07%	46.15%
Auto ARIMA (quadratic error) vs Auto ARIMA (absolute error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	51.35%	49.55%	49.55%	51.35%
Finance	49.66%	48.27%	48.27%	50.34%
Industry	58.38%	58.68%	58.68%	58.98%
Macro	53.53%	52.56%	52.56%	50.96%
Micro	52.11%	51.48%	51.48%	51.69%
Other	50.00%	53.85%	53.85%	51.92%

Table 6.11: Percentage of time series, in each category, where a model showed better forecast performance across metrics. The percentages represent the first model in each comparison section.

Continuing with the previous analysis, Figure 6.7 presents the distribution of error metrics across different time series categories. Notably, each metric exhibited distinct characteristics across the series categories. For instance, the demographic time series category displayed greater variability in its results compared to the Micro category. This observation highlights the importance of considering the specific characteristics of each time series category when assessing forecast accuracy.

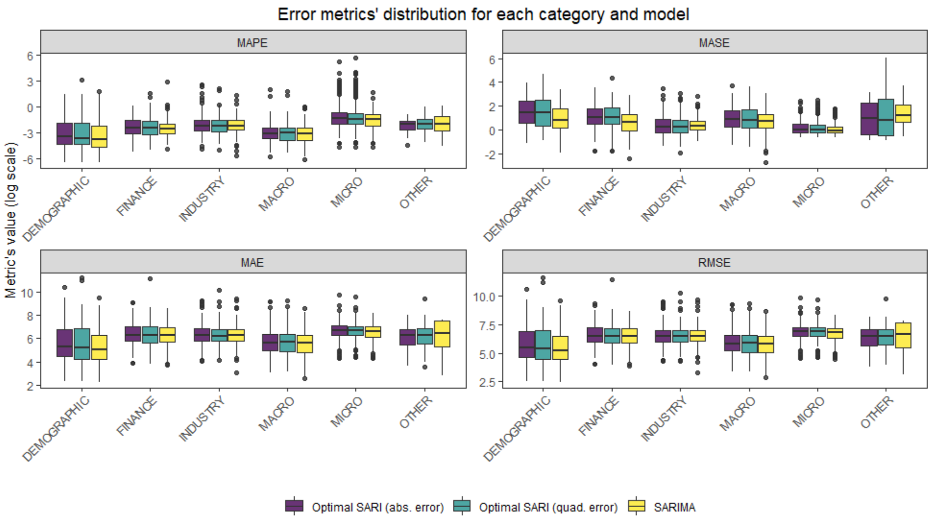

Figure 6.7: Boxplots comparing the distribution of each error metric (in log scale) for each model.

Analyzing the MAPE results, similar behavior was observed, particularly in the Industry and Macro categories. For demographic time series, the SARIMA model exhibited the smallest median MAPE. In contrast, for the Finance and Other categories, the key distinction between the models lay in their variability. SARIMA showed a seemingly smaller variance in the Finance category, while both versions of the proposed model demonstrated lower variability in the MAPE metric than the benchmark methodology in the Other category. In the Micro category, the presence of outliers was more frequent in the proposed models.

Regarding the MASE metric, SARIMA displayed a notably smaller median MASE than both proposed models in the Demographic and Finance categories. For the remaining categories, similar results were observed in terms of median, with SARIMA tending to exhibit less variability. The MAE and RMSE metrics showed comparable patterns, with all three models yielding similar results, except in the Other category where the proposed models showed potentially lower variability than SARIMA.

Furthermore, Table 6.12 presents the p-values obtained from the bilateral Wilcoxon test used to compare the performance of the three models within each metric and time series category. Specifically, comparing the proposed model with a quadratic error objective function to SARIMA, the test indicated a significant difference only in the Macro and Micro categories, with significance levels of 10% and at least 2%, respectively. The MASE metric revealed the most pronounced differences between the methodologies, except in the Industry and Other categories where there was insufficient evidence to reject the equality hypothesis. Similarly, the MAE and RMSE metrics exhibited similar patterns, indicating divergent performances between the models primarily in the Micro category.

SARIMA vs Auto ARIMA (quadratic error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	0.3470	< 0.01	0.3343	0.3250
Finance	0.5611	< 0.01	0.7516	0.7773
Industry	0.5145	0.1639	0.4096	0.3728
Macro	0.0918	0.0155	0.1616	0.1801
Micro	0.0171	< 0.01	< 0.01	< 0.01
Other	0.6514	0.3089	0.3343	0.4685
SARIMA vs Auto ARIMA (absolute error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	0.1464	< 0.01	0.1672	0.1441
Finance	0.5763	< 0.01	0.8886	0.7751
Industry	0.9685	0.4374	0.8838	0.8730
Macro	0.0828	< 0.01	0.1453	0.1785
Micro	< 0.01	< 0.01	< 0.01	< 0.01
Other	0.2908	0.2084	0.3343	0.3120
Auto ARIMA (quadratic error) vs Auto ARIMA (absolute error)				
Category	**MAPE**	**MASE**	**MAE**	**RMSE**
Demographic	0.6367	0.7571	0.7827	0.7239
Finance	0.9687	0.8709	0.8599	0.9542
Industry	0.5208	0.5317	0.4882	0.4427
Macro	0.8819	0.8248	0.9471	0.9552
Micro	0.8166	0.6689	0.8609	0.8262
Other	0.5783	0.8479	0.7923	0.7231

Table 6.12: P-value of the bilateral Wilcoxon test for all three model combinations, categories, and metrics.

It is possible to notice that the tests indicate similar conclusions in the comparison of the proposed model with absolute error objective function and SARIMA. Significant differences were detected in time series of the Micro category, in terms of all considered metrics. Using the MAPE metric, the test also indicates significant differences in the Macro category, considering a significance level of at least 10%. Just like before, the MASE metric indicates the major differences, with the industry and other categories being the only ones in which the test did not indicate a significant difference in the forecast results. It is also important to highlight that the test did not find evidence to reject the equality hypothesis for the two versions of the proposed model, considering all metrics and categories.

So, after all this analysis, it is possible to conclude that the proposed methodology was not able to show a better forecast performance than SARIMA. On the other hand, it is necessary to acknowledge that this analysis aimed to identify how distant this first approach is from the SARIMA model. Despite considering only the AR components, this new methodology was able to match the forecast performance of the benchmark in many time series across different categories.

However, in an attempt to understand how the model would perform in a more fair scenario compared to the SARIMA model, the same error metrics were computed, but now only for the series for which SARIMA did not choose any MA component. This enabled the comparison of the forecast performance of the model for series that only have the AR component. Table 6.13 shows the number of remaining series in each category after applying this filtering.

Category	**Number of Series**	**Percentage**
Demographic	21	5.54%
Finance	35	9.23%
Industry	63	16.62%
Macro	78	20.58%
Micro	155	40.90%
Other	27	7.13%
Total	379	100%

Table 6.13: Number of time series that SARIMA did not choose any MA component in each category.

When considering only the selected set of time series with AR components, Figure 6.8 demonstrates that the proposed model consistently outperforms SARIMA in terms of each error metric. Moreover, as previously observed, the performance of the two versions of the proposed model appears to be highly comparable. The box plots in Figure 6.9 further support these findings, illustrating consistent superiority across different categories.

In summary, although additional refinements are required for the proposed approach to consistently outperform SARIMA across all cases, recent findings suggest that the model demonstrates superior performance when applied to series with exclusively autoregressive (AR) components, surpassing the benchmark model.

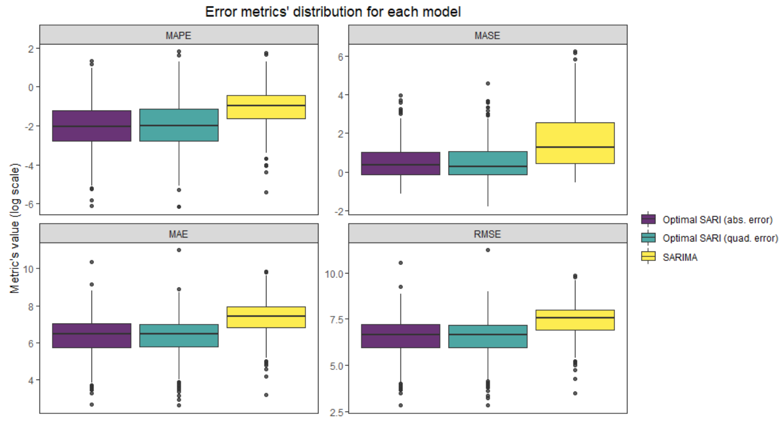

Figure 6.8: Boxplots comparing the distribution of each error metric (in log scale) for each model, considering only the series without MA component.

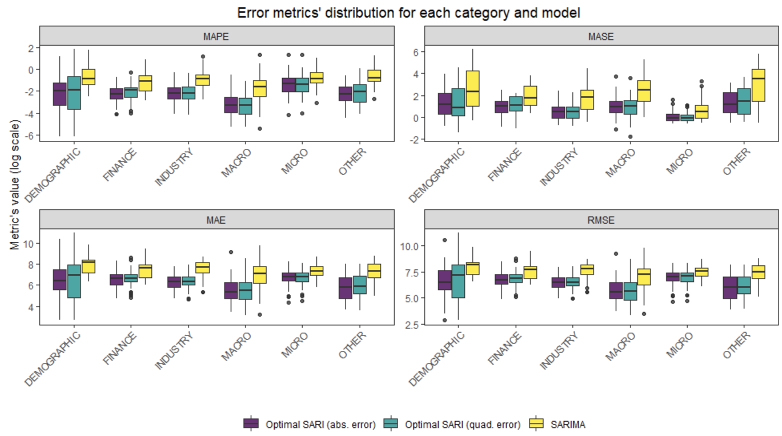

Figure 6.9: Boxplots comparing the distribution of each error metric (in log scale) for each model and each category, considering only the series without MA component.

7
Conclusion and future work

This dissertation presents SARIMAX.jl, a Julia package designed for ARIMA process estimation. Leveraging Julia's computational capabilities and advanced optimization techniques, SARIMAX.jl offers a new approach to time series modeling. Chapter 4 on Estimation and Specification explains the theoretical concepts behind ARIMA models and their integration within the proposed optimization framework.

The main contribution of this work is the separation of model formulation from the estimation process, allowing for the incorporation of various state-of-the-art techniques in ARIMA estimation and giving users the flexibility to add knowledge-based constraints to the model. SARIMAX.jl employs advanced optimization techniques to enhance stability, robustness, and accuracy in modeling ARIMA processes.

Additionally, the proposed modeling approach offers significant flexibility. The optimization framework allows for the integration of regularization techniques and supports diverse objective functions, making SARIMAX.jl a versatile tool for time series analysts. Through a comparative study, SARIMAX.jl demonstrates superior performance across various in-sample metrics and a competitive performance when compared to the R forecast package in the M4 competition monthly series, establishing it as a reliable open-source option for time series modeling.

Furthermore, this dissertation proposes a mixed-integer optimization approach for the specification and estimation of a specific subset of SARIMA models, known as autoregressive integrated (ARI) models. This approach ensures global optimality in parameter estimation and the specification of the integration order and autoregressive part.

In conclusion, SARIMAX.jl represents a competitive tool in time series modeling, characterized by computational sophistication, optimization strength, and reliability. This work paves the way for a series of extensions that can benefit users across various fields, enhancing their ability to model and forecast time series data with greater accuracy and flexibility.

7.1
Future work

Future development of SARIMAX.jl presents several promising directions to enhance its capabilities as a leading tool in time series modeling. A primary focus is incorporating advanced regularization methods into the optimization framework. Integrating techniques such as L1 and L2 regularization will enable users to effectively address outliers and multicollinearity.

Another key area involves extending the optimal specification method to include the Moving Average (MA) component. This will improve SARIMAX.jl's ability to handle the complexities of time series modeling by determining the optimal orders for both the Autoregressive (AR) and MA components. This enhancement aims to provide a comprehensive framework for time series analysts, ensuring the tool remains relevant for diverse real-world data.

Exploring the integration of stationarity and invertibility constraints into the optimization framework is another promising direction. This effort will refine the estimation process and ensure that resulting models adhere to essential statistical properties.

Finally, the use of high-performance computing, such as Graphical Processing Units (GPUs), can significantly enhance the applicability of ARIMA models in commercial applications, enabling faster and more efficient processing of large datasets.

8
Bibliography

ATIQUE, S. et al. Forecasting of total daily solar energy generation using arima: A case study. In: **2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)**. [S.l.: s.n.], 2019. p. 0114–0119.

BERTSIMAS, D.; DUNN, J.; PASCHALIDIS, A. Regression and classification using optimal decision trees. In: **2017 IEEE MIT Undergraduate Research Technology Conference (URTC)**. [S.l.: s.n.], 2017. p. 1–4.

BEZANSON, J. et al. Julia: A fresh approach to numerical computing. **SIAM review**, SIAM, v. 59, n. 1, p. 65–98, 2017. Disponível em: <https://doi.org/10.1137/141000671>.

BOX, G. E. P.; JENKINS, G. M. **Time Series Analysis: Forecasting and Control**. [S.l.]: Wiley, 1970.

BREIMAN, L. **Classification and Regression Trees**. 1st. ed. [S.l.]: Routledge, 1984.

CANOVA, F.; HANSEN, B. Are seasonal patterns constant over time? a test for seasonal stability. **Journal of Business & Economic Statistics**, v. 13, p. 237–52, 02 1995.

CHEN, P. et al. Time series forecasting of temperatures using sarima: An example from nanjing. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 394, n. 5, p. 052024, jul 2018. Disponível em: <https://dx.doi.org/10.1088/1757-899X/394/5/052024>.

DABRAL, P. P.; MURRY, M. Z. Modelling and forecasting of rainfall time series using sarima. **Environmental Processes**, v. 4, n. 2, p. 399–419, 2017. ISSN 2198-7505. Disponível em: <https://doi.org/10.1007/s40710-017-0226-y>.

de O. Santos Júnior, D. S.; de Oliveira, J. F.; de Mattos Neto, P. S. An intelligent hybridization of arima with machine learning models for time series forecasting. **Knowledge-Based Systems**, v. 175, p. 72–86, 2019. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705119301327>.

de Oliveira, E. M.; Cyrino Oliveira, F. L. Forecasting mid-long term electric energy consumption through bagging arima and exponential smoothing methods. **Energy**, v. 144, p. 776–788, 2018. ISSN 0360-5442. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360544217320820>.

DICKEY, D. A.; FULLER, W. A. Distribution of the estimators for autoregressive time series with a unit root. **Journal of the American Statistical Association**, Taylor & Francis, v. 74, n. 366a, p. 427–431, 1979.

DIVISEKARA, R. W.; JAYASINGHE, G. J. M. S. R.; KUMARI, K. W. S. N. Forecasting the red lentils commodity market price using sarima models. **SN**

Business & Economics, v. 1, n. 1, p. 20, 2020. ISSN 2662-9399. Disponível em: <https://doi.org/10.1007/s43546-020-00020-x>.

DUARTE, L. F. C. et al. Mixed integer optimization for sarima specification and estimation. In: GALOá. **ANAIS DO SIMPóSIO BRASILEIRO DE PESQUISA OPERACIONAL**. São José dos Campos, 2023. Accessed on 07 Jan. 2024.

DURBIN, T.; KOOPMAN, S. **Time Series Analysis by State Space Methods: Second Edition**. OUP Oxford, 2012. (Oxford Statistical Science Series). ISBN 9780199641178. Disponível em: <https://books.google.com.br/books?id=fafSS3mw-NwC>.

FISHER, R. A. On the mathematical foundations of theoretical statistics. **Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character**, v. 222, n. 594-604, p. 309–368, 1922.

GÓMEZ, V. Automatic model identification in the presence of missing observations and outliers. In: . [s.n.], 1998. Disponível em: <https://api.semanticscholar.org/CorpusID:122522037>.

GOODRICH, R. The forecast pro methodology. **International Journal of Forecasting**, v. 16, p. 533–535, 02 2000.

HANNAN, E. J.; RISSANEN, J. Recursive estimation of mixed autoregressive-moving average order. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 69, n. 1, p. 81–94, 1982. ISSN 00063444. Disponível em: <http://www.jstor.org/stable/2335856>.

HARVEY, A. C. **Forecasting, Structural Time Series Models and the Kalman Filter**. Cambridge: Cambridge University Press, 1990.

HEWAMALAGE, H.; BERGMEIR, C.; BANDARA, K. Recurrent neural networks for time series forecasting: Current status and future directions. **International Journal of Forecasting**, v. 37, n. 1, p. 388–427, 2021. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207020300996>.

HYNDMAN, R.; ATHANASOPOULOS, G. **Forecasting: Principles and Practice**. 3rd. ed. Melbourne, Australia: OTexts, 2021. <https://otexts.com/fpp3>.

HYNDMAN, R. J. Commentaries on the m3-competition. **International Journal of Forecasting**, v. 17, n. 4, p. 537–584, 2001. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207001001194>.

HYNDMAN, R. J.; KHANDAKAR, Y. Automatic time series forecasting: The forecast package for r. **Journal of Statistical Software**, v. 27, n. 3, p. 1–22, 2008. Disponível em: <https://www.jstatsoft.org/index.php/jss/article/view/v027i03>.

JONG, P. de. Smoothing and interpolation with the state-space model. **Journal of the American Statistical Association**, [American Statistical Association, Taylor & Francis, Ltd.], v. 84, n. 408, p. 1085–1088, 1989. ISSN 01621459. Disponível em: <http://www.jstor.org/stable/2290087>.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Journal of Basic Engineering**, v. 82, n. 1, p. 35–45, March 1960. ISSN 0021-9223.

KIM, J.; RICHARD, J.-P. P.; TAWARMALANI, M. Piecewise polyhedral relaxations of multilinear optimization.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, Taylor & Francis, v. 47, n. 260, p. 583–621, 1952. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441>.

KUMAR, M.; ANAND, M. An application of time series arima forecasting model for predicting sugarcane production in india. **Studies in Business and Economics**, Lucian Blaga University of Sibiu, Faculty of Economic Sciences, v. 9, n. 1, p. 81–94, 2014.

KWIATKOWSKI, D. et al. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? **Journal of Econometrics**, v. 54, n. 1, p. 159–178, 1992. ISSN 0304-4076. Disponível em: <https://www.sciencedirect.com/science/article/pii/030440769290104Y>.

LIU, L.-M. Identification of seasonal arima models using a filtering method. **Communications in Statistics - Theory and Methods**, Taylor & Francis, v. 18, n. 6, p. 2279–2288, 1989. Disponível em: <https://doi.org/10.1080/03610928908830035>.

LJUNG, G. M.; BOX, G. E. P. On a measure of lack of fit in time series models. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 65, n. 2, p. 297–303, 1978. ISSN 00063444. Disponível em: <http://www.jstor.org/stable/2335207>.

LUBIN, M. et al. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. **Mathematical Programming Computation**, v. 15, p. 581–589, 2023.

MAKRIDAKIS, S.; HIBON, M. The m3-competition: results, conclusions and implications. **International Journal of Forecasting**, v. 16, n. 4, p. 451–476, 2000. ISSN 0169-2070. The M3- Competition. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207000000571>.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The m4 competition: 100,000 time series and 61 forecasting methods. **International Journal of Forecasting**, v. 36, n. 1, p. 54–74, 2020. ISSN 0169-2070. M4 Competition. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207019301128>.

MARTINEZ, E. Z.; SILVA, E. A. S. d.; FABBRO, A. L. D. A sarima forecasting model to predict the number of cases of dengue in campinas, state of são paulo, brazil. **Revista da Sociedade Brasileira de Medicina Tropical**, Sociedade Brasileira de Medicina Tropical - SBMT, v. 44, n. 4, p. 436–440, Jul 2011. ISSN 0037-8682. Disponível em: <https://doi.org/10.1590/S0037-86822011000400007>.

NING, Y.; KAZEMI, H.; TAHMASEBI, P. A comparative machine learning study for time series oil production forecasting: Arima, lstm, and prophet. **Computers & Geosciences**, v. 164, p. 105126, 2022. ISSN 0098-3004. Disponível em: <https://www.sciencedirect.com/science/article/pii/S009830042200084X>.

NOBRE, F. F. et al. Dynamic linear model and sarima: a comparison of their forecasting performance in epidemiology. **Statistics in Medicine**, v. 20, n. 20, p. 3051–3069, 2001. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.963>.

ORD, K.; LOWE, S. Automatic forecasting. **The American Statistician**, [American Statistical Association, Taylor & Francis, Ltd.], v. 50, n. 1, p. 88–94, 1996. ISSN 00031305. Disponível em: <http://www.jstor.org/stable/2685050>.

PETRICĂ, A.-C.; STANCU, S.; TINDECHE, A. Limitation of arima models in financial and monetary economics. **Theoretical & Applied Economics**, v. 23, n. 4, 2016.

PHAN, T.-T.-H.; NGUYEN, X. H. Combining statistical machine learning models with arima for water level forecasting: The case of the red river. **Advances in Water Resources**, v. 142, p. 103656, 2020. ISSN 0309-1708. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0309170819311546>.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2021. Disponível em: <https://www.R-project.org/>.

ROSSANA, R. J.; SEATER, J. J. Temporal aggregation and economic time series. **Journal of Business & Economic Statistics**, [American Statistical Association, Taylor & Francis, Ltd.], v. 13, n. 4, p. 441–451, 1995. ISSN 07350015. Disponível em: <http://www.jstor.org/stable/1392389>.

ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

SAAVEDRA, R.; BODIN, G.; SOUTO, M. Statespacemodels.jl: a julia package for time-series analysis in a state-space framework. **arXiv preprint arXiv:1908.01757**, 2019.

SEABOLD, S.; PERKTOLD, J. statsmodels: Econometric and statistical modeling with python. In: **9th Python in Science Conference**. [S.l.: s.n.], 2010.

SIAMI-NAMINI, S.; NAMIN, A. S. Forecasting economics and financial time series: ARIMA vs. LSTM. **CoRR**, abs/1803.06386, 2018. Disponível em: <http://arxiv.org/abs/1803.06386>.

SMITH, A. C.; BROWN, E. N. Estimating a state-space model from point process observations. **Neural computation**, v. 15, n. 5, p. 965—991, May 2003. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/089976603765202622>.

SMITH, T. G. et al. **pmdarima: ARIMA estimators for Python**. 2017–. [Online; accessed <today>]. Disponível em: <http://www.alkaline-ml.com/pmdarima>.

TAYLOR, S. J.; LETHAM, B. Forecasting at scale. **The American Statistician**, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018. Disponível em: <https://doi.org/10.1080/00031305.2017.1380080>.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society. Series B (Methodological)**, [Royal Statistical Society, Wiley], v. 58, n. 1, p. 267–288, 1996. ISSN 00359246. Disponível em: <http://www.jstor.org/stable/2346178>.

VAGROPOULOS, S. I. et al. Comparison of sarimax, sarima, modified sarima and ann-based models for short-term pv generation forecasting. In: **2016 IEEE International Energy Conference (ENERGYCON)**. [S.l.: s.n.], 2016. p. 1–6.

VALIPOUR, M. Long-term runoff study using sarima and arima models in the united states. **Meteorological Applications**, v. 22, n. 3, p. 592–598, 2015. Disponível em: <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.1491>.

WILCOXON, F. Individual comparisons by ranking methods. **Biometrics Bulletin**, JSTOR, v. 1, n. 6, p. 80–83, 1945.

WäCHTER, A.; BIEGLER, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical Programming**, v. 106, n. 1, p. 25–57, 03 2006. ISSN 1436-4646. Disponível em: <https://doi.org/10.1007/s10107-004-0559-y>.

XU, H.; CARAMANIS, C.; MANNOR, S. Robust regression and lasso. **IEEE Transactions on Information Theory**, v. 56, n. 7, p. 3561–3574, 2010.

ZENG, Y.; WU, S. **State-Space Models: Applications in Economics and Finance**. [S.l.: s.n.], 2013. 1-347 p. ISBN 978-1-4614-7788-4.