A NEGATIVE BINOMIAL INGARCH MODEL FOR OVERDISPERSED COUNT TIME SERIES

STRUCTURE, PARAMETER ESTIMATION
AND APPLICATION TO REAL DATA

By

Carols Mulamfu

A dissertation submitted to the University of Zambia in partial fulfillment of the requirements for the degree of Master of Science in Statistics

THE UNIVERSITY OF ZAMBIA
LUSAKA

2025

Declaration

I, **Carols Mulamfu**, hereby declare that this dissertation represents my own work (except where otherwise indicated), and that it has not previously been submitted for a degree, diploma or other qualification at this or another university.

Signature: .... Date: ....

Certificate of Approval

This dissertation of Carols Mulamfu has been approved as fulfilling the requirements or partial fulfilment of the requirements for the award of the degree of Master of Science in Statistics by the University of Zambia.

....	....	....
Examiner I	Signature	Date
....	....	....
Examiner II	Signature	Date
....	....	....
Examiner III	Signature	Date
....	....	....
Chairperson Board of Examiners	Signature	Date
....	....	....
Supervisor	Signature	Date

Abstract

This dissertation develops a negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q (negative binomial INGARCH(p, q)) for overdispersed count time series data with potential extreme values. The model is formulated such that its conditional distribution follows the negative binomial distribution, allowing the conditional variance to exceed the conditional mean and dynamically adjust for overdispersion based on past observations. Furthermore, the unconditional variance exceeding the unconditional mean demonstrates the model's capability to capture extreme values in the data.

A simulation study evaluates the finite sample performance of the Yule–Walker, conditional least squares, and maximum likelihood estimation methods for the three sparsely parameterized negative binomial INGARCH(p, q) models. Results indicate that maximum likelihood estimation is the most efficient and reliable approach. The conditional log-likelihood function is maximized numerically using MATLAB's `fmincon` function, with constraints to ensure stationarity and non-negativity of parameters. Conditional least squares estimates serve as initial values to facilitate convergence and enhance stability.

For application, the negative binomial INGARCH(p, q) model is applied to syphilis count data from the R ZIM package, originally sourced from the CDC Morbidity and Mortality Weekly Report CDC MMWR. The dataset consists of weekly syphilis cases in Maryland, United States, from January 2007 to May 2010, with 209 observations. The empirical mean (3.47) and variance (9.28) confirm overdispersion, justifying the use of the negative binomial distribution. Model performance is assessed and compared to the Poisson INGARCH(p, q) and double Poisson INGARCH(p, q) models using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Additionally, tail probabilities of residuals are analyzed to evaluate the models' ability to capture extreme values. Results from AIC, BIC, and tail probability analysis indicate that the negative binomial INGARCH(p, q) model outperforms the Poisson INGARCH(p, q) and double Poisson INGARCH(p, q) models.

To my beloved family and friends, whose unwavering support and encouragement have been the foundation of my academic journey.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. J. Musonda, for his valuable and constructive suggestions throughout my research and the writing of this dissertation. His willingness to give his time so generously has been very much appreciated.

Furthermore, I would like to acknowledge Dr. V. M. Nawa, Dr. I. D. Tembo, and Mrs. Jain for the knowledge they imparted to me as my lecturers during the coursework of my studies, which has been collectively useful in carrying out my research work. I am grateful to the Department of Mathematics and Statistics, School of Natural Sciences, University of Zambia, for the invaluable support throughout my studies. The provision of study materials, access to the Maths Lab, and internet facilities has been instrumental in the completion of my studies. The dedication of the department to fostering a conducive learning environment has greatly contributed to my academic growth and success. I am also thankful to my fellow postgraduate students in the Department of Mathematics and Statistics, both present and past, for their support. Special thanks go to Fizzie Fulumaka, Kanaan Nyirongo, Zielesa Zulu, Edward Kapoka, Dominique Sampa, Bornwell Kasense, and Memory Zulu for their encouragement throughout my research and the writing of this dissertation.

I thank my employers, the Ministry of Education, for offering me study leave to pursue this programme. I further wish to thank Mr. K. Milupi, the head teacher of Mulobezi Secondary School, for his help in processing my study leave. I also thank Mr. M. Mudenda, Mr. Bwalya, Mr. M. Mukelabai, Mr. O. Syang'andu, Mr. M. Syamusokwe, Mr. W. Halwiindi, and all my other workmates at Mulobezi Secondary School for their overwhelming encouragement during my postgraduate studies at the University of Zambia.

To my family and friends, thank you very much for your tireless encouragement throughout my studies, which provided me with extra energy to successfully complete my work. My heartfelt thanks go to my wife, Catherine Muchanga, for her unwavering encouragement and support, especially during moments of doubt.

Finally, I am deeply indebted to many others, too numerous to mention, who contributed in various ways to the success of my studies. May Jehovah God richly bless you all.

Lusaka, January 2025
Carols Mulamfu

Table of Contents

List of Symbols and Terms

Below is a list of some symbols, abbreviations, distributions, and statistical models used in this dissertation, along with their corresponding meanings.

Symbols

θ	Vector of parameters to be estimated.
θ^0	Vector of true parameters.
$l(\theta)$	Log-likelihood function.
$\mathcal{F}_{t-1}$	σ-field generated by a time series process up to time $t-1$.
$\gamma_X(k)$	Autocovariance function of the time series process X_t at lag k.

Abbreviations

YW	Yule–Walker.
CLS	Conditional least squares.
MLE	Maximum likelihood estimation.
MADE	Mean Absolute Deviation Error.
AIC	Akaike Information Criterion.
BIC	Bayesian Information Criterion.

Distributions

$\text{NB}(r,p)$	Negative binomial distribution with parameters $r > 0$ and $0 < p < 1$.
$\text{P}(\lambda)$	Poisson distribution with mean $\lambda > 0$.
$\text{DP}(\lambda,\gamma)$	Double Poisson distribution with parameters $\lambda > 0$ and $\gamma > 0$.

Statistical Models

$\text{GARCH}(p,q)$	Generalized autoregressive conditional heteroscedastic model of order p, q.
$\text{INGARCH}(p,q)$	Integer-Valued $\text{GARCH}(p,q)$.
$\text{PINGARCH}(p,q)$	Poisson $\text{INGARCH}(p,q)$.
$\text{DPINGARCH}(p,q)$	Double Poisson $\text{INGARCH}(p,q)$.
$\text{NBINGARCH}(p,q)$	Negative binomial $\text{INGARCH}(p,q)$.

Chapter 1

Introduction

Time series of counts are commonly observed in real-world applications and play a vital role in understanding patterns and trends over time. Various fields generate such data, including the insurance industry, economics, medicine, epidemiology, queueing systems, meteorology, and communications. These datasets often consist of discrete, non-negative integer values that represent the occurrence of specific events within a given time frame. Examples include the number of road accidents reported daily, patient admissions in hospitals weekly, recorded crime victimizations monthly, transmitted data packets in communication networks hourly, and detected system errors in computing environments in real-time.

Analysing count time series is essential for developing predictive models, making informed decisions, and implementing effective policies. However, these data often exhibit unique statistical properties, such as overdispersion, which require specialized modelling approaches for accurate analysis and forecasting. Overdispersion, where the variance exceeds the mean, is frequently attributed to positive correlations between observed events or variability in event probabilities (see Weiß [45]). To address this challenge, numerous models have been proposed for analysing count time series (see [12, 19–40]), with many researchers employing overdispersed Poisson and binomial regression models. However, the Poisson distribution assumes equal mean and variance, making it unsuitable for overdispersed data. In contrast, the negative binomial distribution, a natural extension of the Poisson distribution, offers greater flexibility by accommodating overdispersion, thus providing a more robust framework for accurate analysis and forecasting.

The Poisson integer-valued generalized autoregressive conditional heteroscedastic (Poisson INGARCH) model, introduced by Ferland [19], is a commonly used model for overdispersed count time series. However, it has key limitations: it does not accommodate covariates, its autocorrelation function is always positive, and its conditional mean equals the conditional variance. This dissertation focuses on the last limitation, as it can compromise model performance in the presence of extreme observations. To address this, we propose a negative binomial INGARCH model that effectively handles both overdispersion and extreme values.

1.1 Statement of the Problem

Despite the development of various models for overdispersed count time series, including the Poisson INGARCH model, significant challenges remain in accurately modelling datasets with extreme observations. The Poisson INGARCH model's assumption that the conditional mean equals the conditional variance limits its ability to capture overdispersion, leading to biased parameter estimates and poor predictive performance. This gap underscores the need for a more flexible modelling framework that can effectively handle overdispersion and extreme values in count time series data.

1.2 Aim of the Study

The aim of this study is to propose a negative binomial INGARCH model that can effectively address overdispersion and extreme values in count time series data, focusing on its structure, parameter estimation, and application to real-world data.

1.3 Research Objectives

(i) To formulate a negative binomial INGARCH model for overdispersed count time series.

(ii) To develop an approach for parameter estimation for the proposed model.

(iii) To apply the proposed model to real-world count data and assess its performance.

(iv) To compare the performance of the negative binomial INGARCH model to that of the Poisson and the double Poisson INGARCH models.

1.4 Research Questions

(i) How can a negative binomial INGARCH model be formulated to effectively model overdispersed count time series data?

(ii) What is an appropriate method for estimating the parameters of the proposed negative binomial INGARCH model?

(iii) How does the proposed negative binomial INGARCH model perform when applied to real-world overdispersed count data?

(iv) How does the performance of the negative binomial INGARCH model compare to that of the Poisson and double Poisson INGARCH models in modelling overdispersed count time series data?

1.5 Significance of the Study

This study contributes to the field of time series analysis by addressing the limitations of the Poisson INGARCH model. The proposed negative binomial INGARCH model provides a more flexible framework for modelling overdispersed count data, leading to improved parameter estimation and forecasting accuracy. This advancement is crucial for applications in fields such as epidemiology, finance, and communications, where accurate modelling of count data is essential for effective decision-making and policy development.

1.6 Literature Review

Time series models for count data have seen extensive development due to their wide-ranging applications in fields such as epidemiology, finance, and environmental studies. Among these, the Poisson INGARCH model, introduced by Ferland et al. [19], has been particularly influential for modeling overdispersed count time series. This model has received significant theoretical attention, with contributions such as Zhu and Wang [52] establishing conditions for higher-order moments and Weiß [45] deriving equations for variance and autocorrelation. Extensions to mixture models by Zhu et al. [52] and advancements in geometric ergodicity and likelihood-based inference by Fokianos et al. [20] have enriched its theoretical foundation.

Despite these advancements, the Poisson INGARCH model exhibits limitations that hinder its applicability to diverse real-world datasets. A key drawback is its assumption that the conditional mean equals the conditional variance, making it unsuitable for datasets exhibiting overdispersion or extreme observations. Heinen [27] proposed an alternative INGARCH framework using the double-Poisson distribution, and Grahramani et al. [25] extended this approach to higher-order models. Additional alternatives include INGARCH models based on the generalized Poisson and Conway-Maxwell Poisson distributions [42, 51]. However, while these models offer improvements in flexibility, they often face challenges in parameter estimation and computational efficiency, particularly for large-scale datasets.

A notable study by Zhu [53] introduced a negative binomial integer-valued GARCH (NBINGARCH) model, which extends the INGARCH framework by incorporating a GARCH-like structure to model conditional variance dynamics in count time series. This study established conditions for stationarity and ergodicity and developed an estimation approach based on maximum likelihood. Additionally, the empirical results demonstrated the NBINGARCH model's superior performance in capturing overdispersion and volatility clustering compared to Poisson-based alternatives. By allowing the variance to exceed the mean, the negative binomial INGARCH model provides a robust framework for handling overdispersion and extreme values in count time series. Furthermore, its ability to account for covariates and complex autocorrelation structures makes it well-suited for real-world applications.

Building on this foundation, this research aims to further explore the negative binomial

INGARCH model by developing efficient parameter estimation techniques and evaluating its performance against Poisson and double-Poisson INGARCH models using real-world datasets.

1.7 Methodology

This section presents the methodology adopted in this study, which is structured into five main subsections corresponding to the key steps of the research process. The steps include model formulation, parameter estimation, simulation study, application to real-world data, and model comparison.

1.7.1 Model Formulation

The negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q, denoted as NBINGARCH(p, q), is designed to handle overdispersed count time series by allowing the conditional variance to exceed the conditional mean. Let $\{X_t\}$ denote a time series of counts, and $\mathcal{F}_{t-1}$ represent the σ-field generated by $\{X_{t-1}, X_{t-2}, \ldots\}$. The NBINGARCH$(p, q)$ process is formulated such that the conditional distribution of X_t follows the negative binomial distribution:

$$X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \tag{1.1}$$

where r is a positive integer representing the number of successes in a sequence of independent Bernoulli trials, and p_t is the success probability parameter that satisfies the relation:

$$\frac{1 - p_t}{p_t} = \lambda_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j}, \tag{1.2}$$

with $\alpha_0 > 0$, $\alpha_i \geq 0$ for $i = 1, \ldots, p$, and $\beta_j \geq 0$ for $j = 1, \ldots, q$. Here, $p \geq 1$ and $q \geq 0$.

This formulation implies that $\text{var}(X_t|\mathcal{F}_{t-1}) > E(X_t|\mathcal{F}_{t-1})$ (see Corollary 3.1.5 in Chapter 3), highlighting how the model dynamically adjusts for overdispersion based on past information, capturing both serial dependence and clustering in the data. Similarly, it confirms that $\text{var}(X_t) > E(X_t)$, demonstrating the model's ability to represent persistent overdispersion across the entire series. This flexibility makes the NBINGARCH(p, q) process particularly well-suited for real-world applications where count data exhibit high variability, such as in epidemiological case counts, insurance claims, and financial transaction counts.

However, to ensure the validity of the NBINGARCH(p, q) model, it is important to consider the stationarity conditions. For the NBINGARCH(p, q) model to be first-order stationary

(stationary in the mean), a necessary and sufficient condition is that all roots of the equation

$$1-\sum_{i=1}^{q}(r\alpha_i+\beta_i)z^i-\sum_{i=q+1}^{p}r\alpha_i z^i=0 \tag{1.3}$$

lie outside the unit circle, meaning their absolute values exceed 1 (see Proposition 3.1.7). This implies that the parameters must satisfy $0 \le \alpha_i < 1$ for $i = 1, \ldots, p$ and $0 \le \beta_j < 1$ for $j = 1, \ldots, q$, which ensures that the model exhibits stable behaviour over time.

1.7.2 Parameter Estimation

The model parameters $\theta = (\alpha_0, \alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)^T = (\theta_0, \theta_1, \ldots, \theta_{p+q})^T$ are jointly estimated using maximum likelihood estimation (MLE). It is natural to estimate θ by maximizing the conditional log-likelihood function $l(\theta)$. However, it turns out that the estimates have no closed form and the numerical optimization methods have to be used. Specifically, the log-likelihood function $l(\theta)$ is maximized with respect to θ for selected values of r, where searching for the maximizer of $l(\theta)$ is implemented in MATLAB by using the constrained nonlinear optimization function `fmincon`. Here the constrained conditions are $\alpha_0 > 0$ and the first-order stationarity condition (see Proposition 3.1.7 in Chapter 3 for this condition). We choose the conditional least squares (CLS) estimates as the corresponding initial values and set $\lambda_0 = \bar{X}$ and $\partial\lambda_0/\partial\theta_i = 0$. The estimate $\hat{r}$ is determined by the r value that yields the smallest Akaike information criterion (AIC) or Bayesian information criterion (BIC).

1.7.3 Simulation Study

A simulation study was conducted to evaluate the finite sample performance of the Yule–Walker (YW), conditional least squares (CLS), and maximum likelihood (ML) estimates for the three most sparsely parameterized NBINGARCH models: NBINGARCH(1), NBINGARCH(2) and NBINGARCH(1,1), given as follows:

$$\begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r,p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} \end{cases}, \quad \begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r,p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} \end{cases} \text{ and } \begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r,p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1\lambda_{t-1}, \end{cases}$$

respectively, where $\lambda_t = (1-p_t)/p_t$. Three set-ups were considered as follows:

(i) NBINGARCH(1) with $(\alpha_0^0, \alpha_1^0, r)^T$=$(A1)(2, 0.3, 1)^T$ and $(A2)(4, 0.3, 2)^T$;

(ii) NBINGARCH(2) with $(\alpha_0^0, \alpha_1^0, \alpha_2^0, r)^T = (B1)(2, 0.4, 0.2, 1)^T$ and $(B2)(3, 0.3, 0.1, 2)^T$;

(iii) NBINGARCH(1,1) with $(\alpha_0^0, \alpha_1^0, \beta_1^0, r)^T = (C1)(2, 0.2, 0.4, 1)^T$ and $(C2)(3, 0.1, 0.3, 2)^T$,

where θ^0 represents the true parameter values of the models. Following Davis and Wu [12], the parameter estimation assumed that r was known. The log-likelihood function was maximized using MATLAB's constrained nonlinear optimization function `fmincon`, with the

CLS estimates as initial values. The imposed constraints were $\alpha_0 > 0$ and the first-order stationarity condition.

For the CLS estimation of NBINGARCH(1,1) models, we set $p^* = \lfloor \sqrt{n} \rfloor$, where $\lfloor X \rfloor$ denotes the integer part of X. The performance of the estimators was assessed using the mean absolute deviation error (MADE), defined as

$$\frac{1}{m} \sum_{j=1}^{m} \left| \hat{\theta}_j - \theta_j^0 \right| .$$

The simulations were conducted with sample sizes of $n = 100, 500$, and 1000, using $m = 200$ replications for each scenario.

1.7.4 Application to Real Data

In this dissertation, the NBINGARCH(p, q) model is applied to an overdispersed count time series of syphilis data. This dataset is included in the R software ZIM-package, sourced from the CDC Morbidity and Mortality Weekly Report CDC MMWR. It consists of the weekly number of syphilis cases in Maryland, United States, from January 2007 to May 2010, comprising a total of 209 observations. The empirical mean (3.47) and variance (9.28) indicate overdispersion, suggesting that a negative binomial distribution may provide a better fit than the Poisson distribution. Other authors who considered this type of data include the Baltimore City Health Department [2], the Centers for Disease Control and Prevention [11], and the Maryland Department of Health and Mental Hygiene [37].

A time plot of the series, which shows no apparent trends or seasonality, confirms stationarity. Autocorrelation function (ACF) and partial autocorrelation function (PACF) plots are used to identify the appropriate order of the NBINGARCH(p, q) model. Model parameters are estimated using the approach detailed in Subsection 1.7.2.

To assess model performance, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are computed for different values of r. The model with the lowest AIC and BIC values is selected as the best-performing model. A Pearson residual analysis is conducted to evaluate the adequacy of the fitted model in capturing the data's dynamics.

1.7.5 Model Comparison

The NBINGARCH(p, q) model's performance is compared to the Poisson INGARCH(p, q) and double Poisson INGARCH(p, q) models using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) as evaluation criteria. Both AIC and BIC are calculated from the log-likelihood functions of each model, with adjustments for the number of parameters to penalise complexity. Lower values of these criteria indicate better model performance by balancing goodness of fit and model simplicity.

In addition to AIC and BIC, the models are further assessed based on their ability to capture extreme values through the analysis of tail probabilities for residuals. Tail probabilities are calculated by evaluating the likelihood of extreme residuals (values lying beyond a specified threshold) under each model. Models that yield higher tail probabilities demonstrate a greater capacity to account for extreme variations in the data, which is particularly important for applications involving overdispersed count time series.

This evaluation approach provides a clear framework for assessing model performance. The use of AIC and BIC ensures that both the overall fit and the trade-offs associated with model complexity are systematically addressed. Additionally, the analysis of tail probabilities highlights each model's ability to represent overdispersion and capture extreme fluctuations effectively. By combining these complementary metrics, the methodology offers a robust and reliable basis for comparing the NBINGARCH(p, q) model to existing alternatives.

1.8 Limitations and Future Work

1.8.1 Limitations

Computational Complexity

The NBINGARCH (p, q) model, while effective in addressing overdispersion in count time series data, can be computationally intensive. The estimation process involves iterative optimization techniques that require significant computational resources and time, especially for large datasets or higher-order models. This limitation could restrict its application in real-time or large-scale data analysis.

Assumptions of Stationarity

The model assumes stationarity in the underlying time series data, which may not always hold in practical applications. Non-stationary data can lead to biased parameter estimates and unreliable predictions. While techniques such as differencing or transformations can address non-stationarity, they may also obscure important features of the data.

Model Interpretability

The NBINGARCH (p, q) model, while effective in capturing overdispersion and temporal dependencies, can be challenging to interpret for non-technical stakeholders. The complexity of the model parameters and their interactions may hinder its adoption in fields where interpretability is critical, such as public health or policy-making. Simplifying the model or developing tools to enhance result interpretability could help address this limitation.

Data Requirements

Reliable parameter estimation for the NBINGARCH (p, q) model often requires large datasets with sufficient observations over time. In cases where data are sparse or incomplete, the model's performance may be compromised. Future work could explore methods to handle small sample sizes or missing data effectively.

1.8.2 Future Work

Improved Computational Efficiency

Future research could focus on developing more efficient algorithms or approximation methods to reduce the computational burden of the model. Techniques such as parallel computing or machine learning-based optimization could provide promising solutions.

Relaxing Stationarity Assumptions

Extending the model to accommodate non-stationary time series would make it applicable to a broader range of datasets. This could involve integrating trend and seasonality components or developing hybrid models that combine INGARCH with other statistical frameworks.

Model Extensions

Future studies could explore:

(i) Incorporating alternative distributions for count data, such as zero-inflated models (which account for excess zeros by modeling them separately) or hurdle models (which distinguish between zero and non-zero counts).

(ii) Developing multivariate extensions of the NBINGARCH (p, q) model for analyzing multiple correlated time series simultaneously.

Application to Diverse Domains

While this study focuses on epidemiological data, future research could investigate applications in other fields such as finance, environmental science, and social media analytics. These applications would demonstrate the versatility and robustness of the model across various disciplines.

Real-Time Implementation

Developing real-time forecasting systems using the NBINGARCH (p, q) model could provide practical tools for decision-making in dynamic environments such as disease outbreak monitoring or stock market analysis.

1.9 Organisation of the Dissertation

The rest of the dissertation is organised as follows:

Chapter 2. This chapter contains the requisite preliminaries needed to understand the content of this dissertation. The style is intentionally concise, as this chapter is designed to serve as a reference rather than a comprehensive exposition. Section 2.1 provides a detailed overview of the negative binomial distribution, including its mathematical formulation, properties, and applications. In Section 2.2, we discuss three commonly used approaches for estimating parameters in time series analysis: Yule–Walker estimation, conditional least squares estimation, and maximum likelihood estimation. Each of these approaches has its own advantages and is suited to different types of time series models. Finally, in Section 2.3, we outline the steps involved in conducting simulation studies, along with their significance in assessing time series models.

Chapter 3. This chapter introduces the negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q, denoted as NBINGARCH(p, q), and outline an approach for parameter estimation. Section 3.1 defines the model, detailing its key assumptions, mathematical formulation, and fundamental properties. We also derive the first- and second-order stationarity conditions and present some equations to compute the autocovariance and autocorrelation functions. Section 3.2 provides a detailed example for the specific case of $p = 1, q = 1$. Finally, in Section 3.3, we discuss the Yule–Walker and conditional least squares estimation methods for specific cases of the NBINGARCH model, followed by the development of the maximum likelihood estimation approach.

Chapter 4. This chapter presents a simulation study to evaluate estimation approaches, followed by a real data example demonstrating the application of the NBINGARCH(p, q) model. Specifically, Section 4.1 compares the finite-sample performance of the Yule–Walker, conditional least squares, and maximum likelihood estimates through a simulation study. In Section 4.2, the NBINGARCH(p, q) model is applied to an overdispersed count time series of syphilis data.

Chapter 5. In this chapter, we discuss the results and implications of the negative binomial INGARCH model, reflecting on its effectiveness in addressing overdispersion in count time series data. The aim and objectives of this dissertation are restated and discussed in Section 5.1 while the employed methodology to achieve these goals is discussed in Section 5.2. In Section 5.3, we discuss the structure of the negative binomial INGARCH(p, q) (NBINGARCH(p, q)) model, including its formulation, suitability for modelling count time series with overdispersion and extreme values, and the stationarity of the model. Approaches for parameter estimation of the model are discussed in Section 5.4. Here, we discuss the Yule–Walker and conditional least squares estimation methods for specific cases of the NBINGARCH(p, q) model, followed by a discussion on the development of the maximum likelihood estimation approach. A simulation study to evaluate parameter estimation approaches is discussed in Section 5.5. Here, the simulation study compares the finite-sample performance of the Yule–Walker, conditional least squares, and maximum likelihood estimates. In Section 5.6, we discuss the application of the NBINGARCH(p, q) model to an overdispersed count time series of syphilis data. In Section 5.7, we discuss the comparison of the performance of the negative binomial INGARCH model to that of the Poisson and the double Poisson INGARCH models.

Chapter 6. In this final chapter, we summarize the main findings of our research and underscore their implications. We offer recommendations based on our insights to guide future actions and initiatives. The summary of the main findings is given in Section 6.1 while the recommendations are given in Section 6.2.

Appendices. These appendices provide supplementary materials supporting the main content of the dissertation. Appendix A presents the data set utilized for the real data example discussed in Chapter 4, offering a detailed overview of the variables and data collection methods. Appendix B contains the MATLAB code employed for the simulation study outlined in Chapter 4, facilitating a deeper understanding of the computational processes used in the analysis. Finally, Appendix C includes the MATLAB code relevant to the real data example in Chapter 4, further illustrating the practical implementation of the methods discussed.

Chapter 2

Preliminaries

This chapter contains the requisite preliminaries needed to understand the content of this dissertation. The style is intentionally concise, as this chapter is designed to serve as a reference rather than a comprehensive exposition. Section 2.1 provides a detailed overview of the negative binomial distribution, including its mathematical formulation, properties, and applications. In Section 2.2, we discuss three commonly used approaches for estimating parameters in time series analysis: Yule–Walker (YW) estimation, conditional least squares (CLS) estimation, and maximum likelihood (ML) estimation. Each of these approaches has its own advantages and is suited to different types of time series models. Finally, in Section 2.3, we outline the steps involved in conducting simulation studies, along with their significance in assessing time series models.

2.1 Negative Binomial Distribution

The negative binomial distribution is one of the most prominent discrete probability distributions in probability theory and statistics, widely used in modelling the number of failures observed before achieving r successes in a sequence of independent Bernoulli trials[1]. Its relevance spans diverse fields such as epidemiology, ecology, and insurance. This section provides a detailed overview of the negative binomial distribution, including its probability mass function, properties, and applications.

2.1.1 Definition and Some Examples

A discrete random variable X is said to follow a negative binomial distribution with parameters $r > 0$ and $0 < p < 1$ if its probability mass function is given by

$$P(X = x) = \binom{x + r - 1}{r - 1} p^r (1 - p)^x, \quad x = 0, 1, 2, \dots. \tag{2.1}$$

[1]A Bernoulli trial is a random experiment with exactly two possible outcomes, "success" and "failure", in which the probability of success is the same every time the experiment is conducted.

Here, X represents the number of failures observed before achieving r successes in a sequence of independent Bernoulli trials, where each trial has a success probability of p.

Indeed equation (2.1) defines a valid probability mass function, since

$$\begin{aligned}
\sum_{x=0}^{\infty} P(X=x) &= \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} p^r (1-p)^x \\
&= p^r \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} z^x, \quad \text{where } z = 1-p \\
&= p^r (1-z)^{-r}, \quad \text{since } \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} z^x \text{ is the series expansion of } (1-z)^{-r} \\
&= p^r \Big[1-(1-p)\Big]^{-r} \\
&= p^r p^{-r} \\
&= 1.
\end{aligned}$$

Some examples of random variables that generally follow the negative binomial probability distribution (that is, they obey equation (2.1)) are as follows:

(i) The number of failed attempts before achieving a specified number of successful sales in a business setting.

(ii) The number of defective items inspected before finding a certain number of non-defective items in quality control.

(iii) The number of missed targets before hitting a specified number of bullseyes in archery.

(iv) The number of rainy days before observing a fixed number of dry days in meteorology.

(v) The number of unsuccessful customer calls before achieving a certain number of completed calls in telecommunication.

(vi) The number of failed clinical trials before achieving a specified number of successful outcomes in medical research.

(vii) The number of rejected job applications before receiving a fixed number of job offers.

(viii) The number of unsuccessful free throws before scoring a certain number of successful free throws in basketball.

(ix) The number of broken machine parts before producing a set number of flawless components in manufacturing.

(x) The number of unsuccessful fishing attempts before catching a specified number of fish.

2.1.2 Moments and Properties

We write $X \sim NB(r, p)$ to indicate that the random variable X follows a negative binomial distribution with parameters $r > 0$ and $0 < p < 1$. A key property of the negative binomial distribution is that its variance exceeds its mean, a phenomenon known as overdispersion.

Proposition 2.1.1. *If $X \sim NB(r, p)$, then the expected value and variance of X are*

$$E(X) = \frac{r(1-p)}{p} \quad \textit{and} \quad var(X) = \frac{r(1-p)}{p^2},$$

respectively. Thus, the variance of a negative binomial random variable exceeds its mean.

Proof. First we derive the expression for the mean as follows:

$$\begin{aligned}
E(X) &= \sum_{x=1}^{\infty} xP(X = x) \\
&= \sum_{x=1}^{\infty} x\binom{x+r-1}{r-1} p^r(1-p)^x, \quad \text{by equation (2.1)} \\
&= \sum_{x=1}^{\infty} x\frac{(x+r-1)!}{x!(r-1)!} p^r(1-p)^x \\
&= \sum_{x=1}^{\infty} \frac{r(x+r-1)!}{(x-1)!r!} p^r(1-p)^x \\
&= r\sum_{x=1}^{\infty} \binom{x+r-1}{r} p^r(1-p)^x \\
&= r\sum_{x=1}^{\infty} \binom{x+r-1}{x-1} p^r(1-p)^x
\end{aligned}$$

Re-indexing the summation by letting $y = x - 1$ and $q = r + 1$ gives

$$\begin{aligned}
E(X) &= r\sum_{y=0}^{\infty} \binom{y+q-1}{y} p^{q-1}(1-p)^{y+1} \\
&= \frac{r(1-p)}{p}\sum_{y=0}^{\infty} \binom{y+q-1}{y} p^{q}(1-p)^{y} \\
&= \frac{r(1-p)}{p}, \quad \text{since} \quad \sum_{y=0}^{\infty} \binom{y+q-1}{y} p^{q}(1-p)^{y} = 1,
\end{aligned}$$

which is the required expression for the expected value of X.

Next, we prove the expression for the variance. In order to do this, we first derive an expression for $E(X^2)$. Since we can write $X^2 = X(X-1) + X$, we have

$$E(X^2) = E[X(X-1)] + E(X), \tag{2.2}$$

where

$$\begin{aligned}
E[X(X-1)] &= \sum_{x=0}^{\infty} x(x-1)P(X=x) \\
&= \sum_{x=0}^{\infty} x(x-1)\binom{x+r-1}{r-1}p^r(1-p)^x, \quad \text{by equation (2.1)} \\
&= \sum_{x=0}^{\infty} x(x-1)\frac{(x+r-1)!}{x!(r-1)!}\,p^r(1-p)^x \\
&= \sum_{x=0}^{\infty} x(x-1)\frac{(x+r-1)!(r+1)r}{x(x-1)(x-2)!(r+1)r(r-1)!}\,p^r(1-p)^x \\
&= (r+1)r\sum_{x=0}^{\infty} \frac{(x+r-1)!}{(x-2)!(r+1)!}p^r(1-p)^x \\
&= (r+1)r\sum_{x=0}^{\infty} \binom{x+r-1}{r+1}p^r(1-p)^x \\
&= (r+1)r\sum_{x=0}^{\infty} \binom{x+r-1}{x-2}p^r(1-p)^x
\end{aligned}$$

Re-indexing the summation by letting $y = x-2$ and $q = r+2$ gives

$$\begin{aligned}
E[X(X-1)] &= r(r+1)\sum_{y=0}^{\infty} \binom{y+q-1}{y}p^{q-2}(1-p)^{y+2} \\
&= r(r+1)\frac{(1-p)^2}{p^2}\sum_{y=0}^{\infty} \binom{y+q-1}{y}p^q(1-p)^y \\
&= r(r+1)\frac{(1-p)^2}{p^2}, \quad \text{since} \quad \sum_{y=0}^{\infty} \binom{y+q-1}{y}p_t^q(1-p)^y = 1
\end{aligned}$$

Now, using the method of computing variance, we have

$$\begin{aligned}
\operatorname{var}(X) &= E\big(X^2\big) - [E(X)]^2 \\
&= E[X(X-1)] + E(X) - [E(X)]^2, \quad \text{by equation (2.2)} \\
&= r(r+1)\frac{(1-p)^2}{p^2} + \frac{r(1-p)}{p} - \frac{r^2(1-p)^2}{p^2} \\
&= \frac{r(1-p)}{p}\left[\frac{(r+1)(1-p)+p-r(1-p)}{p}\right] \\
&= \frac{r(1-p)}{p}\left[\frac{r(1-p)+1-p+p-r(1-p)}{p}\right] \\
&= \frac{r(1-p)}{p}\left(\frac{1}{p}\right) \\
&= \frac{r(1-p)}{p^2},
\end{aligned}$$

which is the required expression for the variance. □

Another important characteristic of a negative binomial random variable X is its moment generating function, $M_X(t)$, defined for all real values of t by

$$M_X(t) = E(e^{tX}).$$

Proposition 2.1.2. *If $X \sim NB(r, p)$, then the moment generating function of X is*

$$M_X(t) = \left(\frac{p}{1 - (1-p)e^t} \right)^r.$$

Proof. By the definition of the moment generating function,

$$\begin{aligned}
M_X(t) &= E(e^{tX}) \\
&= \sum_{x=0}^{\infty} e^{tx} P(X = x) \\
&= \sum_{x=0}^{\infty} e^{tx} \binom{x+r-1}{r-1} p^r (1-p)^x, \quad \text{by equation (2.1)} \\
&= p^r \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} (1-p)^x e^{tx} \\
&= p^r \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} \Big((1-p)e^t \Big)^x \\
&= p^r \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} z^x, \quad \text{where } z = (1-p)e^t \\
&= p^r (1-z)^{-r}, \quad \text{since } \sum_{x=0}^{\infty} \binom{x+r-1}{r-1} z^x \text{ is the series expansion of } (1-z)^{-r} \\
&= p^r \Big(1 - (1-p)e^t \Big)^{-r} \\
&= \left(\frac{p}{1 - (1-p)e^t} \right)^r,
\end{aligned}$$

which is the required expression for the moment generating function of X. □

Remark 2.1.3. *The expected value and variance of a negative binomial random variable can also be derived from its moment generating function.*

The negative binomial distribution possesses several interesting and useful properties. One such property relates to the sum of independent negative binomial random variables. In the following, we present two propositions that demonstrate this property. We shall also present a corollary which is a direct consequence of these two propositions, specifically considering the case where the individual random variables are not only independent but also identically distributed.

Proposition 2.1.4. *If $X_1 \sim NB(r_1, p)$ and $X_2 \sim NB(r_2, p)$ are independent, then*

$$X_1 + X_2 \sim NB(r_1 + r_2, p).$$

Proof. By the definition of the moment generating function,

$$\begin{aligned}
M_{X_1+X_2}(t) &= E\big(e^{t(X_1+X_2)}\big) \\
&= E\big(e^{tX_1}e^{tX_2}\big) \\
&= E\big(e^{tX_1}\big)E\big(e^{tX_2}\big), \quad \text{since } X_1 \text{ and } X_2 \text{ are independent} \\
&= M_{X_1}(t)M_{X_2}(t) \\
&= \left(\frac{p}{1-(1-p)e^t}\right)^{r_1}\left(\frac{p}{1-(1-p)e^t}\right)^{r_2}, \quad \text{by Proposition 2.1.2} \\
&= \left(\frac{p}{1-(1-p)e^t}\right)^{r_1+r_2}.
\end{aligned}$$

This is the moment generating function of a negative binomial distribution with parameters $r_1 + r_2$ and p. Since the moment generating function uniquely determines the distribution of a random variable, it follows that $X_1 + X_2 \sim NB(r_1 + r_2, p)$. □

This result can be extended to n independent negative binomial random variables.

Proposition 2.1.5. *If $X_1, X_2, \ldots, X_n$ are independent random variables such that $X_i \sim NB(r_i, p)$ for $i = 1, 2, \ldots, n$, then*

$$\sum_{i=1}^{n} X_i \sim NB\left(\sum_{i=1}^{n} r_i, p\right).$$

Proof. Let $S_n = \sum_{i=1}^{n} X_i$. By the definition of the moment generating function,

$$\begin{aligned}
M_{S_n}(t) = E(e^{tS_n}) = E\left(e^{t\sum_{i=1}^{n} X_i}\right) &= E\left(\prod_{i=1}^{n} e^{tX_i}\right) \\
&= \prod_{i=1}^{n} E\big(e^{tX_i}\big), \quad \text{since } X_1, X_2, \ldots, X_n \text{ are independent} \\
&= \prod_{i=1}^{n} M_{X_i}(t) \\
&= \prod_{i=1}^{n}\left(\frac{p}{1-(1-p)e^t}\right)^{r_i}, \quad \text{by Proposition 2.1.2} \\
&= \left(\frac{p}{1-(1-p)e^t}\right)^{\sum_{i=1}^{n} r_i}.
\end{aligned}$$

This is the moment generating function of a negative binomial distribution with parameters

$\sum_{i=1}^{n} r_i$ and p. Since the moment generating function uniquely determines the distribution of a random variable, it follows that

$$\sum_{i=1}^{n} X_i \sim NB\left(\sum_{i=1}^{n} r_i, p\right).$$

□

Corollary 2.1.6. *If $X_1, X_2, \ldots, X_n$ are independent and identically distributed (i.i.d.) random variables, each following a $NB(r, p)$ distribution, then*

$$\sum_{i=1}^{n} X_i \sim NB(nr, p).$$

Proof. It follows directly from the Proposition 2.1.5 that

$$\sum_{i=1}^{n} X_i \sim NB\left(\sum_{i=1}^{n} r, p\right) = NB(nr, p),$$

as required. This completes the proof of the corollary. □

2.1.3 Some Applications of the Negative Binomial Distribution

The negative binomial distribution is widely used in various fields to model the number of failures required to achieve a predetermined number of successes in a sequence of independent Bernoulli trials. It is particularly valuable when the data exhibit overdispersion, where the variance exceeds the mean. In this subsection, we present examples of its applications across different domains, demonstrating the flexibility and utility of the negative binomial distribution in modelling real-world phenomena across diverse disciplines.

1. Customer Arrivals in a Queue

In a queueing system, the negative binomial distribution can model the number of failures (unsuccessful customer services) required until a specific number of successful services (e.g., processed customers) is achieved. For instance, if the probability of successfully serving a customer in a single interval is $p = 0.2$ and we are interested in the probability of requiring 10 arrivals (failures + successes) to successfully serve 3 customers, we can calculate this probability using equation (2.1) as follows:

$$P(X = 7) = \binom{9}{2}(0.2)^3(0.8)^7 \approx 0.302.$$

Here, X is the number of failures, and $r = 3$ is the target number of successful services.

2. Epidemiology

In public health, the negative binomial distribution is used to model the number of days required to observe a certain number of new cases during a disease outbreak. For instance, during a flu epidemic, it can predict the number of days until a specific number of hospital admissions occur, accounting for the variability in daily case numbers.

If the probability of a daily admission is $p = 0.3$, and we want to find the probability of requiring 7 days (failures + successes) to observe 3 hospital admissions (successes), we can calculate this probability using equation (2.1) as follows:

$$P(X = 4) = \binom{6}{2}(0.3)^3(0.7)^4 \approx 0.0595.$$

Here, X is the number of failures, and $r = 3$ is the target number of successful admissions.

3. Quality Control

Manufacturing industries use this distribution to determine the number of items that need to be inspected before finding a specific number of defective products. For example, in a factory producing lightbulbs, it may predict the number of bulbs tested until five defective ones are identified.

Assuming the probability of finding a defective bulb is $p = 0.05$, we want to find the probability that 12 items (failures + successes) must be inspected to identify 3 defective products (successes). We can calculate this probability using equation (2.1) as follows:

$$P(X = 9) = \binom{11}{2}(0.05)^3(0.95)^9 \approx 0.0084.$$

Here, X is the number of failures, and $r = 3$ is the target number of defective products.

4. Insurance and Risk Analysis

The negative dinomial distribution helps insurers estimate the number of claims filed before reaching a total payout threshold.

For instance, if the probability of a claim exceeding a certain payout is $p = 0.1$, we can calculate the probability of requiring 9 claims (failures + successes) to reach 2 significant payouts (successes) using equation (2.1) as follows:

$$P(X = 7) = \binom{8}{1}(0.1)^2(0.9)^7 \approx 0.0477.$$

Here, X denotes the number of failures, and $r = 2$ is the target number of significant payouts.

5. Marketing and Sales

In marketing, the negative binomial distribution is applied to predict the number of customer interactions required to secure a set number of sales. For example, if a sales team has a 25% success rate ($p = 0.25$), the probability of making 8 interactions (failures + successes) to close 3 sales (successes) can be calculated using equation (2.1) as follows:

$$P(X = 5) = \binom{7}{2}(0.25)^3(0.75)^5 \approx 0.1055.$$

Here, X is the number of failures, and $r = 3$ is the target number of sales.

6. Ecology and Conservation

Ecologists use the negative binomial distribution to model the number of individual observations required to count a predetermined number of rare species in a given area.

For instance, if the probability of observing a rare species in a given attempt is $p = 0.1$, the probability of needing to make 10 observations (failures + successes) to observe 4 rare species (successes) can be calculated using equation (2.1) as follows:

$$P(X = 6) = \binom{9}{3}(0.1)^4(0.9)^6 \approx 0.0291.$$

Here, X denotes the number of failures, and $r = 4$ is the target number of rare species.

7. Sports Analytics

The negative binomial distribution is used to analyse the number of attempts a player needs to achieve a certain number of successful actions, such as goals or free throws.

For example, if a player has a 70% success rate in making a free throw ($p = 0.7$), the probability of needing to take 9 shots (failures + successes) to score 5 successful free throws (successes) can be calculated using equation (2.1) as follows:

$$P(X = 4) = \binom{8}{4}(0.7)^5(0.3)^4 \approx 0.1980.$$

Here, X is the number of failures, and $r = 5$ is the target number of successful free throws.

8. Finance and Economics

In financial modelling, the negative binomial distribution can describe the number of losing trades required before achieving a fixed profit target.

For instance, if a trader has a 40% probability of making a profitable trade ($p = 0.4$), the probability of making 6 trades (failures + successes) before achieving 4 profitable trades (successes) can be calculated using equation (2.1) as follows:

$$P(X=2)=\binom{5}{3}(0.4)^4(0.6)^2\approx 0.0576.$$

Here, X denotes the number of failures, and $r=4$ is the target number of profitable trades.

9. Reliability Engineering

Engineers use the negative binomial distribution to predict the number of operational cycles before observing a set number of mechanical faults in a system.

For example, suppose the probability of a fault in a single cycle is $p=0.02$. The probability of needing 22 cycles (failures + successes) before detecting 3 faults (successes) can be calculated using equation (2.1) as follows:

$$P(X=19)=\binom{21}{2}(0.02)^3(0.98)^{19}\approx 0.0006.$$

Here, X is the number of failures, and $r=3$ is the target number of mechanical faults.

10. Traffic Accidents

The negative binomial distribution can model the number of weeks with no accidents (failures) required until a fixed number of accidents (successes) occur at a specific intersection. For example, if the probability of an accident in a given week is $p=0.3$, and we want the probability of requiring 5 weeks to observe 2 accidents, we can use equation (2.1) to compute:

$$P(X=3)=\binom{4}{1}(0.3)^2(1-0.3)^3=\binom{4}{1}(0.3)^2(0.7)^3\approx 0.185.$$

Here, X is the number of failures, and $r=2$ is the target number of accidents.

2.2 Parameter Estimation in Time Series Analysis

In this section, we discuss three commonly used approaches for estimating parameters in time series analysis: Yule–Walker (YW) estimation, conditional least squares (CLS) estimation, and maximum likelihood (ML) estimation. Each of these approaches has its own advantages and is suited to different types of time series models.

2.2.1 Yule–Walker Estimation

The Yule–Walker equations provide a straightforward approach to estimating the parameters of autoregressive (AR) models. This method uses the autocovariance function of the time series to derive a system of linear equations that relate the model's parameters to the observed data. By solving these equations, parameter estimates can be efficiently obtained.

Consider an autoregressive model of order p, abbreviated as AR(p) and defined as:

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \cdots + a_p X_{t-p} + \varepsilon_t, \tag{2.3}$$

where ε_t is a white noise process with mean zero and constant variance, and $a_1, a_2, \ldots, a_p$ are the parameters to be estimated. The Yule–Walker equations are expressed as:

$$\gamma_X(k) = \sum_{j=1}^{p} a_j \gamma_X(k - j), \quad k = 1, 2, \ldots, p, \tag{2.4}$$

where $\gamma_X(k)$ denotes the autocovariance of the AR(p) process X_t at lag k. To estimate the parameters $a_1, a_2, \ldots, a_p$, the autocovariances $\gamma_X(k)$ are computed from the observed data, and the resulting system of equations is solved.

This method is computationally efficient and is particularly useful for stationary time series data, as it assumes the statistical properties of the series do not change over time. However, the Yule–Walker approach has some limitations. It is restricted to pure AR models and does not perform well for models containing moving average (MA) components or for non-stationary time series. In such cases, alternative methods like conditional least squares or maximum likelihood estimation may be more appropriate.

2.2.2 Conditional Least Squares Estimation

Conditional least squares (CLS) estimation is a widely used approach for estimating parameters in time series models, particularly in autoregressive integrated moving average (ARIMA) models. CLS estimation minimizes the sum of squared differences between the observed values and the predicted values, conditioned on past observations.

Consider a first-order autoregressive (AR(1)) model defined by

$$X_t = \alpha X_{t-1} + \varepsilon_t, \tag{2.5}$$

where ε_t is a white noise process with mean zero and constant variance, and α is the parameter to be estimated. The CLS estimate of α is obtained by minimizing:

$$S(\alpha) = \sum_{t=2}^{N} (x_t - \alpha x_{t-1})^2. \tag{2.6}$$

CLS estimation has several advantages, including computational simplicity compared to maximum likelihood estimation (MLE) and greater robustness to certain violations of model assumptions. However, it can be sensitive to the choice of initial conditions and may not perform well with non-stationary data or data with heavy-tailed residuals.

2.2.3 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) is a powerful and flexible method for estimating parameters in a wide variety of statistical models, including time series models. The MLE approach involves constructing a log-likelihood function based on the assumed probability distribution of the observed data. By maximizing this log-likelihood function with respect to the model parameters, we obtain the parameter estimates that are most consistent with the observed data. Consider a first-order autoregressive (AR(1)) model defined by

$$X_t = \alpha X_{t-1} + \varepsilon_t, \tag{2.7}$$

where $\varepsilon_t \sim N(0, \sigma^2)$ is a white noise process, and α and σ^2 are the parameters to be estimated. The MLE estimates of α and σ^2 are obtained by maximizing the log-likelihood function:

$$l(\alpha, \sigma^2) = \sum_{t=2}^{N} \left(-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(x_t - \alpha x_{t-1})^2}{2\sigma^2} \right).$$

To derive this log-likelihood function, we first observe that the AR(1) model is given by

$$x_t = \alpha x_{t-1} + \varepsilon_t,$$

where $\varepsilon_t \sim N(0, \sigma^2)$, thus the conditional distribution of x_t given x_{t-1} is:

$$x_t | x_{t-1} \sim N(\alpha x_{t-1}, \sigma^2).$$

The log-likelihood function for the observed data $x_2, \ldots, x_N$ can then be derived as follows:

$$\begin{aligned}
l(\alpha, \sigma^2) = \log \prod_{t=2}^{N} f(x_t | x_{t-1}) &= \sum_{t=2}^{N} \log f(x_t | x_{t-1}) \\
&= \sum_{t=2}^{N} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_t - \alpha x_{t-1})^2}{2\sigma^2} \right) \right) \\
&= \sum_{t=2}^{N} \left(\log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{(x_t - \alpha x_{t-1})^2}{2\sigma^2} \right) \\
&= \sum_{t=2}^{N} \left(-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(x_t - \alpha x_{t-1})^2}{2\sigma^2} \right)
\end{aligned}$$

MLE is known for its asymptotic efficiency and normality, offering precise estimates for well-specified models. Its advantages include statistical efficiency and asymptotic unbiasedness, making it suitable for complex and non-stationary models. However, MLE also has some limitations: it can be computationally intensive, especially for large datasets, and it requires correct specification of the error distribution.

2.3 Simulation Studies in Time Series Analysis

Simulation studies are fundamental in time series analysis, providing a systematic approach to evaluate the performance of statistical models under controlled conditions. By generating artificial data based on specified models and parameters, researchers can rigorously test the effectiveness of various estimation methods. This section outlines the steps involved in conducting simulation studies, along with their significance in assessing time series models.

2.3.1 Steps Involved in Conducting Simulation Studies

The process of conducting simulation studies in time series analysis typically involves the following steps:

Step 1: Define the Objective

Clearly outline the objectives of the simulation study. This includes identifying the specific estimation methods to be evaluated (e.g., maximum likelihood estimation (MLE), conditional least squares estimation (CLSE) and the time series models of interest.

Step 2: Specify the Model and Parameters

Select the time series model to be simulated and specify the parameters. For example, if using an integer-valued autoregressive (INAR) model, define the appropriate parameter values (e.g., autoregressive coefficients) and the distribution of the error term.

Step 3: Generate Artificial Data

Utilize the defined model and parameters to generate artificial time series data. This can be done using statistical software or programming languages such as R or Python. It is essential to create multiple datasets to ensure the robustness of the results.

Step 4: Apply Estimation Methods

For each generated dataset, apply the selected estimation methods (e.g., MLE, CLSE) to estimate the model parameters. Record the estimated values for further analysis.

Step 5: Evaluate Performance Metrics

Assess the performance of the estimation methods by calculating relevant performance metrics. Common metrics include mean absolute error (MAE), mean squared error (MSE), mean absolute deviation error (MADE), or root mean squared error (RMSE). Since the true underlying model is known, it is possible to directly compare the estimated results with the true parameters or assess forecasting accuracy.

Step 6: Conduct Monte Carlo Experiments

If necessary, perform Monte Carlo experiments to examine how estimation methods behave under varying conditions, such as different sample sizes or noise levels. This step helps to understand the robustness of the methods across different scenarios.

Step 7: Analyze Results

Analyze the results obtained from the simulations. This includes comparing the performance of different estimation methods and interpreting how changes in model parameters impact forecasting accuracy.

Step 8: Report Findings

Finally, compile and report the findings of the simulation studies. Include visual aids such as graphs and tables to effectively communicate the results. Discuss the implications of the findings for practitioners in the field of time series analysis.

2.3.2 Significance of Simulation Studies

Simulation studies provide valuable insights into the effectiveness of estimation methods in time series analysis. For instance, they can help assess the performance of MLE and CLSE when fitting an INAR model. By systematically varying conditions, such as sample size or noise levels, researchers can examine the behavior of estimators like the Yule–Walker estimators. These studies allow researchers to quantify uncertainties, explore the impact of parameter changes on forecasting accuracy, and ultimately enhance the robustness and reliability of time series models. By providing a deeper understanding of the dynamics at play, simulation studies contribute significantly to the advancement of time series analysis methodologies.

Chapter 3

Model Structure and Approaches for Parameter Estimation

In this chapter, we introduce the negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q, denoted as NBINGARCH(p, q), and outline an approach for parameter estimation. Section 3.1 formally defines the model, detailing its key assumptions, mathematical formulation, and fundamental properties. We also derive the first- and second-order stationarity conditions and present equations to compute the autocovariance and autocorrelation functions. Section 3.2 provides a detailed example for the specific case of $p = 1, q = 1$. Finally, in Section 3.3, we discuss the Yule–Walker (YW) and conditional least squares (CLS) estimation methods for specific cases of the NBINGARCH model, followed by the development of the maximum likelihood (ML) estimation approach.

3.1 Structure of the Model

To rigorously address the challenges of modeling overdispersed count time series data, this dissertation introduces an innovative application of the Negative Binomial Integer-Valued Generalized Autoregressive Conditional Heteroscedastic (NBINGARCH) model. The following key terms are used throughout this dissertation:

(i) **Sigma-field (or Sigma-algebra):** The σ-field, denoted $\mathcal{F}_{t-1}$, represents all available information from the past history of the time series up to time $t-1$. It is generated by the past values of the time series, $\{X_{t-1}, X_{t-2}, \ldots\}$. This means that, given $\mathcal{F}_{t-1}$, all relevant information for predicting future values is contained in the past observations.

(ii) **NBINGARCH(p, q):** A model where the conditional distribution of the count at time t follows a negative binomial distribution. The conditional mean depends on the past p values of the time series and the past q values of a conditional mean process, allowing for dynamic dependence.

(iii) **Heteroscedasticity:** A condition in which the variance of a process changes over time rather than remaining constant.

(iv) **Generalized Autoregressive Conditional Heteroscedasticity (GARCH):** A framework used to model time-varying conditional variance, where variance depends on past squared errors and past conditional variances. While classical GARCH models apply to continuous-valued data (e.g., financial returns), NBINGARCH extends this concept to integer-valued count data.

Let $\{X_t\}$ be a time series of counts and let $\mathcal{F}_{t-1}$ be the σ-field generated by $\{X_{t-1}, X_{t-2}, \ldots\}$. We assume that, conditional on $\mathcal{F}_{t-1}$, the random variables $X_1, \ldots, X_n$ are independent and the conditional distribution of X_t follows a negative binomial (NB) distribution. More precisely, we define the model as follows:

Definition 3.1.1. *Let $\{X_t\}$ be a time series of counts and let $\mathcal{F}_{t-1}$ be the σ-field generated by $\{X_{t-1}, X_{t-2}, \ldots\}$. The process X_t is called a negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q, denoted NBINGARCH(p, q), if the conditional distribution of X_t follows a negative binomial distribution:*

$$X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \tag{3.1}$$

where r is a positive number, and p_t is the conditional probability parameter of the negative binomial distribution at time t, given $\mathcal{F}_{t-1}$, representing the probability of success in a sequence of Bernoulli trials leading to the rth failure. The conditional mean λ_t and the parameter p_t satisfy the relation:

$$\frac{1-p_t}{p_t} = \lambda_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j}, \tag{3.2}$$

where λ_t represents the conditional mean of the negative binomial distribution at time t, conditional on the information set $\mathcal{F}_{t-1}$, and is interpreted as the expected number of counts at time t, given the past values of the time series, with $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, $i = 1, \ldots, p$, $j = 1, \ldots, q$, $p \geq 1$, $q \geq 0$.

Remark 3.1.2. *If $r = 1$, then the NB distribution becomes the geometric distribution, and in this case, the NBINGARCH model can be referred to as the geometric INGARCH model.*

We now derive the conditional probability mass function of the NBINGARCH process X_t. This mass function will be used to derive the conditional mean and variance of X_t.

Proposition 3.1.3. *Let X_t be an NBINGARCH(p, q) process. The conditional probability mass function of X_t is given by*

$$P(X_t = x_t|\mathcal{F}_{t-1}) = \binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t}, \qquad x_t = 0, 1, 2, \ldots,$$

where $p_t = 1/(1 + \lambda_t)$ and $q_t = 1 - p_t = \lambda_t/(1 + \lambda_t)$.

Proof. From equation (3.1) of Definition 3.1.1, we have

$$X_t|\mathcal{F}_{t-1} \sim NB(r, p_t).$$

Hence, the conditional probability mass function of X_t is

$$P(X_t = x_t|\mathcal{F}_{t-1}) = \binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t}, \qquad x_t = 0, 1, 2, \ldots.$$

Next, from equation (3.2) of Definition 3.1.1, we have

$$\lambda_t = \frac{1 - p_t}{p_t}.$$

This can be rearranged to solve for p_t as follows:

$$\begin{aligned} \lambda_t p_t &= 1 - p_t \\ \lambda_t p_t + p_t &= 1 \\ (\lambda_t + 1) p_t &= 1 \\ p_t &= \frac{1}{\lambda_t + 1} \end{aligned}$$

so that

$$q_t = 1 - p_t = 1 - \frac{1}{\lambda_t + 1} = \frac{\lambda_t}{1 + \lambda_t}.$$

This confirms the expressions for p_t and q_t, as required. □

Having defined the probability mass function of the NBINGARCH(p, q) process, it is essential to explore its key statistical properties. Specifically, understanding the conditional mean and variance offers insight into how the model captures overdispersion in the data. The following proposition provides explicit expressions for these conditional moments.

Proposition 3.1.4. *Let X_t be an NBINGARCH(p, q) process. The conditional mean and variance of X_t are respectively given by*

$$E(X_t|\mathcal{F}_{t-1}) = \frac{r(1 - p_t)}{p_t} = r\lambda_t, \tag{3.3}$$

$$var(X_t|\mathcal{F}_{t-1}) = \frac{r(1 - p_t)}{p_t^2} = r\lambda_t(1 + \lambda_t). \tag{3.4}$$

Proof. First we derive the expression for the conditional mean as follows:

$$
\begin{aligned}
E(X_t \mid \mathcal{F}_{t-1}) &= \sum_{x_t=1}^{\infty} x_t P(X_t = x_t|\mathcal{F}_{t-1}) \\
&= \sum_{x_t=1}^{\infty} x_t \binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t} \\
&= \sum_{x_t=1}^{\infty} x_t \frac{(x_t + r - 1)!}{x_t!(r - 1)!} p_t^r (1 - p_t)^{x_t} \\
&= \sum_{x_t=1}^{\infty} \frac{r(x_t + r - 1)!}{(x_t - 1)!r!} p_t^r (1 - p_t)^{x_t} \\
&= r \sum_{x_t=1}^{\infty} \binom{x_t + r - 1}{r} p_t^r (1 - p_t)^{x_t} \\
&= r \sum_{x_t=1}^{\infty} \binom{x_t + r - 1}{x_t - 1} p_t^r (1 - p_t)^{x_t}
\end{aligned}
$$

Re-indexing the summation by letting $y_t = x_t - 1$ and $q_t = r + 1$ gives

$$
\begin{aligned}
E(X_t \mid \mathcal{F}_{t-1}) &= r \sum_{y_t=0}^{\infty} \binom{y_t + q_t - 1}{y_t} p_t^{q_t - 1} (1 - p_t)^{y_t + 1} \\
&= \frac{r(1 - p_t)}{p_t} \sum_{y_t=0}^{\infty} \binom{y_t + q_t - 1}{y_t} p_t^{q_t} (1 - p_t)^{y_t} \\
&= \frac{r(1 - p_t)}{p_t}, \quad \text{since} \quad \sum_{y_t=0}^{\infty} \binom{y_t + q_t - 1}{y_t} p_t^{q_t} (1 - p_t)^{y_t} = 1 \\
&= r\lambda_t,
\end{aligned}
$$

which is the required expression for the conditional mean. Next, we prove the expression for the conditional variance. In order to do this, we first derive an expression for $E(X_t^2 \mid \mathcal{F}_{t-1})$. Since we can write $X^2 = X(X - 1) + X$, we have

$$
E(X_t^2 \mid \mathcal{F}_{t-1}) = E[X_t(X_t - 1) \mid \mathcal{F}_{t-1}] + E(X_t \mid \mathcal{F}_{t-1}),
$$

where

$$\begin{aligned}
E[X_t(X_t-1) \mid \mathcal{F}_{t-1}] &= \sum_{x_t=0}^{\infty} x_t(x_t-1)\binom{x_t+r-1}{r-1} p_t^r(1-p_t)^{x_t} \\
&= \sum_{x_t=0}^{\infty} x_t(x_t-1)\frac{(x_t+r-1)!}{x_t!(r-1)!} p_t^r(1-p_t)^{x_t} \\
&= \sum_{x_t=0}^{\infty} x_t(x_t-1)\frac{(x_t+r-1)!(r+1)r}{x_t(x_t-1)(x_t-2)!(r+1)r(r-1)!} p_t^r(1-p_t)^{x_t} \\
&= (r+1)r\sum_{x_t=0}^{\infty} \frac{(x_t+r-1)!}{(x_t-2)!(r+1)!} p_t^r(1-p_t)^{x_t} \\
&= (r+1)r\sum_{x_t=0}^{\infty} \binom{x_t+r-1}{r+1} p_t^r(1-p_t)^{x_t} \\
&= (r+1)r\sum_{x_t=0}^{\infty} \binom{x_t+r-1}{x_t-2} p_t^r(1-p_t)^{x_t}
\end{aligned}$$

Re-indexing the summation by letting $y_t = x_t - 2$ and $q_t = r+2$ gives

$$\begin{aligned}
E[X_t(X_t-1) \mid \mathcal{F}_{t-1}] &= r(r+1)\sum_{y_t=0}^{\infty} \binom{y_t+q_t-1}{y_t} p_t^{q_t-2}(1-p_t)^{y_t+2} \\
&= r(r+1)\frac{(1-p_t)^2}{p_t^2}\sum_{y=0}^{\infty} \binom{y_t+q_t-1}{y_t} p_t^{q_t}(1-p_t)^{y_t} \\
&= r(r+1)\frac{(1-p_t)^2}{p_t^2}, \quad \text{since} \quad \sum_{y=0}^{\infty} \binom{y_t+q_t-1}{y_t} p_t^{q_t}(1-p_t)^{y_t} = 1
\end{aligned}$$

Now, using the method for the conditional variance, we have

$$\begin{aligned}
\operatorname{var}(X_t \mid \mathcal{F}_{t-1}) &= E\left(X_t^2 \mid \mathcal{F}_{t-1}\right) - [E(X_t \mid \mathcal{F}_{t-1})]^2 \\
&= E[X_t(X_t-1) \mid \mathcal{F}_{t-1}] + E[X_t \mid \mathcal{F}_{t-1}] - [E(X_t \mid \mathcal{F}_{t-1})]^2 \\
&= r(r+1)\frac{(1-p_t)^2}{p_t^2} + \frac{r(1-p_t)}{p_t} - \frac{r^2(1-p_t)^2}{p_t^2} \\
&= \frac{r(1-p_t)}{p_t}\left[\frac{(r+1)(1-p_t)+p_t-r(1-p_t)}{p_t}\right] \\
&= \frac{r(1-p_t)}{p_t}\left[\frac{r(1-p_t)+1-p_t+p_t-r(1-p_t)}{p_t}\right] \\
&= \frac{r(1-p_t)}{p_t}\left[\frac{1}{p_t}\right] \\
&= \frac{r(1-p_t)}{p_t^2} \\
&= r\lambda_t(1+\lambda_t),
\end{aligned}$$

which is the required expression for the conditional variance. □

The conditional variance in Proposition 3.1.4 includes an additional λ_t term, suggesting that the variability of the process can exceed its mean. This characteristic is indicative of overdispersion, a common feature in count time series data. The following corollary formalises this property by establishing that both the conditional and unconditional variances of the NBINGARCH(p, q) process are strictly greater than their corresponding means.

Corollary 3.1.5. *Let X_t be an NBINGARCH(p, q) process. The following inequalities hold:*

$$var(X_t|\mathcal{F}_{t-1}) > E(X_t|\mathcal{F}_{t-1}), \tag{3.5}$$

$$var(X_t) > E(X_t). \tag{3.6}$$

Proof. It follows from Proposition 3.1.4 that

$$\begin{aligned} E(X_t|\mathcal{F}_{t-1}) &= r\lambda_t \\ &< r\lambda_t(1+\lambda_t), \quad \text{since } \lambda_t > 0 \\ &= \text{var}(X_t|\mathcal{F}_{t-1}). \end{aligned}$$

Furthermore, by the law of total variance (conditional variance formula), we have

$$\begin{aligned} \text{var}(X_t) &= E\Big(\text{var}(X_t|\mathcal{F}_{t-1})\Big) + \text{var}\Big(E(X_t|\mathcal{F}_{t-1})\Big) \\ &= E\Big(r\lambda_t(1+\lambda_t)\Big) + \text{var}(r\lambda_t) \\ &= rE(\lambda_t) + rE(\lambda_t)^2 + r^2\text{var}(\lambda_t) \\ &= rE(\lambda_t) + r(E(\lambda_t))^2 + r\text{Var}(\lambda_t) + r^2\text{var}(\lambda_t) \\ &= rE(\lambda_t) + r(E(\lambda_t))^2 + (r+r^2)\text{var}(\lambda_t) \qquad (3.7) \\ &> rE(\lambda_t) = E(X_t), \end{aligned}$$

as required. □

Remark 3.1.6. *Corollary 3.1.5 shows that the NBINGARCH(p, q) process X_t, as defined in Definition 3.1.1, effectively accounts for overdispersion in count time series by establishing that the conditional and unconditional variances of X_t consistently exceed their corresponding means. This property is critical in modeling count data, where the assumption of equal mean and variance in models like the Poisson distribution is often violated due to the presence of excess variability. The inequality $var(X_t|\mathcal{F}_{t-1}) > E(X_t|\mathcal{F}_{t-1})$ highlights how the model dynamically adjusts for overdispersion based on past information, capturing both serial dependence and clustering in the data. Similarly, the unconditional inequality $var(X_t) > E(X_t)$ confirms the model's ability to represent persistent overdispersion across the entire series. This flexibility makes the NBINGARCH(p, q) process particularly well-suited for real-world applications where count data exhibit high variability, such as epidemiological case counts, insurance claims, and financial transaction counts.*

To simplify notation, we assume $p \geq q$ and proceed to establish the first- and second-order stationarity conditions for the NBINGARCH model.

Proposition 3.1.7. *Let X_t be an NBINGARCH(p, q) process. A necessary and sufficient condition for X_t to be stationary in the mean is that all roots of the equation*

$$1 - \sum_{i=1}^{q}(r\alpha_i + \beta_i)z^i - \sum_{i=q+1}^{p} r\alpha_i z^i = 0 \tag{3.8}$$

lie outside the unit circle, meaning their absolute values exceed 1.

Proof. Let $\mu_t = E(X_t)$ be the mean of the process at time t. Then,

$$\begin{aligned}
\mu_t &= E(X_t) \\
&= E\Big(E(X_t \mid \mathcal{F}_{t-1})\Big), \quad \text{by the law of total expectation} \\
&= E(r\lambda_t), \quad \text{by Proposition 3.1.4} \\
&= rE(\lambda_t) \\
&= rE\left(\alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j}\right), \quad \text{by equation (3.2)} \\
&= r\alpha_0 + \sum_{i=1}^{p} r\alpha_i E(X_{t-i}) + \sum_{j=1}^{q} \beta_j E(r\lambda_{t-j}) \\
&= r\alpha_0 + \sum_{i=1}^{p} r\alpha_i \mu_{t-i} + \sum_{j=1}^{q} \beta_j \mu_{t-j} \\
&= r\alpha_0 + \sum_{i=1}^{q} r\alpha_i \mu_{t-i} + \sum_{i=q+1}^{p} r\alpha_i \mu_{t-i} + \sum_{j=1}^{q} \beta_j \mu_{t-j}, \quad \text{since } p \geq q \\
&= r\alpha_0 + \sum_{i=1}^{q} (r\alpha_i + \beta_i)\mu_{t-i} + \sum_{i=q+1}^{p} r\alpha_i \mu_{t-i}.
\end{aligned}$$

Moving all terms involving μ_t to the left side of the equation gives

$$\mu_t - \sum_{i=1}^{q} (r\alpha_i + \beta_i)\mu_{t-i} - \sum_{i=q+1}^{p} r\alpha_i \mu_{t-i} = r\alpha_0 \tag{3.9}$$

$$\left(1 - \sum_{i=1}^{q} (r\alpha_i + \beta_i)B^i - \sum_{i=q+1}^{p} r\alpha_i B^i\right)\mu_t = r\alpha_0,$$

where B denotes the backshift operator, defined such that $B\mu_t = \mu_{t-1}$ and $B^k\mu_t = \mu_{t-k}$ for any positive integer k. It hence follows that the associated characteristic equation is

$$1 - \sum_{i=1}^{q} (r\alpha_i + \beta_i)z^i - \sum_{i=q+1}^{p} r\alpha_i z^i = 0.$$

The process X_t is stationary in the mean if and only if all roots of this equation lie outside the unit circle. This condition ensures that the contributions of past values decay geometrically, thereby stabilizing the mean over time. See Bollerslev [5] on page 311 for a detailed discussion of stationarity conditions in time series stochastic processes. □

Proposition 3.1.8. *Let X_t be an NBINGARCH(p, q) process. If X_t is first-order stationary, then its mean at any time t is given by:*

$$\mu = E(X_t) = \frac{r\alpha_0}{1 - \sum_{i=1}^{q}(r\alpha_i + \beta_i) - \sum_{i=q+1}^{p} r\alpha_i}.$$

Consequently, a necessary condition for X_t to be first-order stationary is that:

$$0 \leq \sum_{i=1}^{q}(r\alpha_i + \beta_i) + \sum_{i=q+1}^{p} r\alpha_i < 1.$$

Proof. In the proof of Proposition 3.1.7, equation (3.9) is given as:

$$\mu_t - \sum_{i=1}^{q}(r\alpha_i + \beta_i)\mu_{t-i} - \sum_{i=q+1}^{p} r\alpha_i\mu_{t-i} = r\alpha_0$$

Assuming X_t is first-order stationary implies that $\mu_t = \mu$ (a constant) for all t, so we have:

$$\mu - \sum_{i=1}^{q}(r\alpha_i + \beta_i)\mu - \sum_{i=q+1}^{p} r\alpha_i\mu = r\alpha_0$$

$$\mu\left(1 - \sum_{i=1}^{q}(r\alpha_i + \beta_i) - \sum_{i=q+1}^{p} r\alpha_i\right) = r\alpha_0$$

$$\mu = \frac{r\alpha_0}{1 - \sum_{i=1}^{q}(r\alpha_i + \beta_i) - \sum_{i=q+1}^{p} r\alpha_i},$$

which is the required expression for μ. But this expression implies that for μ to be definite and well-defined, the denominator must be positive:

$$1 - \sum_{i=1}^{q}(r\alpha_i + \beta_i) - \sum_{i=q+1}^{p} r\alpha_i > 0$$

Rewriting, this condition is equivalent to the condition:

$$0 \leq \sum_{i=1}^{q}(r\alpha_i + \beta_i) + \sum_{i=q+1}^{p} r\alpha_i < 1.$$

The first inequality follows from Definition 3.1.1, which states that $r > 0$, $\alpha_i \geq 0$ and $\beta_j \geq 0$, ensuring that each term is nonnegative. □

We now establish the second-order stationarity conditions for the NBINGARCH(p,q) model. For simplicity, we focus on a special case where $q=0$, and we denote this simplified model as NBINGARCH(p). This simplification allows for a more tractable analysis of the model.

Proposition 3.1.9. *Let X_t be a first-order stationary NBINGARCH(p) process . A necessary and sufficient condition for X_t to be second-order stationary is that all roots of*

$$1 - C_1 z - \cdots - C_p z^p = 0$$

lie outside the unit circle, where for $u,l = 1,\dots,p-1$, we have

$$C_u = (r+r^2)\left(\alpha_u^2 - \sum_{v=1}^{p-1}\sum_{|i-j|=v} \alpha_i\alpha_j b_{vu}\beta_{u0}\right), \qquad C_p = (r+r^2)\alpha_p^2,$$

$$\beta_{l0} = r\alpha_l, \qquad \beta_{ll} = r\sum_{|i-l|=l}\alpha_i - 1, \quad \text{and} \quad \beta_{lu} = r\sum_{|i-l|=u}\alpha_i, \quad u \neq l,$$

where B and B^{-1} are $(p-1)\times(p-1)$ matrices such that $B = (\beta_{ij})_{i,j=1}^{p-1}$ and $B^{-1} = (b_{ij})_{i,j=1}^{p-1}$

Proof. Let $\gamma_{it} = E(X_t X_{t-i})$ for $i = 0,1,2,\dots,p$, and let C be a constant independent of t. Assuming the process X_t is second-order stationary implies that $\gamma_{st} = \gamma_{s,t-i}$ for $i = 0,1,\dots,p$. We proceed by first examining the conditional second moment:

$$\begin{aligned} E(X_t^2|\mathcal{F}_{t-1}) &= \text{var}(X_t|\mathcal{F}_{t-1}) + [E(X_t|\mathcal{F}_{t-1})]^2 \\ &= r\lambda_t(1+\lambda_t) + (r\lambda_t)^2, \quad \text{by Proposition 3.1.4} \\ &= r\lambda_t + r\lambda_t^2 + r^2\lambda_t^2 \\ &= r\lambda_t + (r+r^2)\lambda_t^2. \end{aligned} \tag{3.10}$$

Since $q=0$ for the present case, equation (3.2) in Definition 3.1.1 becomes

$$\lambda_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}. \tag{3.11}$$

Substituting this expression of λ_t into equation (3.10) gives

$$\begin{aligned} E(X_t^2|\mathcal{F}_{t-1}) &= r\left(\alpha_0 + \sum_{i=1}^{p}\alpha_i X_{t-i}\right) + (r+r^2)\left(\alpha_0 + \sum_{i=1}^{p}\alpha_i X_{t-i}\right)^2 \\ &= r\alpha_0 + r\sum_{i=1}^{p}\alpha_i X_{t-i} + (r+r^2)\left(\alpha_0^2 + 2\alpha_0\sum_{i=1}^{p}\alpha_i X_{t-i} + \sum_{i=1}^{p}\alpha_i^2 X_{t-i}^2 + \sum_{\substack{i,j=1\\ i\neq j}}^{p}\alpha_i\alpha_j X_{t-i}X_{t-j}\right) \\ &= r\alpha_0 + (r+r^2)\alpha_0^2 + \big(r + 2(r+r^2)\alpha_0\big)\sum_{i=1}^{p}\alpha_i X_{t-i} + (r+r^2)\left(\sum_{i=1}^{p}\alpha_i^2 X_{t-i}^2 + \sum_{\substack{i,j=1\\ i\neq j}}^{p}\alpha_i\alpha_j X_{t-i}X_{t-j}\right). \end{aligned}$$

For $l = 1, \ldots, p-1$, the covariance between X_t and X_{t-l} is

$$
\begin{aligned}
\gamma_{lt} &= E(X_t X_{t-l}) \\
&= E\Big(E(X_t|\mathcal{F}_{t-1})X_{t-l}\Big), \quad \text{by the law of total expectation} \\
&= E(r\lambda_t X_{t-l}), \quad \text{by Proposition 3.1.4} \\
&= E\left(r\left(\alpha_0 + \sum_{i=1}^{p}\alpha_i X_{t-i}\right)X_{t-l}\right), \quad \text{by expression (3.11)} \\
&= rE\left(\alpha_0 X_{t-l} + \sum_{i=1}^{p}\alpha_i X_{t-i}X_{t-l}\right) \\
&= r\left(\alpha_0 E(X_{t-l}) + \sum_{i=1}^{p}\alpha_i E(X_{t-i}X_{t-l})\right) \\
&= r\alpha_0\mu + r\sum_{i=1}^{p}\alpha_i E(X_{t-i}X_{t-l}), \\
&= r\alpha_0\mu + r\alpha_l\gamma_{0,t-l} + r\sum_{\substack{i=1,\\ i\neq l}}^{p}\alpha_i\gamma_{|i-l|,t-i}, \quad \text{since } \mathrm{E}(\mathrm{X}_{t-i}X_{t-l}) = \begin{cases}\gamma_{0,t-l} & \text{if } i = l,\\ \gamma_{|i-l|,t} & \text{if } i \neq l,\end{cases} \\
&= r\alpha_0\mu + r\alpha_l\gamma_{0,t-l} + r\sum_{\substack{i=1,\\ i\neq l}}^{p}\alpha_i\gamma_{|i-l|,t}, \quad \text{replacing } \gamma_{s,t-i} \text{ by } \gamma_{st} \text{ for } i = 1, \ldots, p-1 \\
&= r\alpha_0\mu + r\alpha_l\gamma_{0,t-l} + r\left(\sum_{|i-l|=1}\alpha_i\gamma_{1t} + \cdots + \sum_{|i-l|=l}\alpha_i\gamma_{lt} + \cdots + \sum_{|i-l|=p-1}\alpha_i\gamma_{p-1,t}\right) \\
&= r\alpha_0\mu + \beta_{l0}\gamma_{0,t-l} + \beta_{l1}\gamma_{1t} + \cdots + \beta_{ll}\gamma_{lt} + \cdots + \beta_{l,p-1}\gamma_{p-1,t}, \quad \text{by definition} \\
&= r\alpha_0\mu + \beta_{l0}\gamma_{0,t-l} + \sum_{u=1}^{p-1}\beta_{lu}\gamma_{ut}
\end{aligned}
$$

To ensure that X_t is second-order stationary, we replaced $\gamma_{s,t-i}$ by γ_{st} for $i = 1, \ldots, p-1$, so

$$r\alpha_0\mu + \beta_{l0}\gamma_{0,t-l} + \sum_{u=1}^{p-1}\beta_{lu}\gamma_{ut} = 0.$$

It then follows that

$$B(\gamma_{1t}, \ldots \gamma_{p-1,t})^T = -(r\alpha_0\mu + \beta_{10}\gamma_{0,t-1}, \ldots r\alpha_0\mu + \beta_{p-1,0}\gamma_{0,t-p+1})^T,$$

from which it follows that

$$\gamma_{lt} = -r\alpha_0\mu\sum_{u=1}^{p-1}b_{lu} - \sum_{u=1}^{p-1}b_{lu}\beta_{u0}\gamma_{0,t-u}, \quad l = 1\ldots p-1.$$

The unconditional second moment can be rewritten as

$$\begin{aligned}
\gamma_{0t} &= C + (r+r^2)\left[\sum_{u=1}^{p}\alpha_u^2\gamma_{0,t-u} + \sum_{\substack{i,j=1\\ i\neq j}}^{p}\alpha_i\alpha_j\gamma_{|i-j|,t}\right] \\
&= C + (r+r^2)\left[\sum_{u=1}^{p}\alpha_u^2\gamma_{0,t-u} + \sum_{v=1}^{p-1}\sum_{|i-j|=v}\alpha_i\alpha_j\gamma_{vt}\right] \\
&= C_0 + (r+r^2)\left[\sum_{u=1}^{p}\alpha_u^2\gamma_{0,t-u} + \sum_{v=1}^{p-1}\sum_{|i-j|=v}\alpha_i\alpha_j\left(-\sum_{u=1}^{p-1}b_{vu}\beta_{u0}\gamma_{0,t-u}\right)\right] \\
&= C_0 + (r+r^2)\left[\sum_{u=1}^{p}\alpha_u^2\gamma_{0,t-u} - \sum_{u=1}^{p-1}\left(\sum_{v=1}^{p-1}\sum_{|i-j|=v}\alpha_i\alpha_j b_{vu}\beta_{u0}\right)\gamma_{0,t-u}\right] \\
&= C_0 + (r+r^2)\left[\sum_{u=1}^{p-1}\left(\alpha_u^2 - \sum_{v=1}^{p-1}\sum_{|i-j|=v}\alpha_i\alpha_j b_{vu}\beta_{u0}\right)\gamma_{0,t-u} + \alpha_p^2\gamma_{0,t-p}\right],
\end{aligned}$$

or equivalently,

$$\gamma_{0t} = C_0 + \sum_{u=1}^{p}C_u\gamma_{0,t-u}, \quad \text{where} \quad C_0 = C - (r^2+r^3)\alpha_0\mu\sum_{v=1}^{p-1}\sum_{|i-j|=v}\alpha_i\alpha_j\sum_{u=1}^{p-1}b_{vu}.$$

So the non-homogeneous difference equation has a stable solution if all roots of the equation

$$1 - C_1 z - \cdots - C_p z^p = 0$$

lie outside the unite circle. □

In the following, we consider some special cases of Propositions 3.1.7 and 3.1.9.

Corollary 3.1.10. *Let X_t be an NBINGARCH(p) process. For $p = 1$ and $p = 2$, the necessary and sufficient conditions for X_t to be first-order stationary are given by*

$$r\alpha_1 < 1 \quad \text{and} \quad r(\alpha_1+\alpha_2) < 1,$$

respectively. Now suppose that X_t is first-order stationary. Then for $p = 1$ and $p = 2$, the second-order stationary conditions are

$$(r+r^2)\alpha_1^2 < 1 \quad \text{and} \quad \delta_1+\delta_2 < 1,$$

respectively, where

$$\delta_1 = (r+r^2)\left(\alpha_1^2 + \frac{2r\alpha_1^2\alpha_2}{1-r\alpha_2}\right) \quad \text{and} \quad \delta_2 = (r+r^2)\alpha_2^2.$$

Proof. For $p = 1$, Proposition 3.1.7 implies that the first-order stationarity condition is that the root of the characteristic equation $1 - r\alpha_1 z = 0$ lies outside the unit circle, which is equivalent to the required condition:

$$r\alpha_1 < 1.$$

Similarly, by Proposition 3.1.9, the second-order stationary is equivalent to

$$(r + r^2)\alpha_1^2 < 1$$

Next, we prove for the case $p = 2$. By Proposition 3.1.7, the first-order stationarity condition is that all roots of the characteristic equation

$$1 - r\alpha_1 z - r\alpha_2 z^2 = 0$$

lie outside the unit circle, which is equivalent to the following condition:

$$r(\alpha_2 + \alpha_1) < 1, \quad r(\alpha_2 - \alpha_1) < 1, \quad |r\alpha_2| < 1. \tag{3.12}$$

The condition given in equation (3.12) is also equivalent to $r(\alpha_1 + \alpha_2) < 1$. Similarly, by Proposition 3.1.9, the second-order stationary condition is equivalent to

$$\delta_2 + \delta_1 < 1, \quad \delta_2 - \delta_1 < 1, \quad |\delta_2| < 1. \tag{3.13}$$

Then $\delta_1 > 0$ holds under the assumption of first-order stationarity, thus the condition in equation (3.13) is equivalent to $\delta_1 + \delta_2 < 1$. □

The following proposition extends the results of Weib [46] to the NBINGARCH(p) process and gives a set of equations from which the the autocovariance and autocorrelation functions can be obtained.

Proposition 3.1.11. *Let X_t be a second-order stationary NBINGARCH(p) process. Then, the autocovariance functions of X_t and λ_t, defined respectively as*

$$\gamma_X(k) = cov(X_t, X_{t-k}) \quad and \quad \gamma_\lambda(k) = cov(\lambda_t, \lambda_{t-k}),$$

satisfy the following equations:

$$\gamma_X(k) = \sum_{i=1}^{p} r\alpha_i \gamma_X(|k-i|) + \sum_{j=1}^{\min(k-1,q)} \beta_j \gamma_X(k-j) + \sum_{j=k}^{q} r^2 \beta_j \gamma_\lambda(j-k), \quad k \geq 1, \tag{3.14}$$

$$\gamma_\lambda(k) = \sum_{i=1}^{\min(k,p)} r\alpha_i \gamma_\lambda(k-i) + \sum_{i=k+1}^{p} \frac{\alpha_i}{r} \gamma_X(i-k) + \sum_{j=1}^{q} \beta_j \gamma_\lambda(|k-j|), \quad k \geq 0. \tag{3.15}$$

Proof. Let $\mathcal{I}_t$ be the σ-field generated by $\{\lambda_t, \lambda_{t-1}, \dots\}$, then we have

$$\begin{aligned} E(X_t|\mathcal{F}_{t-1}, \mathcal{I}_t) &= E(X_t|\mathcal{F}_{t-1}), \quad \text{since } \mathcal{I}_t \subseteq \mathcal{F}_t \\ &= r\lambda_t, \end{aligned} \tag{3.16}$$

where the last equality follows from equation (3.3). Now for $k \geqslant 0$, we have

$$\begin{aligned} \operatorname{cov}(X_t - r\lambda_t, r\lambda_{t-k}) &= E\Big[(X_t - r\lambda_t)(r\lambda_{t-k} - \mu)\Big], \quad \text{since } E(X_t - r\lambda_t) = 0 \\ &= E\left[E\Big((r\lambda_{t-k} - \mu)(X_t - r\lambda_t)|\mathcal{I}_t\Big)\right], \quad \text{by the law of total expectation} \\ &= E\left[(r\lambda_{t-k} - \mu)E\Big((X_t - r\lambda_t)|\mathcal{I}_t\Big)\right], \quad \text{since } \mathcal{I}_t \text{ is a } \sigma\text{-field for } \lambda_t \\ &= E\left[(r\lambda_{t-k} - \mu)\Big[E\big(E(X_t - r\lambda_t|\mathcal{F}_{t-1}, \mathcal{I}_t)|\mathcal{I}_t\big)\Big]\right], \quad \text{by the law of total expectation} \\ &= E\left[(r\lambda_{t-k} - \mu)\Big[E\big(E(X_t|\mathcal{F}_{t-1}, \mathcal{I}_t)|\mathcal{I}_t\big) - r\lambda_t\Big]\right] \\ &= E\left[(r\lambda_{t-k} - \mu)\Big[E\big(r\lambda_t|\mathcal{I}_t\big) - r\lambda_t\Big]\right], \quad \text{by equation (3.16)} \\ &= 0. \end{aligned} \tag{3.17}$$

Similarly, for $k < 0$, we have

$$\begin{aligned} \operatorname{cov}(X_t, X_{t-k} - r\lambda_{t-k}) &= E\Big[(X_t - \mu)(X_{t-k} - r\lambda_{t-k})\Big], \quad \text{since } E(X_t - \mu) = 0 \\ &= E\left[E\Big((X_t - \mu)\big(X_{t-k} - r\lambda_{t-k}\big)|F_{t-k-1}\Big)\right], \quad \text{by the law of total expectation} \\ &= E\left[(X_t - \mu)E\Big[\big(X_{t-k} - r\lambda_{t-k}\big)|F_{t-k-1}\Big]\right], \quad \text{since } \mathcal{F}_t \text{ is a } \sigma\text{-field for } X_t \\ &= E\left[(X_t - \mu)\Big[E\big(X_{t-k}|\mathcal{F}_{t-k-1}\big) - E\big(r\lambda_{t-k}|\mathcal{F}_{t-k-1}\big]\Big)\right] \\ &= E\left[(X_t - \mu)\Big[r\lambda_{t-k} - E\big(r\lambda_{t-k}|\mathcal{F}_{t-k-1}\big)\Big]\right], \quad \text{by equation (3.3)} \\ &= 0. \end{aligned} \tag{3.18}$$

Then from equations (3.17) and (3.18), we have

$$0 = \operatorname{cov}(X_t - r\lambda_t, r\lambda_{t-k}) = \operatorname{cov}(X_t, r\lambda_{t-k}) - \operatorname{cov}(r\lambda_t, r\lambda_{t-k}), \qquad k \geq 0, \tag{3.19}$$

$$0 = \operatorname{cov}(X_t, X_{t-k} - r\lambda_{t-k}) = \operatorname{cov}(X_t, X_{t-k}) - \operatorname{cov}(X_t, r\lambda_{t-k}), \qquad k < 0. \tag{3.20}$$

It then follows from equation (3.19) and (3.20) that

$$\operatorname{cov}(X_t, r\lambda_{t-k}) = \begin{cases} \operatorname{cov}(r\lambda_t, r\lambda_{t-k}), & k \geq 0, \\ \operatorname{cov}(X_t, X_{t-k}), & k < 0. \end{cases} \tag{3.21}$$

Now for $k \geq 0$, we have

$$\begin{aligned}
&\gamma_\lambda(k) = \operatorname{cov}(\lambda_t, \lambda_{t-k}) \\
&= \operatorname{cov}\left(\sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j}, \lambda_{t-k}\right), \quad \text{by equation (3.2)} \\
&= \sum_{i=1}^{p} \alpha_i \operatorname{cov}(X_{t-i}, \lambda_{t-k}) + \sum_{j=1}^{q} \beta_j \operatorname{cov}(\lambda_{t-j}, \lambda_{t-k}) \\
&= \sum_{i=1}^{p} \frac{\alpha_i}{r} \operatorname{cov}(X_{t-i}, r\lambda_{t-k}) + \sum_{j=1}^{q} \beta_j \operatorname{cov}(\lambda_{t-j}, \lambda_{t-k}) \\
&= \sum_{i=1}^{min(k,p)} \frac{\alpha_i}{r} \operatorname{cov}(r\lambda_{t-i}, r\lambda_{t-k}) + \sum_{i=k+1}^{p} \frac{\alpha_i}{r} \operatorname{cov}(X_{t-i}, X_{t-k}) + \sum_{j=1}^{q} \beta_j \operatorname{cov}(\lambda_{t-j}, \lambda_{t-k}), \quad \text{by (3.21)} \\
&= \sum_{i=1}^{min(k,p)} r\alpha_i \operatorname{cov}(\lambda_{t-i}, \lambda_{t-k}) + \sum_{i=k+1}^{p} \frac{\alpha_i}{r} \operatorname{cov}(X_{t-i}, X_{t-k}) + \sum_{j=1}^{q} \beta_j \operatorname{cov}(\lambda_{t-j}, \lambda_{t-k}) \\
&= \sum_{i=1}^{min(k,p)} r\alpha_i \gamma_\lambda(k-i) + \sum_{i=k+1}^{p} \frac{\alpha_i}{r} \gamma_X(i-k) + \sum_{j=1}^{q} \beta_j \gamma_\lambda(|k-j|),
\end{aligned}$$

which is the required equation (3.15). Similarly, for $k \geq 1$, we have

$$\begin{aligned}
&\gamma_X(k) = \operatorname{cov}(X_t, X_{t-k}) \\
&= r\operatorname{cov}(\lambda_t, X_{t-k}) \\
&= r\operatorname{cov}\left(\sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j}, X_{t-k}\right), \quad \text{by equation (3.2)} \\
&= \sum_{i=1}^{p} r\alpha_i \operatorname{cov}(X_{t-i}, X_{t-k}) + \sum_{j=1}^{q} \beta_j \operatorname{cov}(r\lambda_{t-j}, X_{t-k}) \\
&= \sum_{i=1}^{p} r\alpha_i \operatorname{cov}(X_{t-i}, X_{t-k}) + \sum_{j=1}^{min(k-1,q)} \beta_j \operatorname{cov}(X_{t-j}, X_{t-k}) + \sum_{j=k}^{q} r^2 \beta_j \operatorname{cov}(\lambda_{t-j}, \lambda_{t-k}), \quad \text{by (3.21)} \\
&= \sum_{i=1}^{p} r\alpha_i \operatorname{cov}(X_{t-i}, X_{t-k}) + \sum_{j=1}^{min(k-1,q)} \beta_j \operatorname{cov}(X_{t-j}, X_{t-k}) + \sum_{j=k}^{q} \beta_j \operatorname{cov}(r\lambda_{t-j}, r\lambda_{t-k}) \\
&= \sum_{i=1}^{p} r\alpha_i \gamma_X(|k-i|) + \sum_{j=1}^{\min(k-1,q)} \beta_j \gamma_X(k-j) + \sum_{j=k}^{q} r^2 \beta_j \gamma_\lambda(j-k),
\end{aligned}$$

which is the required equation (3.14). □

Corollary 3.1.12. *Let X_t be a second-order stationary NBINGARCH(p) process. Then the autocovariance function $\gamma_X(k)$ satisfies the equation*

$$\gamma_X(k) = \sum_{i=1}^{p} r\alpha_i \gamma_X(|k-i|), \quad k \geq 1. \tag{3.22}$$

Proof. This result follows directly from equation (3.14) with $q = 0$ in Proposition 3.1.11. □

The equations (3.14) from Corollary 3.1.12 closely resemble the Yule–Walker equations of the standard autoregressive model of order p (AR(p)). Consequently, the NBINGARCH model of order p can be identified using the partial autocorrelation function (PACF) $\phi_X(k)$. Specifically, it follows from Corollary 3.1.12 that $\phi_X(k) = 0$ for $k > p$, which facilitates the identification of the NBINGARCH model's order.

3.2 An Elaborated Example

Consider the NBINGARCH(1,1) model. By Proposition 3.1.11, we have

$$\begin{aligned}
\gamma_X(k) &= r\alpha_1\gamma_X(k-1) + \beta_1\gamma_X(k-1) \\
&= (r\alpha_1 + \beta_1)\gamma_X(k-1) \\
&= (r\alpha_1 + \beta_1)^2\gamma_X(k-2) \\
&= (r\alpha_1 + \beta_1)^3\gamma_X(k-3) \\
&\ \ \vdots \\
&= (r\alpha_1 + \beta_1)^{k-1}\gamma_X(1), \qquad k \geq 2,
\end{aligned} \tag{3.23}$$

$$\begin{aligned}
\gamma_\lambda(k) &= r\alpha_1\gamma_\lambda(k-1) + \beta_1\gamma_\lambda(k-1) \\
&= (r\alpha_1 + \beta_1)\gamma_\lambda(k-1) \\
&= (r\alpha_1 + \beta_1)^2\gamma_\lambda(k-2) \\
&= (r\alpha_1 + \beta_1)^3\gamma_\lambda(k-3) \\
&\ \ \vdots \\
&= (r\alpha_1 + \beta_1)^{k}\gamma_\lambda(0), \qquad k \geq 1.
\end{aligned} \tag{3.24}$$

Furthermore, Proposition 3.1.11 implies that

$$\begin{aligned}
\gamma_X(1) &= r\alpha_1\gamma_X(0) + r^2\beta_1\gamma_\lambda(0) \\
&= r\alpha_1\Big[(r+r^2)\mathrm{var}(\lambda_t) + rE(\lambda_t) + r(E(\lambda_t))^2\Big] + r^2\beta_1\gamma_\lambda(0), \quad \text{by equation (3.7)} \\
&= \Big(\alpha_1 r^2 + \alpha_1 r^3 + \beta_1 r^2\Big)\gamma_\lambda(0) + \alpha_1 r\mu + \alpha_1\mu^2, \quad \text{since } \gamma_\lambda(0) = \mathrm{var}(\lambda_t) \text{ and } rE(\lambda_t) = \mu \\
&= r^2\Big(\alpha_1 + r\alpha_1 + \beta_1\Big)\gamma_\lambda(0) + \alpha_1\Big(r\mu + \mu^2\Big),
\end{aligned} \tag{3.25}$$

and

$$\begin{aligned}
\gamma_\lambda(0) &= \frac{\alpha_1}{r}\gamma_X(1) + \beta_1\gamma_\lambda(1) \\
&= \frac{\alpha_1}{r}\Big[r^2\Big(\alpha_1 + r\alpha_1 + \beta_1\Big)\gamma_\lambda(0) + \alpha_1\Big(r\mu + \mu^2\Big)\Big] + \beta_1\Big(r\alpha_1 + \beta_1\Big)\gamma_\lambda(0), \quad \text{by (3.24) and (3.25)} \\
&= r\alpha_1\Big(\alpha_1 + r\alpha_1 + \beta_1\Big)\gamma_\lambda(0) + \beta_1\Big(r\alpha_1 + \beta_1\Big)\gamma_\lambda(0) + \alpha_1^2\mu + \frac{\alpha_1^2\mu^2}{r} \\
&= \Big(\alpha_1^2 r + \alpha_1^2 r^2 + 2\alpha_1\beta_1 r + \beta_1^2\Big)\gamma_\lambda(0) + \alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big) \\
&= \Big[(r\alpha_1 + \beta_1)^2 + r\alpha_1^2\Big]\gamma_\lambda(0) + \alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big),
\end{aligned}$$

where $\mu = r\alpha_0/[1-(r\alpha_1+\beta_1)]$ by Proposition 3.1.8. From the above expression, we have

$$\begin{aligned}
\gamma_\lambda(0) - \Big[(r\alpha_1 + \beta_1)^2 + r\alpha_1^2\Big]\gamma_\lambda(0) &= \alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big) \\
\Big\{1 - \Big[(r\alpha_1 + \beta_1)^2 + r\alpha_1^2\Big]\Big\}\gamma_\lambda(0) &= \alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big) \\
\gamma_\lambda(0) &= \frac{\alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big)}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}.
\end{aligned}$$

Since $\gamma_\lambda(0) = \text{var}(\lambda_t)$, it follows that the variance of λ_t is given by

$$\text{var}(\lambda_t) = \frac{\alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big)}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}. \tag{3.26}$$

Furthermore, the variance of X_t can be obtained as follows:

$$\begin{aligned}
\text{var}(X_t) &= (r + r^2)\text{var}(\lambda_t) + rE(\lambda_t) + r(E(\lambda_t))^2, \quad \text{by equation (3.7)} \\
&= (r + r^2)\text{var}(\lambda_t) + rE(\lambda_t) + \frac{(rE(\lambda_t))^2}{r} \\
&= (r + r^2)\text{var}(\lambda_t) + \mu + \frac{\mu^2}{r}, \quad \text{since } rE(\lambda_t) = \mu \\
&= (r + r^2)\left(\frac{\alpha_1^2\Big(\mu + \frac{\mu^2}{r}\Big)}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}\right) + \mu + \frac{\mu^2}{r}, \quad \text{by equation (3.26)} \\
&= \left(\frac{\alpha_1^2(r + r^2)}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2} + 1\right)\Big(\mu + \frac{\mu^2}{r}\Big) \\
&= \left(\frac{r\alpha_1^2 + r^2\alpha_1^2 + 1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}\right)\Big(\mu + \frac{\mu^2}{r}\Big) \\
&= \left(\frac{1 - (r\alpha_1 + \beta_1)^2 + r^2\alpha_1^2}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}\right)\Big(\mu + \frac{\mu^2}{r}\Big).
\end{aligned} \tag{3.27}$$

The autocovariance function of X_t is then obtained as follows:

$$
\begin{aligned}
\gamma_X(k) &= (r\alpha_1+\beta_1)^{k-1}\gamma_X(1), \quad \text{by equation (3.23)}\\
&= (r\alpha_1+\beta_1)^{k-1}\Big[r^2\Big(\alpha_1+r\alpha_1+\beta_1\Big)\gamma_\lambda(0)+\alpha_1\Big(r\mu+\mu^2\Big)\Big], \quad \text{by equation (3.25)}\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1^2r^2\Big(\alpha_1+r\alpha_1+\beta_1\Big)\Big(\mu+\dfrac{\mu^2}{r}\Big)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}+\alpha_1\Big(r\mu+\mu^2\Big)\right], \quad \text{by equation (3.26)}\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1^2r^2\Big(\alpha_1+r\alpha_1+\beta_1\Big)\Big(\mu+\dfrac{\mu^2}{r}\Big)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}+\alpha_1 r\Big(\mu+\frac{\mu^2}{r}\Big)\right]\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1^2r^2\Big(\alpha_1+r\alpha_1+\beta_1\Big)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}+\alpha_1 r\right]\Big(\mu+\frac{\mu^2}{r}\Big)\\
&= (r\alpha_1+\beta_1)^{k-1}\alpha_1 r\left[\frac{\alpha_1 r\Big(\alpha_1+r\alpha_1+\beta_1\Big)+1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\Big(\mu+\frac{\mu^2}{r}\Big)\\
&= (r\alpha_1+\beta_1)^{k-1}\alpha_1 r\left[\frac{\alpha_1^2r^2+\alpha_1\beta_1 r+1-(\alpha_1^2r^2+2\alpha_1\beta_1 r+\beta_1^2)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\Big(\mu+\frac{\mu^2}{r}\Big)\\
&= (r\alpha_1+\beta_1)^{k-1}\alpha_1 r\left[\frac{\alpha_1^2r^2+\alpha_1\beta_1 r+1-\alpha_1^2r^2-2\alpha_1\beta_1 r-\beta_1^2}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\Big(\mu+\frac{\mu^2}{r}\Big)\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1 r(1-\alpha_1\beta_1 r-\beta_1^2)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\Big(\mu+\frac{\mu^2}{r}\Big), \quad k\geq 1
\end{aligned}
\tag{3.28}
$$

Finally, the autocorrelation function of X_t is obtained by dividing (3.28) by (3.27) as follows:

$$
\begin{aligned}
\rho_X(k) &= \frac{\gamma_X(k)}{\gamma_X(0)}\\
&= \frac{(r\alpha_1+\beta_1)^{k-1}\left[\dfrac{\alpha_1 r(1-\alpha_1\beta_1 r-\beta_1^2)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\Big(\mu+\dfrac{\mu^2}{r}\Big)}{\left(\dfrac{1-(r\alpha_1+\beta_1)^2+r^2\alpha_1^2}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right)\Big(\mu+\dfrac{\mu^2}{r}\Big)}\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1 r(1-\alpha_1\beta_1 r-\beta_1^2)}{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}\right]\times\frac{1-(r\alpha_1+\beta_1)^2-r\alpha_1^2}{1-(r\alpha_1+\beta_1)^2+r^2\alpha_1^2}\\
&= (r\alpha_1+\beta_1)^{k-1}\left[\frac{\alpha_1 r(1-\alpha_1\beta_1 r-\beta_1^2)}{1-(r\alpha_1+\beta_1)^2+r^2\alpha_1^2}\right], \quad k\geq 1
\end{aligned}
\tag{3.29}
$$

By Lemma 2 in Ferland et al. [19] and by equations (3.28) and (3.29), the partial autocorrelation function of the NBINGARCH(1,1) process can be derived using the same approach as for the autoregressive moving average model of order $1,1$ (ARMA$(1,1)$).

Remark 3.2.1. *A necessary condition for the NBINGARCH(1,1) process X_t to be second-order stationary is that*

$$0 \leq (r\alpha_1 + \beta_1)^2 + r\alpha_1^2 < 1.$$

To see this, we first observe that equation (3.27) *shows that the variance of X_t is given by*

$$var(X_t) = \left(\frac{1 - (r\alpha_1 + \beta_1)^2 + r^2\alpha_1^2}{1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2}\right)\left(\mu + \frac{\mu^2}{r}\right).$$

Hence for $var(X_t)$ to be definite and well-defined, the denominator must be positive:

$$1 - (r\alpha_1 + \beta_1)^2 - r\alpha_1^2 > 0.$$

Rewriting, this condition is equivalent to the condition:

$$0 \leq (r\alpha_1 + \beta_1)^2 + r\alpha_1^2 < 1.$$

The first inequality follows from Definition 3.1.1, which states that $r > 0$, $\alpha_i \geq 0$ and $\beta_j \geq 0$, ensuring that each term is nonnegative.

3.3 Approaches for Parameter Estimation

Let $\{X_t\}$ be a time series of counts and let $\mathcal{F}_{t-1}$ be the σ-field generated by $\{X_{t-1}, X_{t-2}, \ldots\}$. Recall from Definition 3.1.1 that X_t is called a negative binomial integer-valued generalized autoregressive conditional heteroscedastic model of order p, q, denoted NBINGARCH(p, q), if the conditional distribution of X_t follows a negative binomial distribution:

$$X_t | \mathcal{F}_{t-1} \sim NB(r, p_t),$$

where r is a positive number, and the parameter p_t satisfies the relation:

$$\frac{1 - p_t}{p_t} = \lambda_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta_j \lambda_{t-j},$$

with $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, $i = 1, \ldots, p$, $j = 1, \ldots, q$, $p \geq 1$, $q \geq 0$.

Now let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_p)^T$, $\beta = (\beta_1, \ldots, \beta_q)^T$ and $\theta = (\alpha^T, \beta^T)^T = (\theta_0, \theta_1, \ldots, \theta_{p+q})^T$. In the following, we focus on the estimation of θ assuming that the parameter r is known. In practice, r can be simply estimated using the method given in Chapter 4. Assume that we observe $x_{1-p}, \ldots, x_0, x_1, \ldots, x_n$ from the NBINGARCH(p, q) model with true parameter θ^0 or α^0 and β^0. To estimate θ^0, we will first briefly discuss the Yule–Walker (YW) estimation and the conditional least squares (CLS) estimation for some special cases and then develop the maximum likelihood (ML) estimation.

3.3.1 Yule–Walker and Conditional Least Squares Estimations

Consider a special case of the NBINGARCH(p,q) model with $q=0$:

$$\begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r,p_t), \\ \frac{1-p_t}{p_t} = \lambda_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} \end{cases}$$

By Corollary 3.1.12, if X_t is a second-order stationary NBINGARCH(p) process, then the autocovariance function $\gamma_X(k)$ satisfies the equation

$$\gamma_X(k) = \sum_{i=1}^{p} r\alpha_i \gamma_X(|k-i|), \quad k \geq 1. \tag{3.30}$$

The YW estimation of the parameters $(\alpha_1,\dots,\alpha_p)^T$ is to insert sample autocovariances into equation (3.30) and then to solve for the parameters. To estimate the parameter α_0, we can use the method of moments with the help of $E(X_t)$ given in Proposition 3.1.8 as

$$\mu = E(X_t) = \frac{r\alpha_0}{1-\sum_{i=1}^{q}(r\alpha_i+\beta_i) - \sum_{i=q+1}^{p} r\alpha_i}.$$

The CLS estimates the mean $E(X_t)$ by minimizing

$$\sum_{t=1}^{n}(X_t - r\lambda_t)^2, \quad \text{or equivalently,} \quad \sum_{t=1}^{n}(X_t - r\alpha_0 - \sum_{i=1}^{p} r\alpha_i X_{t-i})^2.$$

We now consider another special case of the NBINGARCH(p,q) model with $p=1$ and $q=1$:

$$\begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r,p_t), \\ \frac{1-p_t}{p_t} = \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1 \lambda_{t-1}. \end{cases} \tag{3.31}$$

This model was discussed in Section 3.2, and its CLS estimates will be used as the starting values for the ML estimation discussed in Chapter 4 . By Lemma 2 in Ferland et al. [19] and by equations (3.28) and (3.29), we know that model (3.31) satisfies the ARMA$(1,1)$ model

$$(X_t-\mu) - a(X_{t-1}-\mu) = e_t + be_{t-1}, \tag{3.32}$$

where $a = r\alpha_1+\beta_1$, $b=-\beta_1$, and e_t is a white noise process with variance

$$\text{var}(e_t) = \frac{1-(r\alpha_1+\beta_1)^2}{1-(r\alpha_1+\beta_1)^2 - r\alpha_1^2}\left(\mu + \frac{\mu^2}{r}\right).$$

Note that equation (3.32) can also be written as

$$X_t = r\alpha_0 + aX_{t-1} + e_t + be_{t-1}.$$

The CLS estimation procedure is as follows:

(i) Let $Y_t = X_t - \frac{1}{n}\sum_{i=1}^{n} X_i$, fit the data by using a higher-order AR(p) model, then obtain the CLS estimators for the autoregressive coefficients $\hat{a}_i$ and define

$$\hat{e}_t = Y_t - \sum_{i=1}^{p^*} \hat{a}_i Y_{t-i}.$$

(ii) To obtain CLS estimates $\tilde{\alpha}_0$, $\tilde{a}$ and $\tilde{b}$, minimize

$$\sum_{i=1}^{n} (X_t - r\alpha_0 - aX_{t-1} - b\hat{e}_{t-1})^2$$

then $\tilde{\alpha}_1 = (\tilde{a} + \tilde{b})/r$, $\tilde{\beta}_1 = -\tilde{b}$.

Remark 3.3.1. *Based on equation (3.32), we can also choose the initial values for the ML estimates of the NBINGARCH(1,1) model by using the method of moments discussed in Zhu and Li [51].*

3.3.2 Maximum Likelihood Estimation

Recall from Proposition 3.1.3 that an NBINGARCH(p, q) process X_t has the conditional probability mass function given by

$$P(X_t = x_t | \mathcal{F}_{t-1}) = \binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t}, \qquad x_t = 0, 1, 2, \ldots,$$

where $p_t = 1/(1 + \lambda_t)$ and $q_t = 1 - p_t = \lambda_t/(1 + \lambda_t)$. To describe the maximum likelihood (ML) approach, we first derive the conditional log-likelihood function of X_t as follows:

$$\begin{aligned}
l(\theta) &= \log \prod_{t=1}^{n} \left[\binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t} \right] \\
&= \sum_{t=1}^{n} \log \left[\binom{x_t + r - 1}{r - 1} p_t^r (1 - p_t)^{x_t} \right] \\
&= \sum_{t=1}^{n} \left[\log \binom{x_t + r - 1}{r - 1} + r \log p_t + x_t \log(1 - p_t) \right] \\
&= \sum_{t=1}^{n} \log \binom{x_t + r - 1}{r - 1} + r \sum_{t=1}^{n} \log p_t + \sum_{t=1}^{n} x_t \log (1 - p_t) \\
&= \sum_{t=1}^{n} \log \binom{x_t + r - 1}{r - 1} + r \sum_{t=1}^{n} \log \left(\frac{1}{1 + \lambda_t} \right) + \sum_{t=1}^{n} x_t \log \left(\frac{\lambda_t}{1 + \lambda_t} \right), \quad \text{by Proposition 3.1.3} \\
&= \sum_{t=1}^{n} \log \binom{x_t + r - 1}{r - 1} + r \sum_{t=1}^{n} \Big(\log 1 - \log(1 + \lambda_t) \Big) + \sum_{t=1}^{n} \Big(x_t \big(\log \lambda_t - \log(1 + \lambda_t) \big) \Big)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{t=1}^{n} \log \binom{x_t + r - 1}{r - 1} - r \sum_{t=1}^{n} \Big(\log(1+\lambda_t) \Big) + \sum_{t=1}^{n} x_t \log \lambda_t - \sum_{t=1}^{n} x_t \log(1+\lambda_t) \\
&= \sum_{t=1}^{n} x_t \log \lambda_t - \sum_{t=1}^{n} (r + x_t) \log(1+\lambda_t) + \sum_{t=1}^{n} \log \binom{x_t + r - 1}{r - 1} \\
&= \sum_{t=1}^{n} x_t \log \lambda_t - \sum_{t=1}^{n} (r + x_t) \log(1+\lambda_t) + \sum_{t=1}^{n} \log \Big(\frac{x_t + r - 1}{(r-1)! x_t!} \Big) \\
&= \sum_{t=1}^{n} x_t \log \lambda_t - \sum_{t=1}^{n} (r + x_t) \log(1+\lambda_t) + \sum_{t=1}^{n} \Big(\log(x_t + r - 1) - \log(r-1)! - \log(x_t!) \Big) \\
&= \sum_{t=1}^{n} \Bigg[x_t \log \lambda_t - (r + x_t) \log(1+\lambda_t) + \sum_{v=1}^{x_t} \log(v + r - 1) - \log(x_t!) \Bigg].
\end{aligned}
$$

It is natural to estimate θ^0 by maximizing $l(\theta)$ given above. However, it is easy to see that the estimates have no closed form and the numerical optimization methods have to be used.

To obtain asymptotic standard errors of the ML estimation, we need the first derivatives of $l_t(\theta)$ with respect to $\theta_i (i = 0, 1, 2, \ldots, p + q)$, which has the form

$$
\frac{\partial l_t}{\partial \theta_i} = \left(\frac{X_t}{\lambda_t} - \frac{r + X_t}{1 + \lambda_t} \right) \frac{\partial \lambda_t}{\partial \theta_i}.
$$

The second derivatives are

$$
\frac{\partial^2 l_t}{\partial \theta_i \partial \theta_j} = \left(\frac{X_t}{\lambda_t} - \frac{r + X_t}{1 + \lambda_t} \right) \left(\frac{\partial^2 \lambda_t}{\partial \theta_i \partial \theta_j} \right) - \left(\frac{X_t}{\lambda_t^2} - \frac{r + X_t}{(1 + \lambda_t)^2} \right) \frac{\partial \lambda_t \partial \lambda_t}{\partial \theta_i \partial \theta_j}, \tag{3.33}
$$

for $i, j = 0, 1, \ldots, p + q$. Moreover, we have

$$
\begin{aligned}
\frac{\partial \lambda_t}{\partial \alpha_0} &= 1 + \sum_{k=1}^{q} \beta_k \frac{\partial \lambda_{t-k}}{\partial \alpha_0}, \\
\frac{\partial \lambda_t}{\partial \alpha_i} &= X_{t-i} + \sum_{k=1}^{q} \beta_k \frac{\partial \lambda_{t-k}}{\partial \alpha_i}, \qquad i = 1, \ldots, p, \\
\frac{\partial \lambda_t}{\partial \beta_j} &= \lambda_{t-j} + \sum_{k=1}^{q} \beta_k \frac{\partial \lambda_{t-k}}{\partial \beta_j}, \qquad j = 1, \ldots, q.
\end{aligned}
$$

Taking the expectation on both sides of equation (3.33), it follows from equation (3.3) that

$$
\begin{aligned}
E\left[\frac{\partial^2 l_t}{\partial \theta_i \partial \theta_j} \middle| \mathcal{F}_{t-1} \right] &= \left(\frac{E(X_t|\mathcal{F}_{t-1})}{\lambda_t} - \frac{r + E(X_t|\mathcal{F}_{t-1}}{1 + \lambda_t} \right) \frac{\partial^2 \lambda_t}{\partial \theta_i \partial \theta_j} \\
&\qquad - \left(\frac{E(X_t|\mathcal{F}_{t-1})}{\lambda_t^2} - \frac{r + E(X_t|\mathcal{F}_{t-1})}{(1 + \lambda_t^2)} \right) \frac{\partial \lambda_t \partial \lambda_t}{\partial \theta_i \partial \theta_j} \\
&= -r \left(\frac{1}{\lambda_t} - \frac{1}{1 + \lambda_t} \right) \frac{\partial \lambda_t \partial \lambda_t}{\partial \theta_i \partial \theta_j}.
\end{aligned} \tag{3.34}
$$

Similarly, from equations (3.3) and (3.4) we have

$$
\begin{aligned}
E\left[\frac{\partial l_t}{\partial \theta_i}\frac{\partial l_t}{\partial \theta_j}\middle|\mathcal{F}_{t-1}\right] &= E\left[\left(\frac{X_t}{\lambda_t} - \frac{r+X_t}{1+\lambda_t}\right)^2\middle|\mathcal{F}_{t-1}\right]\frac{\partial \lambda_t}{\partial \theta_i}\frac{\partial \lambda_t}{\partial \theta_j} \\
&= \frac{E((X_t - r\lambda_t)^2|\mathcal{F}_{t-1})}{\lambda_t^2(1+\lambda_t)^2}\frac{\partial \lambda_t}{\partial \theta_i}\frac{\partial \lambda_t}{\partial \theta_j} \\
&= \frac{r\lambda_t(1+\lambda_t)}{\lambda_t^2(1+\lambda_t)^2}\frac{\partial \lambda_t}{\partial \theta_i}\frac{\partial \lambda_t}{\partial \theta_j} \\
&= r\left(\frac{1}{\lambda_t} - \frac{1}{1+\lambda_t}\right)\frac{\partial \lambda_t}{\partial \theta_i}\frac{\partial \lambda_t}{\partial \theta_j}. \qquad (3.35)
\end{aligned}
$$

Then from equation (3.34) and (3.35), we obtain the information matrix equality

$$
-E\left[\frac{\partial^2 l_t}{\partial \theta_i \partial \theta_j}\middle|\mathcal{F}_{t-1}\right] = E\left[\frac{\partial l_t \partial l_t}{\partial \theta_i \partial \theta_j}\middle|\mathcal{F}_{t-1}\right], \qquad i,j = 0,1,\ldots,p+q.
$$

In Ferland et al. [19], it is shown that asymptotic standard errors of ML estimates can be computed from the following matrix:

$$
\frac{1}{n}(\hat{D}_n \hat{S}_n^{-1} \hat{D}_n)^{-1}, \qquad (3.36)
$$

where

$$
\hat{S}_n = \frac{1}{n}\sum_{t=1}^{n}\frac{\partial l_t \partial l_t}{\partial \theta \partial \theta^T} \quad \text{and} \quad \hat{D}^n = -\frac{1}{n}\sum_{t=1}^{n}\frac{\partial^2 l_t}{\partial \theta \partial \theta^T}.
$$

The matrix in equation (3.36) provides a consistent estimator for the asymptotic covariance matrix of the maximum likelihood (ML) estimates. Under standard regularity conditions, the ML estimator $\hat{\theta}$ is consistent and asymptotically normal, that is,

$$
\sqrt{n}(\hat{\theta} - \theta^0) \xrightarrow{d} N\left(0, D^{-1}SD^{-1}\right),
$$

where $D = E\left[-\frac{\partial^2 l_t}{\partial \theta \partial \theta^T}\right]$ and $S = E\left[\frac{\partial l_t \partial l_t^T}{\partial \theta \partial \theta^T}\right]$.

This asymptotic result enables statistical inference on the estimated parameters, such as constructing confidence intervals and hypothesis testing. In practice, the expressions for $\hat{D}_n$ and $\hat{S}_n$ are evaluated numerically due to the complexity of the model.

The next chapter focuses on the practical implementation of the maximum likelihood estimation for the NBINGARCH(p,q) model, including simulation studies to assess estimator performance and the application of the model to real-world data.

Chapter 4

Simulation Study and Application to Real Data

In this chapter, we present a simulation study to evaluate parameter estimation approaches, followed by a real data example demonstrating the application of the NBINGARCH(p, q) model. Specifically, Section 4.1 compares the finite-sample performance of the Yule–Walker, conditional least squares, and maximum likelihood estimates through a simulation study. In Section 4.2, the NBINGARCH(p, q) model is applied to an overdispersed count time series of syphilis data.

4.1 A Simulation Study

A simulation study was conducted to evaluate the finite sample performance of the Yule–Walker (YW), conditional least squares (CLS), and maximum likelihood (ML) estimates for the three most sparsely parameterized NBINGARCH models: NBINGARCH(1), NBINGARCH(2) and NBINGARCH(1,1), given by Definition 3.1.1 as follows:

$$\begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} \end{cases}, \quad \begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} \end{cases} \text{and} \quad \begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1 \lambda_{t-1}, \end{cases}$$

respectively, where $\lambda_t = (1 - p_t)/p_t$. Three set-ups were considered as follows:

(i) NBINGARCH(1) with $(\alpha_0^0, \alpha_1^0, r)^T$=$(A1)(2, 0.3, 1)^T$ and $(A2)(4, 0.3, 2)^T$;

(ii) NBINGARCH(2) with $(\alpha_0^0, \alpha_1^0, \alpha_2^0, r)^T = (B1)(2, 0.4, 0.2, 1)^T$ and $(B2)(3, 0.3, 0.1, 2)^T$;

(iii) NBINGARCH(1,1) with $(\alpha_0^0, \alpha_1^0, \beta_1^0, r)^T = (C1)(2, 0.2, 0.4, 1)^T$ and $(C2)(3, 0.1, 0.3, 2)^T$,

where θ^0 in $(\theta^0, r)^T$ denotes the true parameter values of the models. For the estimation of the parameters, by following Davis and Wu [12], it was assumed that r is known. For the maximization of the log-likelihood function, the constrained nonlinear optimization function fmincon in MATLAB was used and the CLS estimates were used as the initial values. Here the constrained conditions are $\alpha_0 > 0$ and the first-order stationary condition given in Proposition 3.1.7. In the CLS estimates for the NBINGARCH(1,1) models, we choose $p^* = [\sqrt{n}]$, where $[X]$ is the integer part of X. In simulations, we applied the mean absolute deviation error (MADE), which is defined by $\frac{1}{m}\sum_{j=1}^{m}\left|\hat{\theta}_j - \theta_j^0\right|$, as the evaluation criterion. The sample sizes were $n = 100, 500, 1000$, and the number of replications $m = 200$.

Table 4.1 presents a summary of the simulation results. It can be seen that as the sample size increases, the estimates seem to converge to the true parameter values. All three estimation methods seem to perform reasonably well but the ML estimation gave smaller absolute deviation errors than the YW estimation and CLS estimation in most cases. On the other hand, YW estimation and CLS estimation almost gave the same performance.

Table 4.1: Mean of estimates, MADEs (Within parentheses)
for the NBINGARCH models

Model	n	Method	α_0	α_1	α_2 or β_1
A_1	100	YW	2.0974 (0.2945)	0.2596 (0.1053)	
		CLS	2.0996 (0.3002)	0.2623 (0.1055)	
		MLE	2.0091 (0.2925)	0.2995 (0.1070)	
	500	YW	2.0419 (0.1628)	0.2813 (0.0673)	
		CLS	2.0426 (0.1625)	0.2817 (0.0673)	
		MLE	2.0155 (0.1330)	0.2924 (0.0542)	
	1000	YW	2.0197 (0.1168)	0.2917 (0.0433)	
		CLS	2.0199 (0.1171)	0.2920 (0.0433)	
		MLE	2.0061 (0.0884)	0.2972 (0.0348)	
A_2	100	YW	4.8325 (1.1238)	0.2455 (0.0716)	
		CL S	4.7443 (1.1711)	0.2534 (0.0743)	
		MLE	4.2613 (0.7532)	0.2822 (0.0526)	
	500	YW	4.2723 (0.5726)	0.2837 (0.0365)	
		CL S	4.2746 (0.5739)	0.2842 (0.0363)	
		MLE	4.0542 (0.4064)	0.2973 (0.0259)	
	1000	YW	4.2034 (0.4557)	0.2879 (0.0273)	
		CLS	4.2055 (0.4572)	0.2881 (0.0272)	
		MLE	4.0838 (0.3696)	0.2953 (0.0218)	
B_1	100	YW	2.6207 (0.7180)	0.3132 (0.1357)	0.1298 (0.1280)
		CLS	2.5983 (0.7297)	0.3179 (0.1374)	0.1347 (0.1283)
		MLE	2.1520 (0.5224)	0.3863 (0.1349)	0.1851 (0.1103)
	500	YW	2.3183 (0.4100)	0.3522 (0.0940)	0.1697 (0.0742)
		CL S	2.3210 (0.4119)	0.3523 (0.0939)	0.1706 (0.0739)
		MLE	2.2165 (0.3603)	0.3836 (0.0676)	0.2084 (0.0625)

Table 4.1 (continued): Mean of estimates, MADEs (Within parentheses) for the NBINGARCH models

Model	n	Method	α_0	α_1	α_2 or β_1
	1000	YW	2.0489 (0.1611)	0.1941 (0.0393)	0.1888 (0.0409)
		CLS	2.0495 (0.1614)	0.1941 (0.0393)	0.1892 (0.0408)
		MLE	2.0079 (0.1152)	0.1983 (0.0319)	0.1985 (0.0291)
B_2	100	YW	3.3614 (0.6155)	0.2561 (0.1084)	0.0580 (0.0973)
		CLS	3.3567 (0.6213)	0.2589 (0.1102)	0.0602 (0.0994)
		MLE	3.1657 (0.6016)	0.2894 (0.1211)	0.1168 (0.1048)
	500	YW	3.1618 (0.3224)	0.2790 (0.0714)	0.0816 (0.0469)
		CLS	3.1637 (0.3241)	0.2792 (0.0713)	0.0820 (0.0469)
		MLE	3.2222 (0.4234)	0.2817 (0.0631)	0.1264 (0.0694)
	1000	YW	3.0219 (0.2731)	0.3639 (0.0652)	0.1820 (0.0462)
		CLS	3.1246 (0.2751)	0.3632 (0.0640)	0.1825 (0.0460)
		MLE	3.0087 (0.2662)	0.3823 (0.0551)	0.2147 (0.0423)
C_1	100	CLS	4.5471 (2.5955)	0.1730 (0.0682)	0.1754 (0.2615)
		MLE	3.1267 (1.6182)	0.1807 (0.0636)	0.3202 (0.1860)
	500	CLS	2.8969 (0.9504)	0.1913 (0.0347)	0.3239 (0.1096)
		MLE	2.3873 (0.6300)	0.2004 (0.0282)	0.3570 (0.0932)
	1000	CLS	2.2061 (0.4132)	0.1985 (0.0361)	0.3971 (0.0718)
		MLE	2.0538 (0.2853)	0.2029 (0.0284)	0.4056 (0.0572)
C_2	100	CLS	4.8920 (2.0367)	0.0859 (0.0544)	0.0032 (0.3361)
		MLE	3.6942 (1.5658)	0.0878 (0.0511)	0.2177 (0.2475)
	500	CLS	4.0400 (2.0367)	0.0968 (0.0544)	0.1314 (0.3361)
		MLE	3.2651 (0.9807)	0.0926 (0.0243)	0.2893 (0.1695)
	1000	CLS	3.3139 (0.6241)	0.1027 (0.0276)	0.2032 (0.0849)
		MLE	3.1528 (0.4527)	0.1064 (0.0213)	0.3105 (0.0684)

Note: MADE (Mean absolute deviation error), YW (Yule-Walker), CLS (Conditional least squares), and MLE (Maximum likelihood estimation).

The simulation results show that YW, CLS, and ML estimation methods improve with larger sample sizes, with ML consistently yielding the smallest mean absolute deviation errors. ML proves to be the most efficient and reliable method for estimating NBINGARCH model parameters. Thus, ML estimation is recommended for accurate modeling of overdispersed count time series data.

4.2 Application to Real Data

We now apply the NBINGARCH model to an overdispersed count time series of syphilis data. This is part of a data set given in the R software ZIM-package sourced from the Centers for Disease Control and Prevention (CDC) Morbidity and Mortality Weekly Report CDC MMWR. The data set consists of the weekly number of syphilis cases in Maryland, United States, from the first week of January 2007 to the first week of May 2010, giving a total of 209 observations. To give an idea about the data structure, Table 4.2 shows some descriptive statistics of the syphilis time series data.

Table 4.2: Summary statistics of the syphilis time series data

Statistic	Value
Sample Size	209
Maximum	15
Minimum	0
Mean	3.473684
Median	3
Variance	9.279352
Skewness	0.660827
Kurtosis	0.137934
Standard Deviation	3.046203

The empirical mean and variance of the data are 3.473684 and 9.279352 respectively. These indicate that the true marginal distribution of the data exhibits overdispersion and is thus suitable for use as an example of an application of the NBINGARCH model. Other authors who considered this type of data include the Baltimore City Health Department [2], Centers for Disease Control and Prevention [11], and Maryland Department of Health and Mental Hygiene [37]. Figure 4.1 below shows the original series, the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the series.

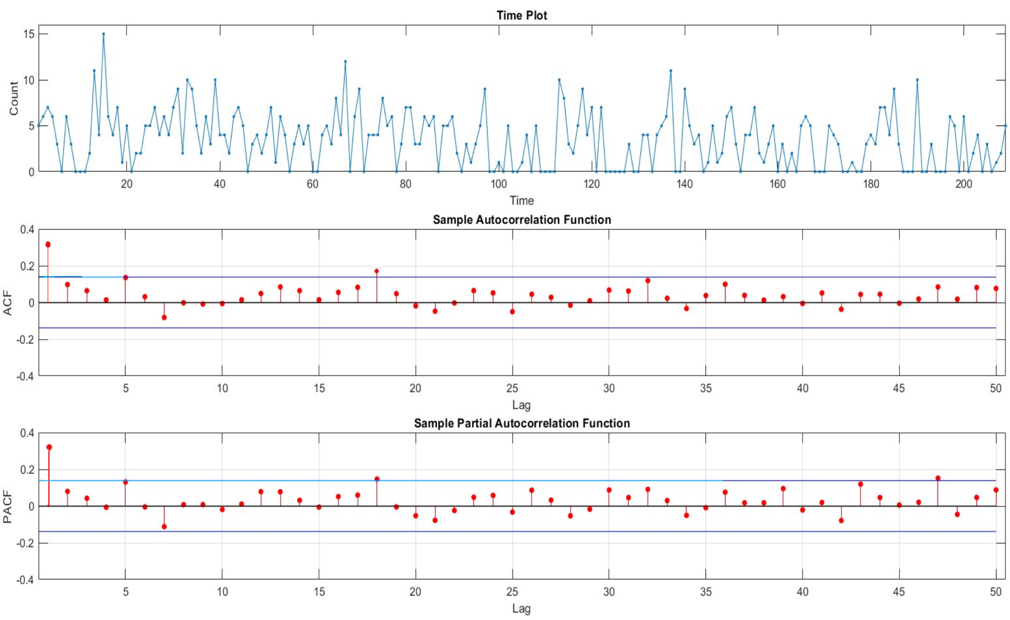

Figure 4.1: Syphilis data: (a) Time plot, (b) the sample autocorrelation function (SACF), and (c) the sample partial autocorrelation function (SPACF)

Figure 4.1 presents a time series with fluctuating counts, suggesting stationarity without any apparent trends or seasonality. Serial dependencies are evident, as indicated by the ACF, which displays significant correlations at multiple lags. When combined with the evidence of overdispersion shown in Table 4.2, these patterns indicate that the NBINGARCH model, which accommodates both overdispersion and serial dependence in count data, may be a suitable choice for modelling this dataset. Additionally, since only the autocorrelation and partial autocorrelation values at lag 1 exceed the significance bounds, the most suitable models to fit the data are the NBINGARCH(1) and NBINGARCH(1,1) models:

$$\begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} \end{cases} \quad \text{and} \quad \begin{cases} X_t|\mathcal{F}_{t-1} \sim NB(r, p_t), \\ \lambda_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1 \lambda_{t-1}, \end{cases}$$

respectively, where $\lambda_t = (1-p_t)/p_t$. The model parameters (θ, r) were jointly estimated using the approaches of Benjamin et al. [3] and Davis and Wu [12]. Specifically, the log-likelihood function $l(\theta)$ was maximized with respect to θ for selected values of r.

Searching for the maximizer of the log-likelihood function is implemented in MATLAB by using fmincon. Here the constrained conditions are $\alpha_0 > 0$ and the first-order stationary condition. We choose the CLS estimates as the corresponding initial values and set $\lambda_0 = \bar{X}$ and $\partial\lambda_0/\partial\theta_i = 0$. The estimate $\hat{r}$ is determined by the r value that yielded the smallest Akaike information criterion (AIC) or Bayesian information criterion (BIC). From Table 4.3 we estimate $\hat{r} = 1$. Asymptotic standard errors are computed by using equation (3.36).

Table 4.3: Akaike information criterion (AIC) and Bayesian information criterion (BIC) values with different initial values of r with NBINGARCH(1) and NBINGARCH(1,1) models

	Initial value of r	1	2	3	4	5
NBINGARCH(1)	AIC	990.9	995.9	1011.1	1025.7	1051.5
	BIC	1005.1	1006.2	1022.3	1037.2	1049.8
NBINGARCH(1,1)	AIC	991.8	996.2	1012.3	1027.2	1038.2
	BIC	1009	1009.1	1024.4	1039.0	1039.9

Furthermore, Table 4.3 indicates that the NBINGARCH(1) model consistently outperforms the NBINGARCH(1,1) model based on both the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Across different r values, the NBINGARCH(1) model consistently yields lower AIC and BIC values. These lower values indicate a better overall fit to the data while effectively managing the trade-off between model complexity and goodness-of-fit. This balance suggests that the NBINGARCH(1) model not only captures the underlying data dynamics more efficiently but also avoids the risk of overfitting. Consequently, the NBINGARCH(1) model emerges as the more suitable choice for modeling the given syphilis time series data.

For comparison, we also consider the Poisson and double Poisson (DP) models, introduced by Ferland et al. [19] and Heinen [27], respectively. The DP model is based on the double Poisson distribution proposed by Efron [17]. The double Poisson model is specified as:

$$X_t|\mathcal{F}_{t-1} \sim \mathcal{DP}(\lambda_t, \gamma),$$

where the approximate conditional probability mass function is given by:

$$f(X_t) = \left(\gamma^{\frac{1}{2}} e^{-\gamma\lambda_t}\right)\left(\frac{e^{-X_t} X_t^{X_t}}{X_t!}\right)\left(\frac{e\lambda_t}{X_t}\right)^{\gamma X_t}, \qquad \gamma > 0, \qquad X_t = 0, 1, 2, \ldots, \tag{4.1}$$

and λ_t is defined in equation (3.2). As shown in Efron [17], the conditional mean of the double Poisson model is λ_t, and the conditional variance is approximately λ_t/γ. For the model in equation (4.1), the log-likelihood function, along with its first and second derivatives, is provided in Heinen [27]. However, to maintain self-containment in this document, we derive and present these details here. The log-likelihood function is derived as follows:

$$\begin{aligned}
l_t(\lambda_t, \gamma) &= \log \prod_{t=1}^{n} \left(\gamma^{\frac{1}{2}} e^{-\gamma\lambda_t}\right)\left(\frac{e^{-X_t} X_t^{X_t}}{X_t!}\right)\left(\frac{e\lambda_t}{X_t}\right)^{\gamma X_t} \\
&= \sum_{t=1}^{n} \log \left[\left(\gamma^{\frac{1}{2}} e^{-\gamma\lambda_t}\right)\left(\frac{e^{-X_t} X_t^{X_t}}{X_t!}\right)\left(\frac{e\lambda_t}{X_t}\right)^{\gamma X_t}\right] \\
&= \sum_{t=1}^{n} \left[\frac{1}{2}\log\gamma - \gamma\lambda_t - \log X_t! - X_t + X_t \log X_t - \gamma X_t \log(e\lambda_t) - \gamma X_t \log X_t\right] \\
&= \sum_{t=1}^{n} \frac{1}{2}\log\gamma - \sum_{t=1}^{n} \gamma\lambda_t - \sum_{t=1}^{n} \log X_t! - \sum_{t=1}^{n} X_t + \sum_{t=1}^{n} X_t \log X_t \\
&\quad - \sum_{t=1}^{n} \gamma X_t \log(e\lambda_t) - \sum_{t=1}^{n} \gamma X_t \log X_t \\
&= \frac{n}{2}\log\gamma - \gamma\sum_{t=1}^{n} \lambda_t - \sum_{t=1}^{n} \log X_t! - \sum_{t=1}^{n} X_t + \sum_{t=1}^{n} X_t \log X_t \\
&\quad - \gamma\sum_{t=1}^{n} X_t \log(e\lambda_t) - \gamma\sum_{t=1}^{n} X_t \log X_t
\end{aligned}$$

Furthermore, the first derivatives are derived as follows:

$$\begin{aligned}
\frac{\partial l_t(\lambda_t, \gamma)}{\partial\gamma} &= \frac{n}{2\gamma} - \sum_{t=1}^{n} \lambda_t - \sum_{t=1}^{n} X_t \log(e\lambda_t) - \sum_{t=1}^{n} X_t \log X_t \\
&= \frac{n}{2\gamma} - \sum_{t=1}^{n} \lambda_t - \sum_{t=1}^{n} X_t\big(\log(e\lambda_t) + 1 + \log X_t\big), \\
\frac{\partial l_t(\lambda_t, \gamma)}{\partial\lambda_t} &= -\gamma\sum_{t=1}^{n} \frac{X_t e}{\lambda_t} - \gamma\sum_{t=1}^{n} 1, = -\gamma\sum_{t=1}^{n} \frac{X_t e}{\lambda_t} - n\gamma,
\end{aligned}$$

and the second derivatives are derived as follows:

$$\frac{\partial^2 l_t(\lambda_t,\gamma)}{\partial\gamma^2} = -\frac{n}{2\gamma^2},$$
$$\frac{\partial^2 l_t(\lambda_t,\gamma)}{\partial\lambda_t^2} = \gamma\sum_{t=1}^{n}\frac{X_t}{\lambda_t^2},$$
$$\frac{\partial^2 l_t(\lambda_t,\gamma)}{\partial\gamma\partial\lambda_t} = -\sum_{t=1}^{n}\frac{X_t}{\lambda_t} - 1.$$

Due to the similar structure of their first derivatives, the Poisson and double Poisson models yield comparable parameter estimates for θ. As noted by Heinen [27], the expected Hessian matrix is block-diagonal, making it efficient to estimate the dispersion parameter γ independently of θ. Table 4.4 summarises the parameter estimates along with their asymptotic standard errors, presented in parentheses. The Akaike information criterion (AIC) and Bayesian information criterion (BIC) values are also provided.

Table 4.4: Parameter estimates with Poisson (P), double Poisson (DP) and negative binomial INGARCH(1) and INGARCH(1,1) models. The asymptotic standard errors are shown in parentheses

	Model	$\hat{\alpha_0}$	$\hat{\alpha_1}$	$\hat{\beta_1}$	$\hat{\gamma}$ or $\hat{r}$	AIC	BIC
INGARCH(1)	P	2.90	0.47			1155.8	1162.5
		(0.1050)	(0.0059)				
	DP	2.8937	0.1652		0.3161	997.3	1009.9
		(0.1040)	(0.0057)		(0.0005)		
	NB	2.32	0.02		1	990.9	957.2
		(0.1020)	(0.0017)				
INGARCH(1,1)	P	1.13	0.41	0.78		1202.9	1205.4
		(0.7565)	(0.0059)	(0.0853)			
	DP	1.10	0. 23	0.51	0.3175	1152.5	1162.7
		(0.7515)	(0.0052)	(0.0853)	(0.0006)		
	NB	1.02	0.04	0.13	1	991.8	1005.1
		(0.4762)	(0.0035)	(0.01149)			

Note: AIC (Akaike information criterion), and BIC (Bayesian information criterion)

Based on AIC and BIC values, the improvement in fit achieved by adding more parameters to the INGARCH models is not significant. Moreover, the INGARCH(1) model captures the overdispersion just as effectively as the more complex INGARCH(1,1), so the more sparsely parameterized INGARCH(1) model should be preferred. For the INGARCH(1) models, the values of AIC and BIC show that the NB fit is the best and the DP fit is the second.

For model adequacy, the fitted conditional mean $\hat{r}\hat{\lambda_t}$ from the INGARCH(1) model is given in Figure 4.2, and the resulting Pearson residual is defined by

$$r_{1t} = \frac{X_t - \hat{r\lambda_t}}{[\hat{r\lambda_t}(1+\hat{\lambda}_t)]^{\frac{1}{2}}}, \tag{4.2}$$

where $\hat{\lambda_t} = \hat{\alpha_0} + \hat{\alpha_1} X_{t-1}$, and the residual analysis is shown in Figure 4.3. Figure 4.3(a),(b) shows the plots for the ACF and PACF of the Pearson residuals. To check for autocorrelation within the residuals up to, say, lag 15, we conduct the Ljung–Box test. The hypotheses are:

- Null Hypothesis (H_0): No autocorrelation up to lag 15.
- Alternative Hypothesis (H_1): There is at least one non zero autocorrelation up to lag 15.

The Ljung–Box test statistic is

$$Q = n(n+2)\sum_{k=1}^{15}\frac{\hat{\rho}_k^2}{n-k} = 209(211)\sum_{k=1}^{15}\frac{\hat{\rho}_k^2}{209-k} = 19.8567 \tag{4.3}$$

Since $Q < \mathcal{X}^2_{0.05}(14) = 23.6848$, we fail to reject the null hypothesis. Alternatively, since the p-value (0.1772) is greater than the significance level of 0.05, we fail to reject the null hypothesis. This suggests that there is no significant autocorrelation in the Pearson residuals up to lag 15, implying that the model adequately captures the data's dynamics.

Figure 4.3(c),(d) shows the kernel density and normal Q–Q plots for the Pearson residuals, which appear highly non-normally distributed. We also examine the normalized conditional randomized quantile residuals of Dunn and Smyth [16] and Benjamin et al. [3], defined as $r_{2t} = \boldsymbol{\phi}^{\mathbf{-1}}(u_t)$, where $\boldsymbol{\phi}^{-1}$ is the inverse cumulative distribution function of a standard normal variable and u_t is randomly drawn from the uniform interval $[F(X_t - 1, \hat{\alpha}, \hat{r}), F(X_t, \hat{\alpha}, \hat{r})]$. Here, $F(X_t, \hat{\alpha}, \hat{r})$ is the fitted conditional negative binomial cumulative distribution function. The residuals analysis in Figure 4.4 shows no residual correlation. Kernel density and normal Q–Q plots appear roughly normal. Figure 4.5 shows the normalized randomized quantile residuals scattered randomly over time, confirming the model adequately fits the data.

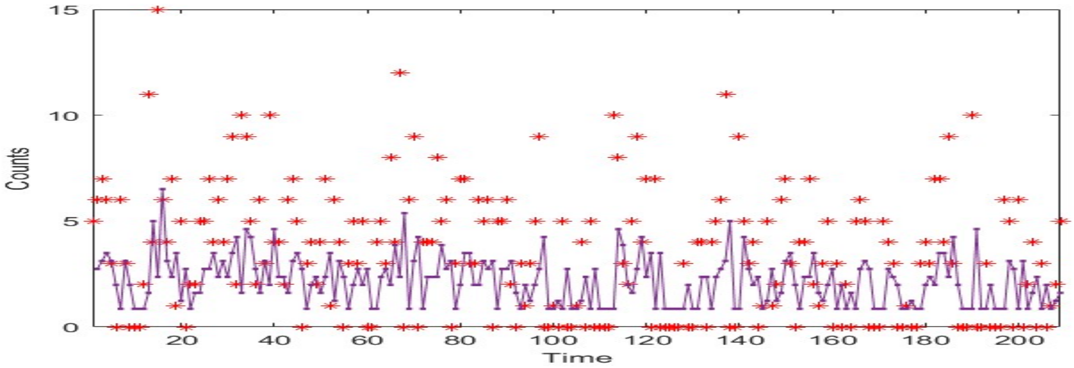

Figure 4.2: Syphilis data (∗) and fitted mean values (–)
from the NBINGARCH(1) model

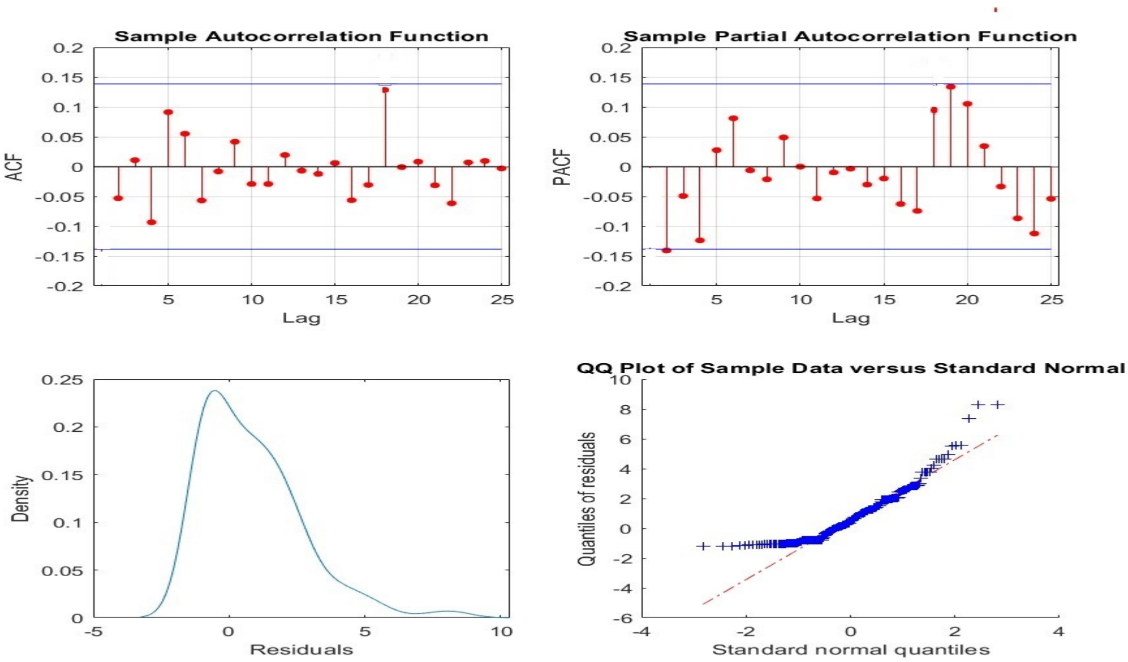

Figure 4.3: Pearson residual analysis: (a) The autocorrelation function of residuals, (b) The partial autocorrelation function of residuals, (c) density estimation of the residuals, (d) Q–Q plot of the residuals

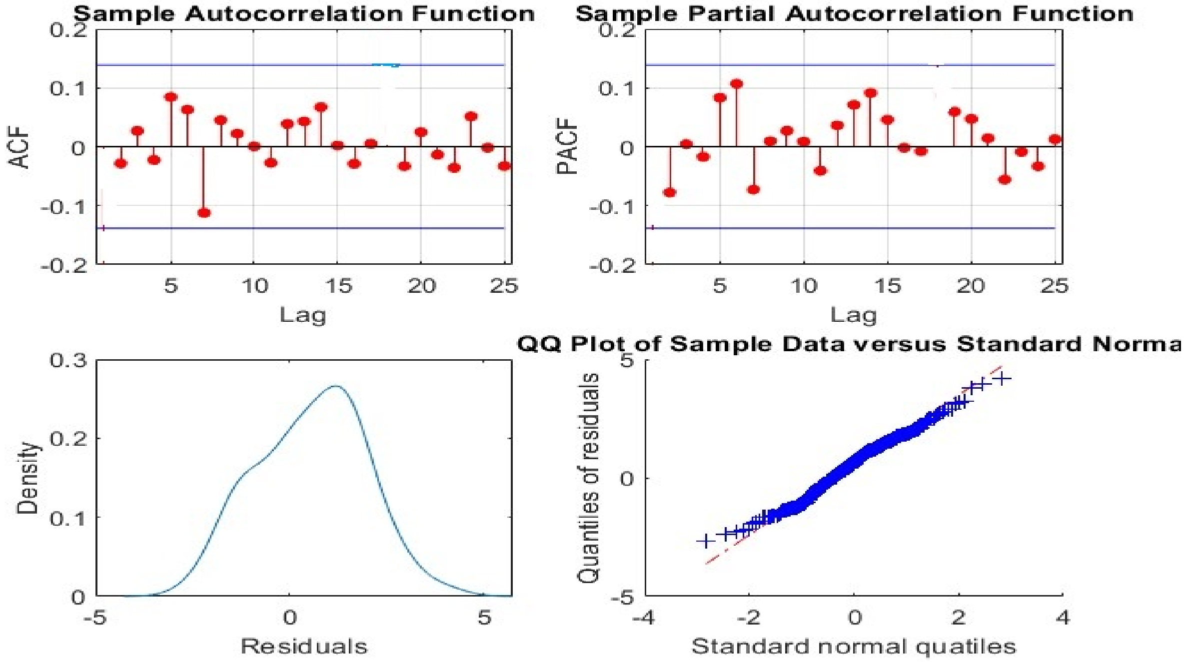

Figure 4.4: Randomized quantile residual analysis. (a) the autocorrelation function of the residuals, (b) the partial autocorrelation function of the residuals, (c) density estimation of the residuals, (d) Q–Q plot of the residuals

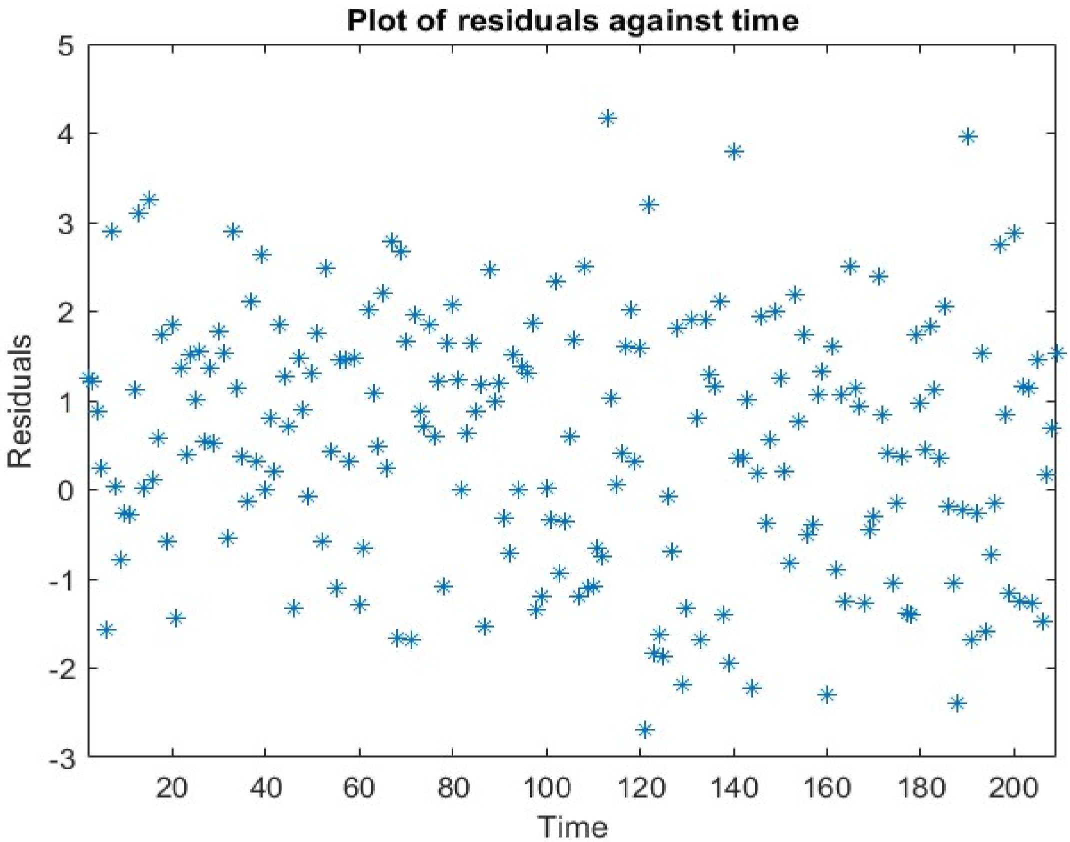

Figure 4.5: Randomized quantile residual against time

Table 4.5 gives the observations X_t for which the upper probability $1-F(X_t-1, \hat{\alpha}, \hat{r}) \leq 0.02$. The table compares tail probabilities for extreme residuals using Poisson, double Poisson, and negative binomial models. The Poisson model consistently shows very low probabilities, highlighting its poor performance in handling extreme values. The double Poisson model offers slight improvement but still struggles with extreme counts. In contrast, the negative binomial model consistently yields higher tail probabilities, effectively capturing the variability in the data. This makes the negative binomial model the most suitable for modelling overdispersed data. This observation can be attributed to the differences in their conditional variances: λ_t, λ_t/γ_t, and $r\lambda_t(1+\lambda_t)$ for the Poisson, double Poisson, and negative binomial models, respectively.

Table 4.5: Tail probabilities for potential extreme residuals

Week	X_t	Poisson	Double Poisson	Negative Binomial
13	11	0.00358		
15	15	0.00015	0.00448	0.00538
33	10	0.00896		
41	10	0.01742		
67	12	0.00123	0.00827	0.01530

Chapter 5

Discussion

In this chapter, we discuss the results and implications of the negative binomial INGARCH model, reflecting on its effectiveness in addressing overdispersion in count time series data. The aim and objectives of this dissertation are restated and discussed in Section 5.1 while the employed methodology to achieve these goals is discussed in Section 5.2. In Section 5.3, we discuss the structure of the negative binomial INGARCH(p, q) (NBINGARCH(p, q)) model, including its formulation, suitability for modelling count time series with overdispersion and extreme values, and the stationarity of the model. Approaches for parameter estimation of the model are discussed in Section 5.4. Here, we discuss the Yule–Walker and conditional least squares estimation methods for specific cases of the NBINGARCH(p, q) model, followed by a discussion on the development of the maximum likelihood estimation approach. A simulation study to evaluate parameter estimation approaches is discussed in Section 5.5. Here, the simulation study compares the finite-sample performance of the Yule–Walker, conditional least squares, and maximum likelihood estimates. In Section 5.6, we discuss the application of the NBINGARCH(p, q) model to an overdispersed count time series of syphilis data. In Section 5.7, we discuss the comparison of the performance of the negative binomial INGARCH model to that of the Poisson and the double Poisson INGARCH models.

5.1 On the Aim and Objectives

The primary aim of this dissertation was to develop a negative binomial INGARCH model that can effectively address overdispersion and extreme values in count time series data, focusing on its structure, parameter estimation, and application to real-world data. To achieve this aim, the following research objectives were formulated:

(i) **To formulate a negative binomial INGARCH model for overdispersed count time series:** This involves integrating the negative binomial distribution within the INGARCH framework to better capture overdispersion in count time series data, by allowing the conditional variance to exceed the conditional mean.

(ii) **To develop an approach for parameter estimation for the proposed model:** This involves employing various parameter estimation techniques, including maximum likelihood, Yule-Walker, and conditional least squares, to obtain reliable and consistent estimates of the model parameters.

(iii) **To apply the proposed model to real-world count data and assess its performance:** This involves applying the NBINGARCH(p, q) model to an overdispersed count time series of syphilis data, and assessing the model performance using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC).

(iv) **To compare the performance of the negative binomial INGARCH model to that of the Poisson and the double Poisson INGARCH models:** This involves comparing the NBINGARCH(p, q) model's performance to that of the Poisson INGARCH(p, q) and double Poisson INGARCH(p, q) models based on their ability to capture extreme values through the analysis of tail probabilities for residuals.

5.2 On the Methodology

The methodology employed in this study provides a comprehensive framework for modeling overdispersed count time series data using the NBINGARCH(p, q) model. The structured approach encompasses key aspects such as model formulation, parameter estimation, simulation studies, application to real-world data, and model comparison. Each of these components contributes to the robustness and applicability of the proposed model.

5.2.1 Model Formulation

The formulation of the NBINGARCH(p, q) model is well-suited for handling count data that exhibit overdispersion. By specifying the conditional distribution of X_t as negative binomial, the model accommodates variability exceeding the mean, which is commonly observed in real-world scenarios such as epidemiology and finance. The dynamic equation governing the conditional mean, expressed as a function of past observations and conditional expectations, ensures that both short-term dependencies and long-term trends are effectively captured.

The stationarity condition, derived from the characteristic equation, provides critical insights into the stability of the model. Ensuring that the roots lie outside the unit circle guarantees that the model produces meaningful forecasts without diverging over time. The constraints on parameters, such as $\alpha_i \geq 0$ and $\beta_j \geq 0$, ensure interpretability and practical relevance. Moreover, the inclusion of higher-order terms in p and q enables the model to capture more complex temporal dependencies. This flexibility makes the NBINGARCH(p, q) model particularly useful for datasets with intricate autocorrelation structures. Additionally, the negative binomial framework inherently accounts for overdispersion, providing a better fit compared to simpler Poisson-based models. The ability to incorporate covariates enhances the model's applicability by integrating external factors influencing count dynamics.

5.2.2 Parameter Estimation

The choice of maximum likelihood estimation (MLE) as the primary estimation technique is justified by its asymptotic efficiency and desirable statistical properties. The conditional log-likelihood function is maximized numerically using MATLAB's `fmincon` function, incorporating constraints to ensure stationarity and non-negativity of parameters. The use of conditional least squares (CLS) estimates as initial values aids in convergence and stability of the optimization process. An important consideration is the selection of the dispersion parameter r, which is determined based on model selection criteria, namely AIC and BIC. This approach balances model complexity with goodness-of-fit, leading to an optimal representation of the underlying data.

5.2.3 Simulation Study

The simulation study is meticulously designed to evaluate the finite sample performance of different estimation methods, including YW, CLS, and ML. The consideration of three model structures: NBINGARCH(1), NBINGARCH(2), and NBINGARCH(1,1), allows for a comprehensive assessment across varying levels of model complexity. The use of the mean absolute deviation error (MADE) as the evaluation criterion provides an intuitive and robust measure of estimation accuracy. By varying sample sizes and conducting multiple replications, the study ensures that conclusions drawn are statistically reliable.

5.2.4 Application to Real Data

The application of the NBINGARCH(p, q) model to syphilis count data showcases its practical utility. The dataset's overdispersion, indicated by a variance greater than the mean, underscores the suitability of the negative binomial distribution over the Poisson alternative. The examination of ACF and PACF plots further aids in identifying the appropriate model order. The step-by-step approach, from data exploration to model selection using AIC and BIC, offers a replicable framework for analyzing similar datasets. The Pearson residual analysis provides additional validation, ensuring that the model adequately captures the data's underlying dynamics.

5.2.5 Model Comparison

The comparative evaluation against Poisson and double Poisson INGARCH models underscores the strengths of the NBINGARCH(p, q) model. The use of AIC and BIC as primary evaluation metrics ensures a balanced assessment of model fit and complexity. The consideration of tail probabilities provides deeper insights into the model's ability to capture extreme values, which is particularly crucial in applications involving rare events.

5.3 On the Structure of the NBINGARCH(p, q) Model

The NBINGARCH(p, q) model, as defined in Definition 3.1.1, extends the family of integer-valued time series models by incorporating the negative binomial distribution. The conditional distribution $X_t|\mathcal{F}_{t-1} \sim NB(r, p_t)$ introduces flexibility through the parameter r, which governs the degree of overdispersion. The relationship between the parameters p_t and λ_t (equation (3.2)) allows for the inclusion of both autoregressive and moving average effects, making it a natural extension of INGARCH models to settings where count data exhibit high variability. This formulation effectively bridges the gap between theory and practical applications by addressing the limitations of Poisson-based models, which often struggle with overdispersed data. By incorporating r and linking p_t to past observations and conditional means, the model accommodates a wide variety of real-world scenarios.

Proposition 3.1.4 derives the conditional mean and variance of the NBINGARCH(p, q) process. The conditional mean $E(X_t|\mathcal{F}_{t-1}) = r\lambda_t$ scales directly with the mean parameter λ_t, while the conditional variance $\text{var}(X_t|\mathcal{F}_{t-1}) = r\lambda_t(1 + \lambda_t)$ grows quadratically with λ_t. This quadratic growth of the variance with respect to the mean highlights the overdispersed nature of the model. Corollary 3.1.5 further reinforces this property, showing that both the conditional and unconditional variances of X_t exceed their respective means. This overdispersion is crucial for capturing the high variability observed in many count time series datasets, such as epidemiological cases, financial transactions, or customer arrivals.

Stationarity is a fundamental property for ensuring the long-term stability of a time series model. Proposition 3.1.7 provides necessary and sufficient conditions for mean stationarity of the NBINGARCH(p, q) process. Specifically, all roots of the characteristic equation (3.8) must lie outside the unit circle. This condition ensures that the process maintains a stable mean over time without diverging. Proposition 3.1.8 offers an explicit expression for the stationary mean of the process, emphasizing the role of parameters α_i, β_j, and r in determining the stability of the model. The constraints $0 \leq \sum_{i=1}^{q}(r\alpha_i + \beta_i) + \sum_{i=q+1}^{p} r\alpha_i < 1$ are critical for ensuring first-order stationarity, which lays the foundation for deriving further moment properties. For second-order stationarity, Proposition 3.1.9 establishes additional conditions involving the roots of another characteristic equation. The coefficients C_u and C_p are derived from the parameters of the model, and their relationships with the matrices B and B^{-1} reflect the intricate dependence structure in higher-order processes.

The autocovariance functions of X_t and λ_t, as derived in Proposition 3.1.11, provide valuable insights into the dependence structure of the NBINGARCH process. equations (3.14) and (3.15) capture the relationships between current and lagged values of X_t and λ_t, incorporating the effects of both autoregressive and moving average components. These equations highlight the rich dependence structure of the NBINGARCH model, enabling it to capture complex temporal patterns in count time series. The inclusion of the overdispersion parameter r improves the model's ability to capture high variability and extreme observations.

5.4 On the Approaches for Parameter Estimation

Several approaches for estimating the parameters of the NBINGARCH(p, q) model have been considered in this dissertation, including the following:

(i) **Maximum Likelihood Estimation:** This approach provided consistent and asymptotically efficient estimates by maximizing the likelihood function of the observed data. MLE is widely used due to its desirable statistical properties such as efficiency, consistency, and asymptotic normality. However, it requires the specification of the correct likelihood function and can be computationally intensive, particularly for complex models with a large number of parameters.

(ii) **Yule-Walker Estimation:** Based on the autocovariance function, this method offered a computationally efficient alternative but was sensitive to model assumptions. The Yule-Walker equations relate the model parameters to the autocorrelations of the observed data, providing a relatively straightforward estimation procedure. However, the accuracy of the YW estimates heavily depends on the underlying assumptions of stationarity and the correct specification of the autocovariance structure, making it less robust in the presence of model misspecification.

(iii) **Conditional Least Squares :** This method minimized the sum of squared deviations between observed and predicted values, serving as an effective initial estimator. CLS estimation is simple to implement and computationally less demanding compared to MLE. It provides reasonable estimates when the model structure is approximately known and serves as a good starting point for more sophisticated estimation methods. However, it does not account for the full probabilistic structure of the data, which can lead to efficiency loss compared to MLE.

5.5 On the Simulation Study

A simulation study was conducted to assess the finite sample performance of the Yule–Walker (YW), conditional least squares (CLS), and maximum likelihood estimation (MLE) methods for the NBINGARCH(1), NBINGARCH(2), and NBINGARCH(1,1) models. The study used different parameter settings for each model type, with sample sizes $n = 100, 500, 1000$ and 200 replications. The mean absolute deviation error (MADE) was applied as the evaluation metric. Table 4.1 presents a summary of the simulation results. It can be seen that as the sample size increases, the estimates seem to converge to the true parameter values. All three estimation methods seem to perform reasonably well but the ML estimation gave smaller absolute deviation errors than the YW estimation and CLS estimation in most cases. On the other hand, YW estimation and CLS estimation almost gave the same performance. The simulation results show that YW, CLS, and ML estimation methods improve with larger sample sizes, with ML consistently yielding the smallest mean absolute deviation errors.

ML proves to be the most efficient and reliable method for estimating NBINGARCH model parameters. Thus, ML estimation is recommended for accurate modeling of overdispersed count time series data.

5.6 On the Application to Real Data

The application of the NBINGARCH model to overdispersed syphilis count time series data demonstrates its effectiveness in capturing both overdispersion and serial dependence. Table 4.2 presents descriptive statistics of the syphilis time series, with an empirical mean and variance of 3.473684 and 9.279352, respectively. These values confirm the presence of overdispersion, justifying the use of the negative binomial distribution. The significant autocorrelation at lag 1 in the ACF and PACF plots in Figure 4.1 further supports the suitability of the NBINGARCH(1) and NBINGARCH(1,1) models. Results in Table 4.3 indicate that the NBINGARCH(1) model consistently outperforms the NBINGARCH(1,1) model based on the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). This suggests that the added complexity of the NBINGARCH(1,1) model does not substantially improve model fit, aligning with the principle of parsimony. The lower AIC and BIC values for the NBINGARCH(1) model indicate it effectively balances goodness-of-fit with model simplicity, capturing the data's dynamics without overfitting.

Residual analysis using the Ljung–Box test confirms that the NBINGARCH(1) model adequately captures serial dependence, leaving no significant autocorrelation in the Pearson residuals. Additionally, the kernel density and normal Q-Q plots in Figure 4.3 suggest that residuals are approximately normally distributed, further supporting the model's adequacy. The normalized conditional randomized quantile residuals in Figure 4.5 also validate the model's fit, showing no systematic patterns.

Moreover, the analysis of parameter estimates for the Poisson, double Poisson, and negative binomial INGARCH models provides valuable insights into model selection and performance for overdispersed count time series data. Table 4.4 details the estimated parameters, their asymptotic standard errors, and model selection criteria, including AIC and BIC. The findings reveal that the INGARCH(1) model generally outperforms the more complex INGARCH(1,1) model, as evidenced by lower AIC and BIC values across all distributional assumptions. This indicates that adding an additional autoregressive component in the INGARCH(1,1) model does not significantly improve fit. Therefore, the INGARCH(1) model is preferable for capturing overdispersion while maintaining simplicity.

However, some limitations should be acknowledged. The selection of the dispersion parameter r through AIC/BIC optimization may not always capture the true underlying overdispersion in the data. Additionally, the NBINGARCH model assumes stationarity, which may not always hold for epidemiological data influenced by seasonal patterns or external factors.

5.7 On Model Comparison

The analysis of parameter estimates for the Poisson, double Poisson, and negative binomial INGARCH(p, q) models provides valuable insights into model selection and performance for overdispersed count time series data. Table 4.4 presents the estimated parameters along with their asymptotic standard errors, and model selection criteria, namely the Akaike information criterion (AIC) and Bayesian information criterion (BIC). Lower values of these criteria indicate better model performance.

The negative binomial model consistently achieves the lowest AIC and BIC values, which indicates its superior ability to account for overdispersion in the data. This is particularly evident in the INGARCH(1) model, where the negative binomial model outperforms both the Poisson and double Poisson models, with AIC and BIC values of 990.9 and 957.2, respectively. The Poisson model, with its restrictive variance assumption of equality to the mean, performs the worst, while the double Poisson model provides a moderate improvement by introducing an additional dispersion parameter. For the INGARCH(1,1) models, the results follow a similar trend, with the negative binomial model yielding the best fit, followed by the double Poisson and Poisson models. However, the relatively smaller improvements in AIC and BIC compared to the INGARCH(1) model suggest that the added complexity may not justify its use in practical applications. The estimated parameters reveal notable differences across the models. In particular, the estimates of $\hat{\alpha_1}$ and $\hat{\beta_1}$ for the Poisson and double Poisson INGARCH(1,1) models suggest stronger dependence structures, while the negative binomial models exhibit smaller autoregressive and moving average components. This indicates that the negative binomial model is better at capturing the inherent variability in the data without relying heavily on past observations.

In addition to AIC and BIC, the models are further assessed based on their ability to capture extreme values through the analysis of tail probabilities for residuals. Table 4.5 presents the observations X_t for which the upper probability $1 - F(X_t - 1, \hat{\alpha}, \hat{r}) \leq 0.02$, allowing for a comparative evaluation of tail probabilities across the Poisson, double Poisson, and negative binomial models. The results highlight distinct differences in the ability of these models to accommodate extreme observations in the data. Models that yield higher tail probabilities demonstrate a greater capacity to account for extreme variations in the data.

The Poisson model consistently yields very low tail probabilities, underscoring its limitations in capturing extreme values. This can be attributed to its inherent assumption of equidispersion, where the mean and variance are equal, making it unsuitable for overdispersed data. Consequently, the Poisson model tends to underestimate the likelihood of observing extreme values, which can lead to biased conclusions in practical applications. The double Poisson model, which introduces an additional dispersion parameter, demonstrates slight improvement over the Poisson model. However, it still struggles to fully accommodate the variability present in the data. While the model provides a better fit than the Poisson distribution, its performance in capturing extreme counts remains suboptimal, suggesting that the added flexibility is insufficient for highly overdispersed data.

In contrast, the negative binomial model consistently yields higher tail probabilities, effectively capturing the variability and overdispersion inherent in the dataset. This superior performance can be attributed to its more flexible variance structure, given by $r\lambda_t(1 + \lambda_t)$, which allows for greater dispersion compared to the Poisson and double Poisson models, whose conditional variances are λ_t and λ_t/γ_t, respectively. The negative binomial model's ability to better accommodate overdispersed data makes it a more appropriate choice for analyzing count data with potential extreme observations.

Chapter 6

Conclusion and Recommendations

In this final chapter, we summarise the main findings of the dissertation and underscore their implications for both theoretical and practical applications. We offer recommendations based on our insights to guide future research and improvements. The summary of the main findings is given in Section 6.1 while the recommendations are given in Section 6.2.

6.1 Conclusion

The findings of this dissertation provide significant insights into the modelling of overdispersed count time series data using the NBINGARCH(p, q) model. The proposed model effectively captures overdispersion and extreme values, offering a robust alternative to traditional Poisson and double Poisson models. The results demonstrate the superiority of the NBINGARCH model in terms of model fit and parameter estimation accuracy, as evidenced by lower AIC and BIC values and the ability to better capture the variability inherent in count data. The inclusion of the dispersion parameter r introduces greater flexibility, making the model well-suited for real-world applications, such as the analysis of syphilis data. The comprehensive methodology adopted in this study, encompassing model formulation, parameter estimation using MLE, CLS, and YW methods, as well as extensive simulation studies, ensures that the proposed approach is both theoretically sound and practically applicable. The performance evaluation using MADE confirms the efficiency of maximum likelihood estimation, particularly with increasing sample sizes.

Despite the model's strengths, several challenges and limitations were identified. The assumption of stationarity may not hold in all practical scenarios, and the estimation of the dispersion parameter r remains a complex issue. Additionally, further work is required to establish the geometric ergodicity of the model, which is crucial for deriving asymptotic properties and ensuring reliable inference.

In conclusion, the findings of this research contribute to the advancement of count time series modelling and provide a foundation for future exploration and development in this field.

6.2 Recommendations

Based on the findings of this dissertation, the following recommendations are proposed:

(1) **Model Refinement:** Future research should explore alternative parameter estimation methods that jointly estimate θ and r, potentially addressing the challenges associated with the discrete nature of r.

(2) **Stationarity Considerations:** Investigating extensions of the NBINGARCH model to account for non-stationary time series, including seasonal or external influences, would enhance its applicability to a wider range of real-world datasets.

(3) **Computational Improvements:** The implementation of more efficient algorithms for maximum likelihood estimation could improve computational feasibility for larger datasets and more complex models.

(4) **Asymptotic Properties:** Further research into the asymptotic distribution of the parameter estimates, leveraging geometric ergodicity or weak dependence concepts, could provide stronger theoretical foundations for statistical inference.

(5) **Application to Other Domains:** The applicability of the NBINGARCH model should be explored in other fields with overdispersed count time series data, such as meteorology, epidemiology, and insurance to validate its effectiveness across diverse datasets.

Appendix A

Data Set for the Real Data Example

In Section 4.2, the NBINGARCH model is applied to an overdispersed count time series of syphilis data. This is part of a data set given in the R software ZIM-package sourced from the CDC Morbidity and Mortality Weekly Report CDC MMWR. The data set consists of the weekly number of syphilis cases in Maryland, United States, from the first week of January 2007 to the first week of May 2010, giving a total of 209 observations.

Table A.1: Syphilis Cases Data: Month, Week, and Cases

Month	Week	Cases	Month	Week	Cases	Month	Week	Cases
Jan	1	5	Jun	24	5	Dec	57	3
Jan	2	6	July	25	5	Dec	58	4
Jan	3	7	July	26	7	Jan	59	2
Jan	4	6	July	27	4	Jan	50	4
Feb	5	3	July	28	6	Jan	51	7
Feb	6	0	Aug	29	4	Jan	52	1
Feb	7	6	Aug	30	7	Feb	53	6
Feb	8	3	Aug	31	9	Feb	54	4
Mar	9	0	Aug	32	2	Feb	55	0
Mar	10	0	Sept	33	10	Feb	56	3
Mar	11	0	Sept	34	9	Mar	57	5
Mar	12	2	Sept	35	5	Mar	58	3
April	13	11	Sept	36	2	Mar	59	5
April	14	4	Oct	37	6	Mar	60	0
April	15	15	Oct	38	3	April	61	0
April	16	6	Oct	39	10	April	62	4
May	17	4	Oct	40	4	April	63	5
May	18	7	Nov	41	4	April	64	3
May	19	1	Nov	42	2	May	65	8
May	20	5	Nov	43	6	May	66	4
Jun	21	0	Nov	44	7	May	67	12
Jun	22	2	Dec	45	5	May	68	0
Jun	23	2	Dec	46	0	Jun	69	6

Table A.2: Syphilis Cases Data: Month, Week, and Cases (continued)

Month	Week	Cases	Month	Week	Cases	Month	Week	Cases
Jun	70	9	Jun	117	5	May	164	0
Jun	71	0	Jun	118	9	Jun	165	5
Jun	72	4	Jun	119	4	Jun	166	6
July	73	4	Jun	120	7	Jun	167	5
July	74	4	Jul	121	0	Jun	168	0
July	75	8	Jul	122	7	Jul	169	0
July	76	5	Jul	123	0	Jul	170	0
Aug	77	6	Jul	124	0	Jul	171	5
Aug	78	0	Aug	125	0	Jul	172	4
Aug	79	3	Aug	126	0	Aug	173	3
Aug	80	7	Aug	127	0	Aug	174	0
Sept	81	7	Aug	138	3	Aug	175	0
Sept	82	3	Sept	139	0	Aug	176	1
Sept	83	3	Sept	130	0	Sept	177	0
Sept	84	6	Sept	131	4	Sept	178	0
Oct	85	5	Sept	132	4	Sept	178	3
Oct	86	6	Oct	133	0	Sept	180	4
Oct	87	0	Oct	134	4	Oct	181	3
Oct	88	5	Oct	135	5	Oct	182	7
Nov	89	5	Oct	136	6	Oct	183	7
Nov	90	6	Nov	137	11	Oct	184	4
Nov	91	2	Nov	138	0	Nov	185	9
Nov	92	0	Nov	139	0	Nov	186	3
Dec	93	3	Nov	140	9	Nov	187	0
Dec	94	1	Dec	141	5	Nov	188	0
Dec	95	3	Dec	142	3	Dec	189	0
Dec	96	5	Dec	143	4	Dec	190	10
Jan	97	9	Dec	144	0	Dec	191	0
Jan	98	0	Jan	145	1	Dec	192	0
Jan	99	0	Jan	146	5	Jan	193	3
Jan	100	1	Jan	147	1	Jan	194	0
Feb	101	0	Jan	148	2	Jan	195	0
Feb	102	5	Feb	149	6	Jan	196	0
Feb	103	0	Feb	150	7	Feb	197	6
Feb	104	0	Feb	151	3	Feb	198	5
Mar	105	1	Feb	152	0	Feb	199	0
Mar	106	4	Mar	153	4	Feb	200	6
Mar	107	0	Mar	154	4	Mar	201	0
Mar	108	5	Mar	155	7	Mar	202	2
April	109	0	Mar	156	2	Mar	203	4
April	110	0	April	157	1	Mar	204	0
April	111	0	April	168	3	April	205	3
April	112	0	April	169	5	April	206	0
May	113	10	April	160	0	April	207	1
May	114	8	May	161	3	April	208	2
May	115	3	May	162	0	May	209	5
May	116	2	May	163	2			

Appendix B

MATLAB Code: Simulation Study

In Section 4.1, a simulation study is conducted to evaluate the finite sample performances of the YW, CLS and ML estimates. For this, three set-ups are considered and they are:

(i) NBINGARCH(1) with parameters $(\alpha_0^0, \alpha_1^0, r)^T$=$(A1)(2, 0.3, 1)^T$ and $(A2)(4, 0.3, 2)^T$;

(ii) NBINGARCH(2) with $(\alpha_0^0, \alpha_1^0, \alpha_2^0, r)^T = (B1)(2, 0.4, 0.2, 1)^T$ and $(B2)(3, 0.3, 0.1, 2)^T$;

(iii) NBINGARCH(1,1) with $(\alpha_0^0, \alpha_1^0, \beta_1^0, r)^T = (C1)(2, 0.2, 0.4, 1)^T$ and $(C2)(3, 0.1, 0.3, 2)^T$.

We now present the MATLAB Code for this simulation study.

B.1 Table 4.1: NBINGARCH(1) Models

```
% Clear workspace and initialize global variables
clear all
global x n r com

% Define parameters
alpha=[2;0.3];    % true value of parameters
n=500; % sample size
s=200; % replication times
r = 1;      % Parameter r

% Define the log-likelihood function for the model
function y = g1(z, r)
    global X n r com
    y = -log(z(1) + z(2) * X(1:n-1)) * X(2:n)' + ...
        log(1 + z(1) + z(2) * X(1:n-1)) * (r + X(2:n))' - ...
        sum(com) + sum(log(factorial(X(2:n))));
end
```

```
% Set a random seed for reproducibility using a specific
random number stream
mystream = RandStream('mt19937ar', 'Seed', 0);
RandStream.setGlobalStream(mystream);

% Initialize arrays to store results
mle1 = zeros(1, s);  % Maximum Likelihood Estimator (MLE)
mle2 = zeros(1, s);
cls1 = zeros(1, s);  % Conditional Least Squares (CLS)
cls2 = zeros(1, s);
yw1 = zeros(1, s);   % Yule-Walker Estimator (YW)
yw2 = zeros(1, s);
err1 = zeros(s, 2);  % Error for YW
err2 = zeros(s, 2);  % Error for CLS
err3 = zeros(s, 2);  % Error for MLE

% Perform simulations for replication times and
generate initial value for X(1) from NBD
for time = 1:s
    X(1) = nbinrnd(r, 1 / (1 + alpha(1)));

% Simulate the process for t = 2 to n using the NBINGARCH(1) model
    for t = 2:n
X(t) = nbinrnd(r, 1 / (1 + alpha(1) + alpha(2) * X(t-1)));
com(t) = sum(log((1:X(t)) + r - 1));  % Compute combinatorial term
    end

% Yule-Walker estimator computation
    yw2(time) = (X(2:n) - mean(X)) * (X(1:n-1) - mean(X))' / ...
    sum((X(1:n) - mean(X)).^2) / r;  % Estimate Yule-Walker parameter
    yw1(time) = mean(X) * (1 - r * yw2(time)) / r;

% Conditional Least Squares (CLS) estimator computation
    cls = [n-1, sum(X(1:n-1)); sum(X(1:n-1)), sum(X(1:n-1).^2)];
    cls_vec = [sum(X(2:n)); X(2:n) * X(1:n-1)'] / r;
    cls1(time) = cls(1);
    cls2(time) = cls(2);
```

```
% Set up constraints and options for ML
    A = [-1, 0; 0, -1; 0, r];b = [0.001; 0.001; 1];
    options = optimset('Algorithm', 'active-set', 'Display', 'off');

    % Perform ML
    mle = fmincon(@g1, [cls(1); cls(2)], A, b, [],
    [], [], [], [], options);

    % Handle NaN values in MLE by replacing them with CLS estimates
    for i = 1:2
        if isnan(mle(i)) == 1
            mle(i) = cls(i);
        end
    end

% Store MLE results
    mle1(time) = mle(1);
    mle2(time) = mle(2);

% Calculate Mean Absolute Deviation Error (MADE)
    err1(time, :) = abs([yw1(time), yw2(time)] - alpha');
    err2(time, :) = abs([cls1(time), cls2(time)] - alpha');
    err3(time, :) = abs([mle1(time), mle2(time)] - alpha');
end

% Compute average results over all replications
average_results = [sum(yw1), sum(yw2); sum(err1, 1); sum(cls1), sum(cls2);
                   sum(err2, 1); sum(mle1), sum(mle2); sum(err3, 1)] / s;
disp('Average Results (YW, CLS, MLE and their Errors):')
disp(average_results)

% Print the results for the final run
disp('Final Run Results:')
disp('Maximum Likelihood Estimates (MLE):')
disp([mle1(end), mle2(end)])
disp('Yule-Walker Estimates (yw1, yw2):')
disp([yw1(end), yw2(end)])
disp('Conditional Least Squares Estimates (cls1, cls2):')
disp([cls1(end), cls2(end)])
```

B.2 Table 4.1: NBINGARCH(2) Models

```
% Clear workspace and initialize global variables
clear all
global x n r com

% Define parameters
alpha=[2;0.3;0.2]; % true value of parameters
n = 500;  % Sample size
s = 200;  % Number of replications
r = 1;    % Parameter r

% Define the log-likelihood function for the NBINGARCH(2) model
function y = g2(z, r)
    global X n r com
    y = -log(z(1) + z(2) * X(2:n-1) + z(3) * X(1:n-2)) * X(3:n)' + ...
        log(1 + z(1) + z(2) * X(2:n-1) + z(3)
        * X(1:n-2)) * (r + X(3:n))' - ...
        sum(com) + sum(log(factorial(X(3:n))));
end

% Set a random seed for reproducibility using a specific
random number stream
mystream = RandStream('mt19937ar', 'Seed', 0);
RandStream.setGlobalStream(mystream);

% Initialize arrays to store results
mle1 = zeros(1, s);  % Maximum Likelihood Estimator (MLE)
mle2 = zeros(1, s);
mle3 = zeros(1, s);
cls1 = zeros(1, s);  % Conditional Least Squares (CLS)
cls2 = zeros(1, s);
cls3 = zeros(1, s);
yw1 = zeros(1, s);   % Yule-Walker Estimator (YW)
yw2 = zeros(1, s);
yw3 = zeros(1, s);
err1 = zeros(s, 3);  % Error for YW
err2 = zeros(s, 3);  % Error for CLS
err3 = zeros(s, 3);  % Error for MLE
```

```
% Perform simulations for replication times and generate
initial values
for X(1) and X(2) from NBD
for time = 1:s
     X(1) = nbinrnd(r, 1 / (1 + alpha(1)));
    X(2) = nbinrnd(r, 1 / (1 + alpha(1) + alpha(2) * X(1)));
    for t = 3:n
        X(t) = nbinrnd(r, 1 / (1 + alpha(1) + alpha(2) * X(t-1) +
        alpha(3) * X(t-2)));
        com(t) = sum(log((1:X(t)) + r - 1));
    end

% Yule-Walker estimator computation
    rho1 = (X(2:n) - mean(X)) * (X(1:n-1) - mean(X))'
    / sum((X(1:n) - mean(X)).^2);
    rho2 = (X(3:n) - mean(X)) * (X(1:n-2) - mean(X))'
    / sum((X(1:n) - mean(X)).^2);
    yw = [1, rho1; rho1, 1] \ [rho1; rho2] / r;
    yw2(time) = yw(1);
    yw3(time) = yw(2);
    yw1(time) = mean(X) * (1 - r * yw2(time) - r * yw3(time)) / r;

% Conditional Least Squares (CLS) estimator computation
    cls_mat = [n-2, sum(X(2:n-1)),
    sum(X(1:n-2)); ...
                sum(X(2:n-1)), sum(X(2:n-1).^2), X(2:n-1)
                * X(1:n-2)'; ...
                sum(X(1:n-2)), X(2:n-1) * X(1:n-2)', sum(X(1:n-2).^2)];
    cls_rhs = [sum(X(3:n)); X(3:n) * X(2:n-1)'; X(3:n)
    * X(1:n-2)'] / r;
    cls = cls_mat \ cls_rhs;
    cls1(time) = cls(1);
    cls2(time) = cls(2);
    cls3(time) = cls(3);

% Set up constraints and options for ML
    A = [-1, 0, 0; 0, -1, 0; 0, 0, -1; 0, r, r];
    b = [0.001; 0.001; 0.001; 1];
    options = optimset('Algorithm', 'active-set', 'Display', 'off');
```

```
% Perform Maximum Likelihood Estimation (MLE)
    mle = fmincon(@g2, [cls(1); cls(2); cls(3)], A, b,
    [], [], [], [], [], options);

    % Handle NaN values in MLE by replacing them with CLS estimates
    for i = 1:3
        if isnan(mle(i)) == 1
            mle(i) = cls(i);
        end
    end

    % Store MLE results
    mle1(time) = mle(1);
    mle2(time) = mle(2);
    mle3(time) = mle(3);

% Calculate Mean Absolute Deviation Error (MADE)
    err1(time, :) = abs([yw1(time), yw2(time), yw3(time)] - alpha');
    err2(time, :) = abs([cls1(time), cls2(time), cls3(time)] - alpha');
    err3(time, :) = abs([mle1(time), mle2(time), mle3(time)] - alpha');
end

% Compute average results over all replications
average_results = [sum(yw1), sum(yw2), sum(yw3); sum(err1, 1); ...
                   sum(cls1), sum(cls2), sum(cls3); sum(err2, 1); ...
                   sum(mle1), sum(mle2), sum(mle3); sum(err3, 1)] / s;
disp('Average Results (YW, CLS, MLE and their Errors):')
disp(average_results)

% Print the results for the final run
disp('Final Run Results:')
disp('Maximum Likelihood Estimates (MLE):')
disp([mle1(end), mle2(end), mle3(end)])
disp('Yule-Walker Estimates (yw1, yw2, yw3):')
disp([yw1(end), yw2(end), yw3(end)])
disp('Conditional Least Squares Estimates (cls1, cls2, cls3):')
disp([cls1(end), cls2(end), cls3(end)])
```

B.3 Table 4.1: NBINGARCH(1,1) Models

```
% Clear workspace and initialize global variables
clear all
global x n r com

% Define parameters
alpha=[3;0.1;0.3];    %true value of parameters
n=100; %sample size
s=200; %replication times
r = 2;      % Parameter r

% Define the log-likelihood function for the model
function y = g3(z, r)
    global X n r com lambda
    y = -log(z(1) + z(2) * X(1:n-1) + z(3) * lambda(1:n-1))
    * X(2:n)' + ...
        log(1 + z(1) + z(2) * X(1:n-1) + z(3) * lambda(1:n-1))
        * (r + X(2:n))'
        - ... sum(com) + sum(log(factorial(X(2:n))));
end

% Set random seed for reproducibility
mystream = RandStream('mt19937ar', 'Seed', 0);
RandStream.setGlobalStream(mystream);

% Initialize arrays to store results
mle1 = zeros(1, s);   % Maximum Likelihood Estimator (MLE)
mle2 = zeros(1, s);
mle3 = zeros(1, s);
cls1 = zeros(1, s);   % Conditional Least Squares (CLS)
cls2 = zeros(1, s);
cls3 = zeros(1, s);
err1 = zeros(s, 3);   % Error for CLS
err2 = zeros(s, 3);   % Error for MLE
```

```
% Perform simulations for replication times and
initialize lambda(1) and X(1)
for time = 1:s
    lambda(1) = alpha(1);
    X(1) = nbinrnd(r, 1 / (1 + lambda(1)));
    for t = 2:n
        lambda(t) = alpha(1) + alpha(2) * X(t-1) + alpha(3)
        * lambda(t-1);
        X(t) = nbinrnd(r, 1 / (1 + lambda(t)));
        com(t) = sum(log((1:X(t)) + r - 1));
    end

 % Calculate the matrix for CLS
    p = floor(sqrt(n));
    for i = 1:n-p
        for j = 1:p
            mat(i, j) = X(p+i-j) - mean(X);
        end
    end
% Calculate residuals for CLS
    for t = p+1:n
        res(t) = X(t) -
        mean(X) - (X(t-1:-1:t-p) - mean(X)) * ((mat' * mat)
        \(mat' * (X(p+1:n) - mean(X))'));
    end

% Construct CLS matrix
    mat2 = [ones(1, n-p); X(p:n-1); res(p:n-1)]';
    clsab = (mat2' * mat2) \ (mat2' * X(p+1:n)');  % CLS estimates
    cls = [clsab(1)/r; (clsab(2) + clsab(3))/r; -clsab(3)];

% Store CLS results
    cls1(time) = cls(1);
    cls2(time) = cls(2);
    cls3(time) = cls(3);

% Set up constraints and options for ML
    A = [-1, 0, 0; 0, -1, 0; 0, 0, -1; 0, r, 1];
    b = [0.001; 0.001; 0.001; 1];
    options = optimset('Algorithm', 'active-set', 'Display', 'off');
```

```matlab
% Perform Maximum Likelihood Estimation (MLE)
    mle = fmincon(@g3, [cls(1); cls(2); cls(3)],
    A, b, [], [], [], [], [], options);

    % Handle NaN values in MLE by replacing them with CLS estimates
    for i = 1:3
        if isnan(mle(i)) == 1
            mle(i) = cls(i);
        end
    end

% Store MLE results
    mle1(time) = mle(1);
    mle2(time) = mle(2);
    mle3(time) = mle(3);

% Calculate Mean Absolute Deviation Error (MADE) for CLS and MLE
    err1(time, :) = abs([cls1(time), cls2(time), cls3(time)] - alpha');
    err2(time, :) = abs([mle1(time), mle2(time), mle3(time)] - alpha');
end

% Compute average results over all replications
average_results = [sum(cls1), sum(cls2), sum(cls3); sum(err1, 1); ...
                   sum(mle1), sum(mle2), sum(mle3); sum(err2, 1)] / s;

% Display average results (CLS, MLE, and their Errors)
disp('Average Results (CLS, MLE and their Errors):')
disp(average_results)

% Print the results for the final run
disp('Final Run Results:')
disp('Maximum Likelihood Estimates (MLE):')
disp([mle1(end), mle2(end), mle3(end)])
disp('Conditional Least Squares Estimates (cls1, cls2, cls3):')
disp([cls1(end), cls2(end), cls3(end)])
```

Appendix C

MATLAB Code: Real Data Example

In Section 4.2, the negative binomial INGARCH model is applied to an overdispersed count time series of syphilis data. This is part of a data set given in the R software ZIM-package sourced from the Centers for Disease Control (CDC) Morbidity and Mortality Weekly Report CDC MMWR. The data set consists of the weekly number of syphilis cases in Maryland, United States, from the first week of January 2007 to the first week of May 2010, giving a total of 209 observations. We now provide the MATLAB code for the computational aspect of this application.

C.1 Syphilis Data in MATLAB

```
% Define the data set
syphilis =
[5, 6, 7, 6, 3, 0, 6, 3, 0, 0, 0, 2, 11, 4, 15, 6, 4, 7, 1, 5,
0, 2, 2, 5, 5, 7, 4, 6, 4, 7, 9, 2, 10, 9, 5, 2, 6, 3, 10, 4,
4, 2, 6, 7, 5, 0, 3, 4, 2, 4,  7, 1, 6, 4, 0, 3, 5, 3, 5, 0,
0, 4, 5, 3, 8, 4, 12, 0, 6, 9, 0, 4, 4, 4, 8, 5, 6, 0, 3, 7,
7, 3, 3, 6, 5, 6, 0, 5, 5, 6, 2, 0, 3, 1, 3, 5, 9, 0, 0, 1,
0, 5, 0, 0, 1, 4, 0, 5, 0, 0, 0, 0, 0, 10, 8, 3, 2, 5, 9, 4,
7, 0, 7, 0, 0, 0, 0, 0, 3, 0, 0, 4, 4, 0, 4, 5, 6, 11, 0, 0,
9, 5, 3, 4, 0, 1, 5, 1, 2, 6,  7, 3, 0, 4, 4, 7, 2, 1, 3, 5,
0, 3, 0, 2, 0, 5, 6, 5, 0, 0,  0, 5, 4, 3, 0, 0, 1, 0, 0, 3,
4, 3, 7, 7, 4, 9, 3, 0, 0, 0, 6, 0, 2, 4, 0, 3, 0, 1, 2, 5];
```

C.2 Table 4.2: Summary statistics

```
% Load the data from the file
data = load('syphilis.txt');

% Compute summary statistics
mean_value = mean(data);
median_value = median(data);
std_dev = std(data);
min_value = min(data);
max_value = max(data);
range_value = range(data);
quartiles = quantile(data, [0.25, 0.5, 0.75]);
variance_value = var(data);
skewness_value = skewness(data);
kurtosis_value = kurtosis(data);

% Display the results
fprintf('Mean: %.2f\n', mean_value);
fprintf('Median: %.2f\n', median_value);
fprintf('Standard Deviation: %.2f\n', std_dev);
fprintf('Min: %.2f\n', min_value);
fprintf('Max: %.2f\n', max_value);
fprintf('Range: %.2f\n', range_value);
fprintf('Quartiles: %.2f, %.2f, %.2f\n', quartiles(1),
quartiles(2), quartiles(3));
fprintf('Variance: %.2f\n', variance_value);
fprintf('Skewness: %.2f\n', skewness_value);
fprintf('Kurtosis: %.2f\n', kurtosis_value);
```

C.3 Figure 4.1: Time plot, the ACF and PACF

```
% Load the data from the file
clear all;
load syphilis.txt;
x = syphilis';  % Transpose the syphilis data
n = length(x);  % Determine the number of observations

% Plot Time Plot, ACF, and PACF
figure;
```

```
% Time plot
subplot(3,1,1);
plot(x, '.-');
title('Time-Plot');
xlim([1, n]);
ylim([0, max(x) + 1]);
xlabel('Time');
ylabel('Count');

% Autocorrelation function (ACF) plot
subplot(3,1,2);
autocorr(x, 50);  % ACF up to lag 50
xlim([0.5, 50.5]);
ylim([-0.4, 0.4]);
ylabel('ACF');

% Partial Autocorrelation function (PACF) plot
subplot(3,1,3);
parcorr(x, 50);  % PACF up to lag 50
xlim([0.5, 50.5]);
ylim([-0.4, 0.4]);
ylabel('PACF');
```

C.4 Table 4.3 and 4.4: Parameter Estimates, AIC and BIC for Poisson and negative binomial

```
% Clear workspace and initialize global variables
clear all
global x n r com

% Load syphilis data and set sample size
load syphilis.txt
x = syphilis';
leng = size(x);
n = leng(2);
```

```
% Create matrices for Poisson estimation
for i = 2:n
    y = [1, x(i-1)]';
    clsmatrix(:,:,i) = y * y';      % Outer product of y
    clsvec(:,i) = x(i) * y;         % Vector for Poisson CLS
end

%% Poisson estimates using ML
% Set constraints and optimization options
r = 1;
A = [-1, 0; 0, -1; 0, 1];            % Constraints matrix
b = [0.001; 0.001; 1];               % Bound constraints
lb = [0.001; 0.001];                 % Lower bound
ub = [Inf; 1];                       % Upper bound
options = optimset('Algorithm', 'active-set', 'Display', 'off');

% Perform ML estimation
mle = fmincon(@gp1, sum(clsmatrix, 3)
\ sum(clsvec, 2), A, b, [], [], lb, ub, [], options);

% Compute lambda, first, and second derivative matrices
for i = 2:n
    la1(i) = mle(1) + mle(2) * x(i-1);   % Lambda (Poisson parameter)
    mat1(:,:,i) = (x(i) / la1(i) - 1)^2 * [1, x(i-1);
    x(i-1), x(i-1)^2];  % First derivative matrix
    mat2(:,:,i) = x(i) / la1(i)^2 * [1, x(i-1);
    x(i-1),x(i-1)^2];          % Second derivative matrix
end

% Calculate standard errors and likelihood
shat = sum(mat1, 3);          % Sum of first derivatives
dhat = sum(mat2, 3);          % Sum of second derivatives
se = (dhat \ shat) / dhat; % Standard errors

% Log-likelihood calculation for Poisson model
likelihood = log(la1(2:n)) * x(2:n)' - sum(la1(2:n))
- sum(log(factorial(x(2:n))));
```

```
% Calculate AIC and BIC for Poisson model
AIC = -2 * likelihood + 2 * 2;
BIC = -2 * likelihood + 2 * log(n - 2);

%% negative binomial estimates, AIC, and BIC
r = 1;

% Compute combinatorial terms for negative binomial model
for i = 2:n
    com(i) = sum(log((1:x(i)) + r - 1));
end

% Set up constraints and perform negative binomial ML estimation
A1 = [-1, 0; 0, -1; 0, r];
mlenb = fmincon(@gnb1, mle, A1, b, [], [], lb, ub, [], options);
la2(1) = mean(x);
for i = 2:n
    la2(i) = mlenb(1) + mlenb(2) * x(i-1);   % Lambda for NB model
end

% Moment estimator of r (for negative binomial model)
mler = sum(x(2:n)) / sum(la2(2:n));

% Compute first and second derivatives matrices for NB model
for i = 2:n
    mat21(:,:,i) = (x(i) / la2(i) - (mler + x(i)) / (1 + la2(i)))^2 *
    [1, x(i-1); x(i-1), x(i-1)^2];  % First derivative
    mat22(:,:,i) = (x(i) / la2(i)^2 - (mler + x(i)) / (1 + la2(i))^2)
    * [1, x(i-1); x(i-1), x(i-1)^2];  % Second derivative
end

% Calculate standard errors for NB model
shatnb = sum(mat21, 3);      % Sum of first derivatives
dhatnb = sum(mat22, 3);      % Sum of second derivatives
senb = (dhatnb \ shatnb) / dhatnb;  % Standard errors for NB model

% Compute combinatorial terms for NB likelihood
for i = 2:n
    com2(i) = sum(log((1:x(i)) + mler - 1));
end
```

```
% Log-likelihood calculation for negative binomial model
likelihood2 = log(la2(2:n)) * x(2:n)' - log(1 + la2(2:n)) *
(mler + x(2:n))' ...
             + sum(com2) - sum(log(factorial(x(2:n))));

% Calculate AIC and BIC for negative binomial model
AIC2 = -2 * likelihood2 + 3 * 2;   % 3 parameters for NBINGARCH(1)
BIC2 = -2 * likelihood2 + 3 * log(n - 3);

%% Display Results
disp('Poisson MLE, AIC, BIC:')
disp([mle', AIC, BIC])
disp('Poisson Standard Errors:')
disp(se)
disp('negative binomial MLE, Moment Estimator of r, AIC, BIC:')
disp([mlenb', mler, AIC2, BIC2])
disp('negative binomial Standard Errors:')
disp(senb)
```

C.5 Table 4.4: Parameter Estimates and AIC and BIC for Double Poisson Model

```
% Double Poisson Model
clear all;
global x n

% Load data
load syphilis.txt;
x = syphilis';  % Transpose the syphilis data
n = length(x);  % Number of observations

% Construct CLS matrix and vector
for i = 2:n
    y = [1, x(i-1)]';  % Create the design matrix
    clsmatrix(:,:,i) = y * y';  % CLS matrix
    clsvec(:,i) = x(i) * y;      % CLS vector
end
```

```
% Likelihood function for the Double Poisson Model (gp1)
function y = gp1(z)
    global x n
    y = -log(z(1) + z(2) * x(1:n-1)) * x(2:n)' +
    sum(z(1) + z(2) * x(1:n-1))
    + sum(log(factorial(x(2:n))));
end

% Likelihood function for INGARCH(1) (gp2)
function y = gp2(zz)
    global x n
    lambda1(1) = mean(x);
    for i = 2:n
        lambda1(i) = zz(1) + zz(2) * x(i-1) + zz(3) * lambda1(i-1);
    end
    y = -log(lambda1(2:n)) * x(2:n)' + sum(lambda1(2:n))
    + sum(log(factorial(x(2:n))));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INGARCH(1) Model

% Set constraints and optimization options
A1 = [-1, 0; 0, -1; 0, 1];
b1 = [0.001; 0.001; 1];
lb1 = [0.001; 0.001];
ub1 = [Inf; 1];
options = optimset('Algorithm', 'active-set', 'Display', 'off');

% Maximum Likelihood Estimation (MLE) for INGARCH(1)
initial_guess = sum(clsmatrix, 3) \ sum(clsvec, 2);
% Initial guess
mle1 = fmincon(@gp1, initial_guess, A1, b1, [], [], lb1, ub1,
[], options);  % MLE optimization

% Lambda estimation for INGARCH(1)
la1(1) = mean(x);
for i = 2:n
    la1(i) = mle1(1) + mle1(2) * x(i-1);
end
```

```
% Likelihood estimation for INGARCH(1)
hatg1 = 0.5 * (n - 1) / sum(-log(la1(2:n)) .* x(2:n) + la1(2:n)
- x(2:n)
+ log(x(2:n) .^ x(2:n)));

% Calculate vectors and matrices for standard errors
for i = 2:n
    ga1(i) = 0.5 / hatg1 + x(i) - la1(i) + log(la1(i)) * x(i)
    - log(x(i)^x(i));
    vec1(:,i) = [(x(i)/la1(i) - 1) * hatg1 * [1, x(i-1)], ga1(i)]';
    mat1(:,:,i) = vec1(:,i) * vec1(:,i)';
    vec2(:,i) = -(x(i)/la1(i) - 1) * [1; x(i-1)];
    mat2(:,:,i) = [x(i)/la1(i)^2 * hatg1 * [1, x(i-1);
    x(i-1), x(i-1)^2],
    vec2(:,i); vec2(:,i)', 0.5 / hatg1^2];
end

% Compute standard errors, likelihood, AIC, and BIC for INGARCH(1)
shat1 = sum(mat1, 3);
dhat1 = sum(mat2, 3);
se1 = (dhat1 \ shat1) / dhat1; % Standard errors
likelihood1 = hatg1 * log(la1(2:n)) * x(2:n)' - hatg1 * sum(la1(2:n))
- sum(log(factorial(x(2:n)))) + (1 - hatg1) * sum(log(x(2:n) .^ x(2:n))
- x(2:n))+ 0.5 * (n - 1) * log(hatg1);
AIC1 = -2 * likelihood1 + 3 * 2; % Akaike Information Criterion
BIC1 = -2 * likelihood1 + 3 * log(n - 3);
% Bayesian Information Criterion

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INGARCH(1,1) Model
p = floor(sqrt(n)); % Set p based on the number of observations

% Construct residual matrix for INGARCH(1,1)
for i = 1:n-p
    for j = 1:p
        m(i,j) = x(p+i-j) - mean(x);
    end
end
```

```
for t = p+1:n
    res(t) = x(t) - mean(x) - (x(t-1:-1:t-p) - mean(x)) * ((m' * m)
    \ (m' * (x(p+1:n) - mean(x))'));
end

% Initial guess for INGARCH(1,1)
m2 = [ones(1, n-p); x(p:n-1); res(p:n-1)]';
clsab = (m2' * m2) \ (m2' * x(p+1:n)');
clsp = [clsab(1); clsab(2) + clsab(3); -clsab(3)];

% Constraints for INGARCH(1,1)
A2 = [-1, 0, 0; 0, -1, -1; 0, 1, 1];
b2 = [0.001; 0.001; 1];
lb2 = [0.001; 0.001; 0.001];
ub2 = [Inf; 1; 1];

% MLE for INGARCH(1,1)
mle2 = fmincon(@gp2, clsp, A2, b2, [], [], lb2, ub2, [], options);

% Lambda estimation for INGARCH(1,1)
la2(1) = mean(x);
for i = 2:n
    la2(i) = mle2(1) + mle2(2) * x(i-1) + mle2(3) * la1(i-1);
end

% Likelihood estimation for INGARCH(1,1)
hatg2 = 0.5 * (n - 1) / sum(-log(la2(2:n)) .* x(2:n) + la2(2:n) - x(2:n)
+ log(x(2:n) .^ x(2:n)));

% Calculate vectors and matrices for standard errors
for i = 2:n
    dalpha0(i) = 1 + mle2(3) * dalpha0(i-1);
    dalpha1(i) = x(i-1) + mle2(3) * dalpha1(i-1);
    dbeta(i) = la2(i-1) + mle2(3) * dbeta(i-1);
    ga2(i) = 0.5 / hatg2 + x(i) - la2(i) + log(la2(i)) * x(i) -
    log(x(i)^x(i));
    vec3(:,i) = [(x(i)/la2(i) - 1) * hatg2 * [dalpha0(i), dalpha1(i),
    dbeta(i)], ga2(i)]';
    mat3(:,:,i) = vec3(:,i) * vec3(:,i)';
end
```

```
% Compute standard errors, likelihood, AIC, and BIC for INGARCH(1,1)
shat2 = sum(mat3, 3);
dhat2 = sum(mat4, 3);
se2 = (dhat2 \ shat2) / dhat2;
likelihood2 = hatg2 * log(la2(2:n)) * x(2:n)' - hatg2 * sum(la2(2:n))
- sum(log(factorial(x(2:n)))) + (1 - hatg2) * sum(log(x(2:n) .^ x(2:n))
- x(2:n))+ 0.5 * (n - 1) * log(hatg2);
AIC2 = -2 * likelihood2 + 3 * 3; % Akaike Information Criterion
BIC2 = -2 * likelihood2 + 3 * log(n - 3);
% Bayesian Information Criterion

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display Results
disp('INGARCH(1) Model:');
disp(['MLE1: ', num2str(mle1)]);
disp(['Standard Errors: ', num2str(diag(se1))]);
disp(['AIC1: ', num2str(AIC1)]);
disp(['BIC1: ', num2str(BIC1)]);

disp('INGARCH(1,1) Model:');
disp(['MLE2: ', num2str(mle2)]);
disp(['Standard Errors: ', num2str(diag(se2))]);
disp(['AIC2: ', num2str(AIC2)]);
disp(['BIC2: ', num2str(BIC2)]);
```

C.6 Table 4.3 and 4.4: Parameter Estimates and AIC and BIC for INGARCH(1,1) Model

```
clear all; % INGARCH(1,1) model setup
global x n r com
load syphilis.txt
x = syphilis';
leng = size(x); n = leng(2);
% Initialize CLS matrix and vector
for i = 2:n
    y = [1, x(i-1)]';
    clsmatrix(:,:,i) = y * y';
    clsvec(:,i) = x(i) * y;
    end
```

```
% Starting values estimation
p = floor(sqrt(n));
for i = 1:n-p
    for j = 1:p
        m(i,j) = x(p+i-j) - mean(x);
    end
end
for t = p+1:n
    res(t) = x(t) - mean(x) - (x(t-1:-1:t-p) - mean(x)) * ...
           ((m' * m) \ (m' * (x(p+1:n) - mean(x))'));
end
m2 = [ones(1,n-p); x(p:n-1); res(p:n-1)]';
clsab = (m2' * m2) \ (m2' * x(p+1:n)');

% Poisson estimation using fmincon
A = [-1,0,0; 0,-1,-1; 0,1,1]; b = [0.001;0.001;1];
lb = [0.001;0.001;0.001]; ub = [Inf;1;1];
options = optimset('Algorithm','active-set', 'Display','off');
clsp = [clsab(1); clsab(2)+clsab(3); -clsab(3)];
mle = fmincon(@gp2, clsp, A, b, [], [], lb, ub, [], options);

% Recursive calculation of lambda and derivatives
la1(1) = mean(x);
for i = 2:n
    la1(i) = mle(1) + mle(2) * x(i-1) + mle(3) * la1(i-1);
    dalpha0(i) = 1 + mle(3) * dalpha0(i-1);
    dalpha1(i) = x(i-1) + mle(3) * dalpha1(i-1);
    dbeta(i) = la1(i-1) + mle(3) * dbeta(i-1);

    % First and second derivative matrices
    vec1(:,i) = (x(i)/la1(i) - 1) * [dalpha0(i); dalpha1(i); dbeta(i)];
    mat1(:,:,i) = vec1(:,i) * vec1(:,i)';
    dalpha0beta(i) = dalpha0(i-1) + mle(3) * dalpha0beta(i-1);
    dalpha1beta(i) = dalpha1(i-1) + mle(3) * dalpha1beta(i-1);
    dbeta2(i) = 2 * dbeta(i-1) + mle(3) * dbeta2(i-1);
end
```

```
% Calculate log-likelihood, AIC, and BIC
shat = sum(mat1,3);
dhat = sum(mat2,3);
se = (dhat \ shat) / dhat;
likelihood = log(la1(2:n)) * x(2:n)' - sum(la1(2:n)) -
sum(log(factorial(x(2:n))));
AIC = -2 * likelihood + 3 * 2;
BIC = -2 * likelihood + 3 * log(n-3);

% negative binomial estimation
r = 1;
for i = 2:n
    com(i) = sum(log((1:x(i)) + r - 1));
end
A1 = [-1,0,0; 0,-r,-1; 0,r,1];
clsnb = [clsab(1)/r; (clsab(2)+clsab(3))/r; -clsab(3)];
mlenb = fmincon(@gnb2, clsnb, A1, b, [], [], lb, ub, [], options);

% negative binomial recursive calculations
la2(1) = mean(x);
for i = 2:n
    la2(i) = mlenb(1) + mlenb(2) * x(i-1) + mlenb(3) * la2(i-1);
end

% Calculate log-likelihood, AIC2, and BIC2
shatnb = sum(mat21,3); dhatnb = sum(mat22,3);
senb = (dhatnb \ shatnb) / dhatnb;
for i = 2:n
    com2(i) = sum(log((1:x(i)) + mler - 1));
end
likelihood2 = log(la2(2:n)) * x(2:n)' - log(1 + la2(2:n)) *
 (mler + x(2:n))' + sum(com2) - sum(log(factorial(x(2:n))));
AIC2 = -2 * likelihood2 + 4 * 2;
BIC2 = -2 * likelihood2 + 4 * log(n-4);

% Output results
[mle', AIC, BIC]se
inv(se)
[mlenb', mler, AIC2, BIC2]
senb
```

C.7 Figure 4.2: Syphilis Data (*) and Fitted Mean Values (-) from the NBINGARCH(1) Model

```
    load syphilis.txt
x=syphilis';
nn=size(x);n=nn(2);
alpha=[2.8,0.2];r=1;   % fitted mean values
plot(x,'*r');
hold on
plot(2:n,r*(alpha(1)+alpha(2)*x(1:n-1)),'.-')
ylabel('Counts');
xlabel('Time')
xlim([1,n]);ylim([0,max(x)]);
```

C.8 Figure 4.3: Pearson residual analysis.

```
    load syphilis.txt
x=syphilis';
nn=size(x);n=nn(2);n1=n-1;
alpha1=[2.8,0.2];r=1;
alpha2=[2.9,2];
m=15;
% Calculate Pearson residuals
pearson1 = (x(2:n) - r * (alpha1(1) + alpha1(2) * x(1:n-1))) ./ ...
           (r * (alpha1(1) + alpha1(2) * x(1:n-1)) .* (1 + alpha1(1)
            + alpha1(2) * x(1:n-1))).^(1/2);

pearson2 = (x(2:n) - alpha2(1) - alpha2(2) * x(1:n-1)) ./ ...
           (alpha2(1) + alpha2(2) * x(1:n-1)).^(1/2);

% Autocorrelation and Q-statistics calculation
for k = 1:m
    r1(k) = sum(pearson1(1:n1-k) .* pearson1(k+1:n1))
     / sum(pearson1.^2);
    r2(k) = sum(pearson2(1:n1-k) .* pearson2(k+1:n1))
     / sum(pearson2.^2);
    rr1(k) = r1(k)^2 / (n1-k);
    rr2(k) = r2(k)^2 / (n1-k);
end
```

```
Q = n1 * (n1 + 2) * [sum(rr1), sum(rr2)];

% Plot residual analysis results
subplot(2,2,1);
autocorr(pearson1, 25);
xlim([0.5, 25.5]); ylim([-0.2, 0.2]);
ylabel('ACF');

subplot(2,2,2);
parcorr(pearson1, 25);
xlim([0.5, 25.5]); ylim([-0.2, 0.2]);
ylabel('PACF');

subplot(2,2,3);
[f, xx] = ksdensity(pearson1, 'kernel', 'normal');
plot(xx, f);
xlabel('Residuals'); ylabel('Density');

subplot(2,2,4);
qqplot(pearson1);
xlabel('Standard normal quantiles');
ylabel('Quantiles of residuals');
```

C.9 Figure 4.4: Randomized quantile residual analysis

```
    load syphilis.txt
x=syphilis';
nn=size(x);n=nn(2);n1=n-1;
alpha=[2.8,0.2];r=1;
m=15;
rand('state', 180);  % Set random state for reproducibility

% Randomized quantile residual analysis
for t = 2:n
    p(t) = 1 / (1 + alpha(1) + alpha(2) * x(t-1));
    ub(t) = nbincdf(x(t), r, p(t));% Upper bound for residual
    lb(t) = nbincdf(x(t) - 1, r, p(t));  % Lower bound for residual
    residual(t) = norminv(lb(t) + (ub(t) - lb(t)) * rand(1));
    % Normalized residual
```

```
end

% Autocorrelation and Q-statistics calculation
for k = 1:m
    r(k) = sum(residual(1:n1-k) .* residual(k+1:n1)) / sum(residual.^2);
    rr(k) = r(k)^2 / (n1-k);
end
Q = n1 * (n1 + 2) * sum(rr);

% Plot residual analysis results
subplot(2,2,1);
autocorr(residual, 25);
xlim([0.5, 25.5]); ylim([-0.2, 0.2]);
ylabel('ACF');

subplot(2,2,2);
parcorr(residual, 25);
xlim([0.5, 25.5]); ylim([-0.2, 0.2]);
ylabel('PACF');

subplot(2,2,3);
[f, xx] = ksdensity(residual, 'kernel', 'normal');
plot(xx, f);
ylabel('Density'); xlabel('Residuals');

subplot(2,2,4);
qqplot(residual);
ylabel('Quantiles of residuals'); xlabel('Standard normal quantiles');
```

C.10 Figure 4.5: Randomized quantile residual against time

```
load syphilis.txt
x=syphilis';
nn=size(x);n=nn(2);
alpha=[2.8,0.2];r=1;
rand('state',180);
for t=2:n  % randomized quantile residual-time analysis
    p(t)=1/(1+alpha(1)+alpha(2)*x(t-1));
    ub(t)=nbincdf(x(t),r,p(t));
    lb(t)=nbincdf(x(t)-1,r,p(t));
    residual(t)=norminv(lb(t)+(ub(t)-lb(t))*rand(1));
end
% Plot the time series of residuals
plot(2:n,residual(2:n),'*')
title('Plot of residuals against time');
xlim([2,n]);
xlabel('Time');
ylabel('Residuals');
```

C.11 Table 4.5: Tail probabilities for potential extreme residuals

```
load syphilis.txt
x=syphilis';
nn=size(x);n=nn(2);
alpha1=[2.8,0.2];r=1; % NB
alpha2=[2.9,2]; % Poisson
gamma=0.3175; % double Poisson
% Loop through the data and calculate probabilities
for t = 2:n

% negative binomial model
    p1(t) = 1 / (1 + alpha1(1) + alpha1(2) * x(t-1));
    lb1(t) = nbincdf(x(t)-1, r, p1(t));  % Lower bound for NB model

% Poisson model
    lam(t) = alpha2(1) + alpha2(2) * x(t-1);
```

```
    lb2(t) = poisscdf(x(t)-1, lam(t));    % Lower bound for Poisson model

% Double Poisson model
    if x(t) == 0
        lb3(t) = 0;  % Special case for x(t) = 0
    elseif x(t) == 1
        lb3(t) = gamma^(1/2) * exp(-lam(t)*gamma)/ ...
                 (1 + (1-gamma)/(12*lam(t)*gamma)*(1 + 1/
                 (lam(t)*gamma)));
    else
        % Calculate PMF for Double Poisson
        pmf(1) = gamma^(1/2) * exp(-lam(t) * gamma);
        for i = 1:139
            pmf(i+1) = gamma^(1/2) * exp(-lam(t)*gamma - i) * ...
                       (i^i/factorial(i))
                        *(exp(1)*lam(t))^(gamma*i) / ...
                       (i^(gamma*i));
        end
        lb3(t) = sum(pmf(1:x(t))) / sum(pmf);
% Lower bound for Double Poisson
    end
end

% Find indices where probabilities are less than or equal to 0.02
index1 = find(1 - lb1 <= 0.02);
index2 = find(1 - lb2 <= 0.02);
index3 = find(1 - lb3 <= 0.02);

% Display results
1 - lb1(index1)
1 - lb2(index2)
1 - lb3(index3)
1 - lb2(35)
```

References

[1] Alzaid, A., Al-osh, M. (2019), *Modeling Integer-Valued Time Series Using negative binomial Autoregressive Moving Average Models*, Communications in Statistics–Theory and Methods **48**(7) 1632–1653.

[2] Baltimore City Health Department. (2016), *Healthy Baltimore 2015 Interim Status Report*, Baltimore City Health Department. Available at: http://health.baltimorecity.gov/.

[3] Benjamin, M. A., Rigby, R. A., and Stasinopoulos, D. M. (2003), *Generalized Autoregressive Moving Average Models.* Journal of the American Statistical Association **98**(461), 214–223.

[4] Billingsley, P. (1961), Statistical Inference for Markov Processes. University of Chicago Press, Chicago.

[5] Bollerslev, T. (1986), *Generalized Autoregressive Conditional Heteroskedasticity.* Journal of Econometrics **31**(3) 307–327.

[6] Box, P.G.E, Jenkins, R. (2015), *Time Series Analysis: Forecasting and Control.* Wiley Series in Probability and Statistics. **5th Edition**.

[7] Brockwell, P. J., and Davis, R. A. (2016), *Introduction to Time Series and Forecasting.* Springer.

[8] Brockwell, P. J., Davis, R. A. (1987), Time Series: Theory and Methods. New York: Springer Verlag.

[9] Bu, R., McCabe, B., Hadri, K. (2008), Maximum likelihood estimation of higher-order integer-valued autoregressive processes. Journal of Time Series Analysis **29**(6), 973–994.

[10] Cowpertwait, P. S. P., and Metcalfe, A. V. (2019), *Introductory Time Series with R (2nd Edition).* Cham, Switzerland: Springer International Publishing.

[11] Centers for Disease Control and Prevention. (2017), *Sexually Transmitted Disease Surveillance 2016.* U.S. Department of Health and Human Services.

[12] Davis, R. A., Wu, R. (2009), *A negative binomial Model for Time Series of Counts.* Biometrika **96** 735 – 49.

[13] Doukhan, P., Latour, A., Oraichi, D. (2006), *A Simple Integer-Valued Bilinear Time Series Model, Advances in Applied Probability* **38** 559–578.

[14] Drost, F., van den Akker, R., Werker, B. (2008), *Note on Integer-Valued Bilinear Time Series Models, Statistics and Probability Letters* **78** 992–996.

[15] Drost, F.C., van den Akker, R.,Werker, B.J.M. (2009), Efficient estimation of autoregression parameters and innovation distributions for semiparametric integer-valued AR(p) models. Journal of the Royal Statistical Society, Series B **7**1(2), 467–485.

[16] Dunn, P. K., Smyth, G. K. (1996), *Randomized quantile residuals. Journal of Computational and Graphical Statistics* **5**, 236–44.

[17] Efron, B. (1986), *Double Exponential Families and their use in Generalized Linear Regression*. Journal of the American Statistical Association **81** 709–721.

[18] Engle, R. F. (1982), *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*. Econometrica **50**(4) 987–1008.

[19] Ferland, R., Latour, A., and Oraichi, D. (2006), *Integer-Valued GARCH Process. Journal of Time Series Analysis* **27**(6) 923–942.

[20] Fokianos, K., and Tjøstheim, D. (2013), *Log-linear Poisson Autoregression*. Journal of Time Series Analysis **34**(3) 306–325.

[21] Fokianos, K., and Fried, R. (2010), *Interventions in INGARCH Processes*. Journal of Time Series Analysis **31** 210–25.

[22] Fong, P. W., Li, W. K., Yau, C. W. and Wong, C. S. (2007), *On a Mixture Vector Autoregressive Model*. Canadian Journal of Statistics **35** 135–50.

[23] Franke, J. (2010), *Weak Dependence of Functional INGARCH Processes*. Working paper, University of Kaiserslautern.

[24] Freeland, R., McCabe, B. (2004), *Forecasting Discrete Valued Low Count Time Series*, International Journal of Forecasting **20** 427–434.

[25] Grahramani, M., and Thavaneswaran, A. (2009), *On some Properties of Autoregressive Conditional Poisson (ACP) Models*. Economic Letters **105** 273–275.

[26] Goldberg, S. (1958), *Introduction to Difference Equations*. New York: Wiley.

[27] Heinen, A. (2003), *Modelling Time Series Count Data: An Autoregressive Conditional Poisson Model*. New Zealand Journal of Mathematics **32** 145–164.

[28] Jacob, P., Lewis, P. (1978), *Discrete-Time Series Generated by Mixtures, Correlational and Runs Properties*, Journal of Royal Statistics Society **40** 94–105.

[29] Kachour, M., Yao, J. (2009), *First-order Rounded Integer-Valued Autoregressive Process*, Journal of Time Series Analysis **30**(4) 417–448.

[30] Karlis, D., and Ntzoufras, I. (2009), *Bayesian Modelling of Moving Average Processes using Integer-Valued Autoregressive Models*. Computational Statistics and Data Analysis **53**(12) 4049–4061. doi: 10.1016/j.csda.2009.06.001.

[31] Karlsen, H., Tjøstheim, D. (1988), *Consistent estimates for the NEAR(2) and NLAR(2) time-series models. Journal of the Royal Statistical Society: Series B (Methodological)* **50**:313–320.

[32] Lee, J., and McLachlan, G. J. (2011), *On the Fitting of negative binomial Integer-Valued GARCH models*. Computational Statistics and Data Analysis **55(2)** 930–943.

[33] Li, M., Zheng, H., Zhang, H., and Zhu, S. (2006), *Count Data Modeling Based on GARCH Processes: An Application to Stock Market Volatility*. Applied Mathematics and Computation **181**(1) 100–113.

[34] Lawless, J. F. (1987), *negative binomial and mixed Poisson regression*. Canadian Journal of Statistics **15**, 209–225.

[35] Macdonald, I. L., and Zucchini, W. (1997), *Hidden Markov and Other Models for Discrete-Valued Time Series*: Chapman and Hall, Boca Rotou, Louisiana.

[36] Mann, H.B., and Wald, A. (1947), *Sequential Tests for the Case of Normal Means*. The Annals of Mathematical Statistics **18**(1) 1–13. doi:10.1214/aoms/1177730491.

[37] Maryland Department of Health and Mental Hygiene. (2012), *Provider Alert: Increasing Rates of Syphilis in Maryland*. Infectious Disease and Environmental Health Administration.

[38] Perumean-Chaney, S., Morgan, C., and McDowall, D. (2013), *Zero-Inflated and Overdispersed: What´s one to do?* Journal of Statistical Computation and Simulation **83**(9):1671–1683.

[39] Ristic, C. M. M., Nasti, C, A.S. and Bakouch, H.S. (2012), Estimation in an integer-valued autoregressive process with negative binomial marginals (NBINAR (1)). CCommunications in Statistics - Theory and Methods, **41** 606–618.

[40] Ristic, C. M. M., Nasti, C, A.S. and Bakouch, H.S. (2009), A new geometric first-order integer-valued autoregressive (NGINAR (1)) process.Journal of Statistical Planning and Inference **139** 2218–2226.

[41] Shumway, R. H., and Stoffer, D. S. (2017), *Time Series Analysis and its Applications: with R Examples*. Springer, New York.

[42] Shmueli, G., Minka, T. and Kadane, J. (2005), *A useful Distribution for Fitting Discrete Data: Revival of the Conway-Maxwell-Poisson Distribution.* Applied Statistics **54** 127–142.

[43] Steel, F., van Harn, K. (1979), *Discrete Analogs of Self-Decomposability and Stability*, The Annals of Probability **7**(5) 893–899.

[44] Silva, I., Silva, M. E. (2006), *Asymptotic distribution of the Yule-Walker estimator for INAR(p) processes*. Statistics and Probability Letters **76**:1655–1663.

[45] Weiß, C. H. (2008), *Thinning Operations for Modeling Time Series of Counts – a Survey*. Advances in Statistical Analysis **92** 319–341.

[46] Weiß, C. H. (2009), *Modelling Time Series of Counts with Overdispersion, Statistical Methods, and Applications* **18** 507–519.

[47] White, G. C., and Bennetts, R. E. (1996). *Analysis of frequency count data using the negative binomial distribution.* **Ecology**, **77**, 2549–2557. Ecological Society of America, Washington, D.C.

[48] Weiss, C. H. (2018), *An Introduction to Discrete-Valued Time Series*. John Wiley and Sons.

[49] Zeger, S. L. (1988), *A Regression Model for Time Series of Counts.* **Biometrika**, **75**, 621–629.

[50] Zheng, Y., and Wang, L. (2018), *Modeling and Forecasting the Number of Forest Fires in China using a negative binomial INGARCH Model.* Environmental Modelling and Software **103** 94–101.

[51] Zhu, F., and Li, Q. (2009), *Moment and Bayesian Estimation of Parameters in the INGARCH (1,1) Model (in Chinese)*. Journal of Jilin University (Science Edition) **47** 899–902.

[52] Zhu, F., Li, Q. and Wang, D. (2010), *A Mixture Integer-Valued ARCH Model*. Journal of Statistical Planning and Inference **140** 2025–36.

[53] Zhu, F. (2010), *A Negative Binomial Integer-Valued GARCH Model*. Journal of Time Series Analysis **31**(6), 451–466. `https://doi.org/10.1111/j.1467-9892.2010.00674.x`.

[54] Zhu, F. (2012), *Modeling Time Series of Counts with COM-Poisson INGARCH Models*. Mathematical and Computer Modelling **56** 191–203.

[55] Zhu, F. (2011), *A negative binomial integer-valued GARCH model.* **Journal of Time Series Analysis**, **32**(1), 54–67.

[56] Zhu, J. (2019), *Modeling Count Time Series: A Review Communications in Statistics-Theory and Methods* **48(16)** 4024–4041.

[57] Zhu, R., and Joe, H. (2010), *Negative binomial time series models based on expectation thinning operators.* **Journal of Statistical Planning and Inference**, **140**, 1874–1888.