Strathmore UNIVERSITY

SU+ @ Strathmore University Library

Electronic Theses and Dissertations

2022

Modeling of count data with an informative time component in the presence of overdispersion.

Owiti, Levi Alfred Orero
Strathmore Institute of Mathematical Sciences
Strathmore University

Recommended Citation

Owiti, L. A. O. (2022). *Modeling of count data with an informative time component in the presence of overdispersion* [Strathmore University]. http://hdl.handle.net/11071/13173

Follow this and additional works at: http://hdl.handle.net/11071/13173

This work is availed for free and open access by Strathmore University Library.
It has been accepted for digital distribution by an authorized administrator of SU+ @Strathmore University.
For more information, please contact library@strathmore.edu

Modeling of Count Data with an Informative Time Component in the Presence of Overdispersion

Owiti, Levi Alfred Orero

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Statistical Science at Strathmore University

Institute of Mathematical Sciences

Strathmore University

Nairobi, Kenya

October 2022

This thesis is available for Library use through open access on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgment.

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University.

Name: **Owiti, Levi Alfred Orero**

Signature: 

Date: August 22, 2022

Approval

The thesis of Owiti, Levi Alfred Orero was reviewed and approved by the following:

Dr. Evans Omondi

Supervisor,

Institute of Mathematical Sciences, Strathmore University.

Professor Benard Omolo

Supervisor,

Institute of Mathematical Sciences, Strathmore University.

Dr. Godfrey Madigu

Dean,

Institute of Mathematical Sciences, Strathmore University.

Dr. Bernard Shibwabo

Director,

Office of Graduate Studies, Strathmore University.

Abstract

In real-world count data, several methods have been applied to handle the common problem of overdispersion. However, these methods have not comprehensively considered unique features that may exist in the data. This study sought to address robust statistical modeling of count response data that contains temporal features. The study proposed a Bayesian Negative Binomial model that will handle overdispersion while taking into account the temporal features of the data. Two count data models were compared and extended to incorporate an informative time component. To test the various models, this study conducted simulation studies under specified parameters to examine how the models behave under certain conditions. The data generation mechanism ensured the simulated data had seasonality as is with the real-world data on fire frequency, temperature, and rainfall. Further, the study examined the effect of the additional components on prediction intervals of the simulation studies for the different count models. The introduction of Bayesian techniques into the modeling was intended to create more accurate prediction intervals that take account of the prior distribution of the data. The Bayesian Negative Binomial model was better than the Negative Binomial model in terms of model bias. When validated on real data to confirm its effectiveness, the Bayesian model had better MASE and the prediction intervals enveloped the actual data in the testing dataset of fires in Kenya between the year 2000 and 2018.

Keywords: overdispersion, Bayesian, Negative Binomial distribution, count data, informative component, spatiotemporal data

Table of contents

List of figures

List of tables

List of abbreviations

ML	Maximum Likelihood	GLM	Generalized Linear Model
NASA	National Aeronautics and Space Administration	EOSDIS	Earth Observing System Data and Information System
GIS	Geospatial Information Systems	FIRMS	Fire Information for Resource Management System
NRT	Near Real-Time	MODIS	Moderate Resolution Imaging Spectroradiometer
VIIRS	Visible Infrared Imaging Radiometer Suite	PIs	Prediction Intervals
BCMP	Bayesian Conway Maxwell-Poisson	CMP	Conway-Maxwell Poisson
GAM	Generalized additive model	MMNPP	Markov-Modulated Poisson Process
MaxEnt	Maximum entropy	GCM	Generalised Climate Model
BNB	Bayesian Negative Binomial Model	NB	Negative Binomial Model
RMSE	Root Mean Square Error	MASE	Mean Absolute Scale Error
BPMCM	Bayesian Poisson Markov Chain Monte Carlo Model	BNBMCM	Bayesian Negative Binomial Markov Chain Monte Carlo Model

Acknowledgement

First and foremost, I have to thank my research supervisors, Dr. Evans Omondi, and Prof. Benard Omolo. Without their assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished. I would like to thank you very much for your support and understanding over these past months. I would also like to thank my supervisor at ICRAF, Lisa Fuchs for her understanding during this busy period, influence in learning how to write well in research, and funding this work. I also acknowledge my senior colleague, Lalisa Duguma for encouraging me to explore this research topic. Finally, I like to thank my family, for always being there when I need them. Many life times will be required to express my gratitude. Thank you.

Dedication

This work is dedicated to God Almighty for giving me wisdom and good health.

Chapter 1

Introduction

1.1 Background to the study

Within the category of discrete response regression models exists a subset of models known as count response models. These models are the best for data that consists of any discrete response of counts. Count response can include the number of goals scored at a football match, number of children born, or the number of days that a patient spends in the hospital (Hilbe, 2011).

The main types of count models include Poisson, Negative Binomial, Zero-Inflated, Zero-Truncated, Hurdle, and Random-effects count models. In literature, there exist many variations of these models that depend on the modeling problem at hand but are all said to be based upon a count process. The Poisson regression is thus expressed as the standard count model, upon which other models of these types are based. Often, the counts are right-skewed and their variance increases with the mean of the distribution from which they are derived (Hilbe, 2011).

The Poisson distribution is unique because of the relationship of the mean to its variance. They are equal, a term which is known as *equidispersion*. This is rarely found in real data, including the data used in this study. This indicates overdispersion of the data and thus, another type of count model is recommended (Hilbe, 2011).

In the presence of overdispersion, the maximum likelihood estimators of model parameters may be consistent, convergent in probability to the values of the parameters but have standard errors that are too small. Because we know that for Poisson and Binomial distributions the variance is a function of the mean, we apply extensions of the Poisson generalized linear model (GLM) and include an extra parameter that will account for overdispersion.

1.2 Example of count data

Thermal activity on earth is detected by the National Aeronautics and Space Administration (NASA) through NASA's Earth Observing System Data and Information System (EOSDIS) which is part of the Earth Science Data program (Ramapriyan et al., 2010). By integrating Geospatial Information Systems (GIS) and remote sensing technologies, NASA delivers satellite data on hotspot/fire locations around the world. NASA's Fire Information for Resource Management System (FIRMS) provides Near Real-Time (NRT) fire/thermal anomaly data within 3 hours of satellite observation from NASA's Moderate Resolution Imaging Spectroradiometer (MODIS) aboard the Terra and Aqua satellites and NASA's Visible Infrared Imaging Radiometer Suite (VIIRS) aboard the joint NASA/NOAA Suomi National Polar-orbiting Partnership (Suomi NPP) and NOAA-20 satellites (NASA, 2022). The MODIS and VIIRS active fire/thermal anomaly data may be from fire, hot smoke, agriculture, or other sources (NASA, 2022). Each active thermal/fire anomaly location captured by MODIS represents the middle of a 1km pixel that is noted by the algorithm as containing one or more fires within it (NASA, 2022). By aggregating these data by month, research is possible on the frequency of fires that occur in a specific geographical region (Figure 1.1). Considering weather changes and climate patterns, fire regimes vary and the data shows overdispersion, meaning it is important to consider various techniques to handle this feature. In this research, tested models from simulation studies will be applied to the real-world fire frequency data to evaluate performance at a larger scale.

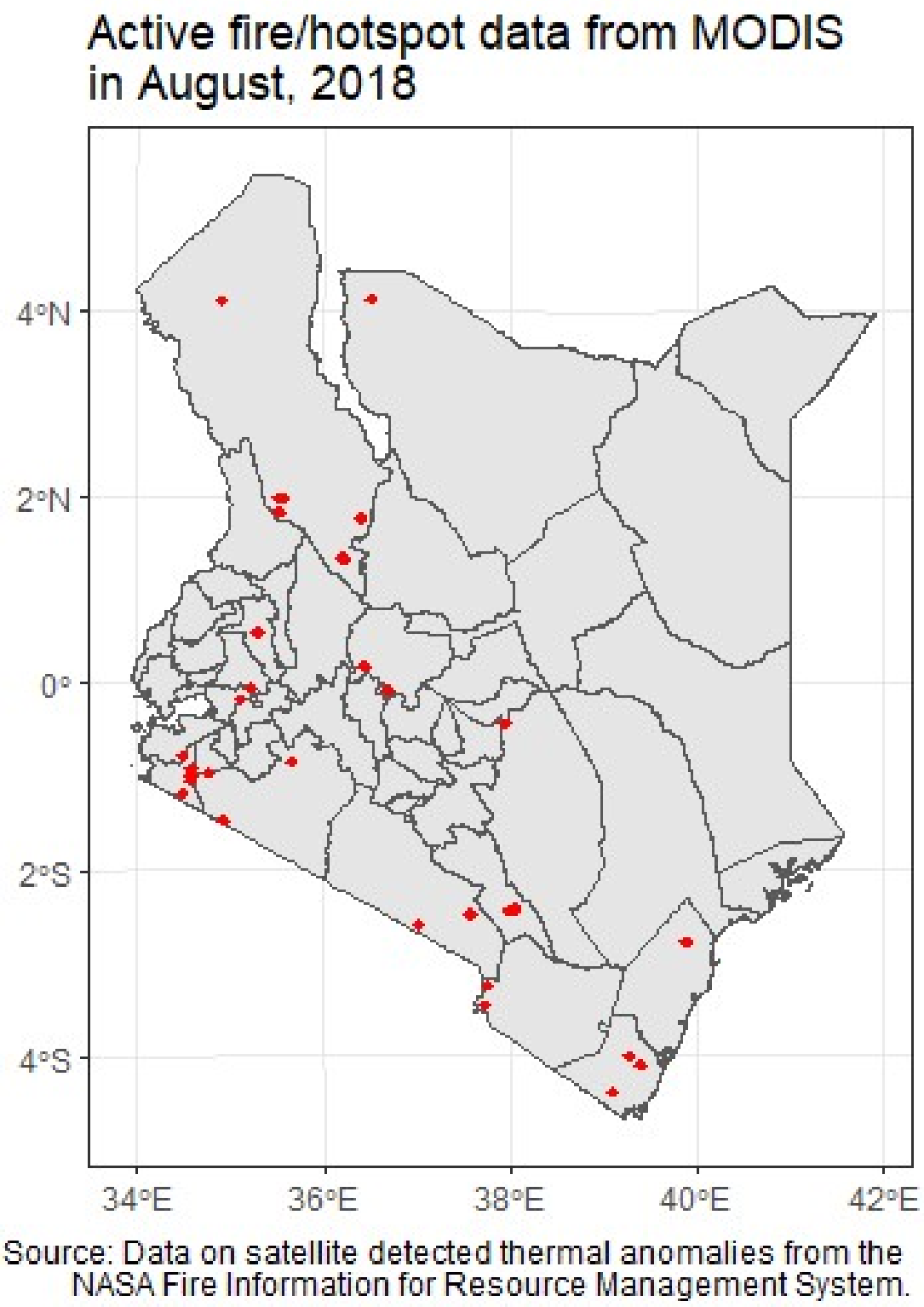

Figure 1.1: Active fire hotspots in Kenya as captured by satellite in August 2018. Source: NASA Fire Information for Resource Management System.

1.3 Statement of the problem

Several methods have been applied to handle overdispersion in count data in various fields (Huang and Kim, 2021; Lindén and Mäntyniemi, 2011; Payne et al., 2017; Trocóniz et al., 2009). Unique features in the data, such as Markov patterns (Trocóniz et al., 2009) or the ability of a model to handle overdispersion and underdispersion have been explored (Huang and Kim, 2021). Unlike these kinds of data, minimal research work has been done on the statistical modeling of count response data that contains spatial and temporal features. Furthermore, there is no comprehensive solution on how best to account for and consider overdispersion in count regression models (Hardin et al., 2007; Hayat and Higgins, 2014; Hilbe, 2011; Xia et al., 2012). In this study, a Bayesian Negative Binomial model for count data was formulated and extended to incorporate an informative time component. In addition, the effect of the prior mean component on posterior prediction intervals of the simulated studies was examined for the Bayesian model. Moreover, the Bayesian Negative Binomial model was applied real data to prove its effectiveness in the prediction of the number of future monthly and cumulative fires that are attributable to weather variables.

1.4 Objectives of the study

1.4.1 General objective

The main objective of this work was to model overdispersed count data that contains time and spatial features.

1.4.2 Specific objectives

1. To develop a statistical model to describe the behavior of count data with temporal explanatory variables, and to carry out an extensive statistical analysis of the model to gain the understanding of the model with varying parameter estimates.

2. To construct prediction intervals (PIs) for discrete-valued, count random variables derived from over-dispersed count models

3. To carry out statistical simulations of the models and establish the optimal set of parameters for overdispersed count data.

1.5 Scope of the study

While the long-term negative impacts of fires on climate have well been documented, the development of fire prediction models in Kenya is poorly understood. This study evaluated several count models and proposed an optimal prediction model based on historical data for Kenya from November 2000 to December 2018 (218 months).

The scope of the study was limited to historical geospatial and raster data on climate variables that are available on WorldClim (Fick and Hijmans, 2017). Further, the development of these models was limited by computational capability and only involved the use of simulated data bound by the distributions suggested by the real data.

1.6 Significance of the study

Liu et al., (2019) stated that fires are a danger as they can affect public safety and ecosystems when frequent (Liu et al., 2019). This study can be applied as technical advice for fire risk prediction considering how temperature and rainfall affect fire frequency. This then affects policymaking around fire management.

In a warming climate, robust methods in the prediction of fires are important amid variability in factors and unpredictability that surrounds fire occurrence (Lima et al., 2018). This study added to the literature that builds and optimizes prediction intervals to build the accuracy of predictive models based on simulated data and validated by real-world data.

In Kenya, this study will pave way for geographically specific models to predict fires in forests and reserves that are home to animals and plants, which in turn affects tourism.

1.7 Limitations of the study

The FIRMS data included thermal anomalies that may be due to controlled human intervention. These are usually difficult to predict and may not be fully accounted for in these analyses. Further, the data was context-specific to Kenya and it may not be applicable in other parts of the world. In the simulation process, elements of the techniques were subjective and thus might not apply in some instances but the study provided ways to mitigate against this. Despite this, it is a good step in evaluating accurate methods that can inform stakeholders on policies required during fire frequent months of the year.

1.8 Thesis outline

In this thesis, the research was organized as follows: In Chapter 2, the study included a literature review based on count models, handling overdispersion, count models on fire data, construction of prediction intervals, and gaps identified in the literature. In Chapter 3, the study explained the data sources and features of available data. Further, the existing models to be compared was extended to the proposed form of the Bayesian Negative Binomial model. The study outlined how the simulation process works and the parameters and software that was used. In Chapter 4, the study presented the descriptive statistics of the real data, simulation study results and the results of the models applied on the real data. In Chapter 5, the work discussed the results, outlined what they mean and why they matter. Finally,we conclude the work with a summary and the relevance of the work and the recommendations for future work.

Chapter 2

Literature review

2.1 Introduction

The Poisson process is widely used to simulate unexplained stochastic fluctuation around the model expectation (for example, a regression line) in models for discrete count data, assuming that the likelihood of detecting the next count event is constant in time or space for each unit in the sample (Lindén and Mäntyniemi, 2011). However, presence of overdispersion in data has led to extensions of count models to handle unique features in the data.

2.2 Applications in fire management systems

Considered dangerous and devastating, wildfires are complex because they are difficult to predict, difficult to put out, and result in huge financial, environmental, and societal losses (Sayad et al., 2019). Important to note is that fires, resulting from human or natural causes will create thermal anomalies that are picked up by earth's satellites (NASA, 2022). Specifically, humans clearing farmland or disposing of waste often create huge fires that are random and difficult to forecast. However, due to variations in rainfall and temperature throughout the year, some months of the year experience a lot of fires (Lima et al., 2018).

By the start of this millennium, researchers published their research on the interconnection between climate change, forest fires, and the impact on forests (Flannigan et al., 2000). Flannigan et al.(2000) report that most climate models (GCMs) were the best techniques to predict the effect of the changes in the future climate on fire regimes. The authors added that variables such as precipitation, wind, cloudiness, etc, will also be altered in the current global warming (Flannigan et al., 2000). Since then, several methods have been applied in

the quest to predict fire occurrences in various parts of the world although they are limited to specific areas (Kwak et al., 2012; Lim et al., 2019; Lima et al., 2018).

2.2.1 MODIS Fire Products

NASA's Fire Information for Resource Management System (FIRMS) has undergone several iterations in improving its fire products used by scientists in many fields (Fornacca et al., 2017; Giglio et al., 2016; NASA, 2022). With specific and detailed country-level datasets, researchers can analyze these data and cross-reference it with other climatic variables (Cai et al., 2017; Lim et al., 2019; Schroeder et al., 2014; Wooster et al., 2005). Fire data from MODIS has been used in many studies in research on fire frequency and distribution all over the world (Cai et al., 2017; Fornacca et al., 2017; Giglio et al., 2016; Lim et al., 2019; Wooster et al., 2005).

2.2.2 Rainfall and Temperature

Rainfall and temperature are some key variables that affect the occurrence of fires and their frequency (Lima et al., 2018; Liu et al., 2019). Country-level data on these variables are available on several websites online although at different aggregation levels and formats (Verdin et al., 2020; World Bank, 2021). The interaction between rainfall and temperature affects the fire regime and fire probability on a significant scale (Trauernicht, 2019).

2.3 Informative time and spatial components

While fires are a big threat to the environment, predictive tools are limited (Trauernicht, 2019). This author used data on historical fires to build a probability-based model that indicated relative fire frequency and not intensity (Trauernicht, 2019). The current study added to this by using count data on fire frequency instead. It built on the spatial term included by using climatic data that is specific to the location of the fire.

Several authors have worked on various methods to predict fires and the likelihood of their occurrence. The application of geostatistical approaches in this prediction has been useful in several studies (Cracknell and Reading, 2014; Lim et al., 2019; Pereira and Turkman, 2019). Lim et al. (2019) used a maximum entropy (MaxEnt) model in the spatial modeling of fire probability to compare the accuracy of satellite and field survey data on fire occurrence. The satellite data used had many detection errors, which reduced model accuracy. The current study implemented their recommendation to improve model accuracy using spatial filtering (Lim et al., 2019). In addition, the current study used count data on fire frequency rather than fire probability data.

Cracknell and Reading, (2014) acknowledged that the use of machine learning models and explicit spatial information generates accurate predictions but should be used in conjunction with geophysical data to generate geologically plausible predictions (Cracknell and Reading, 2014). For this reason, the current study applied geophysical data derived from satellites to build the plausibility of the models applied.

Researchers have compared traditional methods to modern methods that include machine learning techniques (Cracknell and Reading, 2014; Oliveira et al., 2012; Sayad et al., 2019; Vilar et al., 2016). Vilar et al. (2016) analyzed the socio-economic drivers of wildfire occurrence in central Spain. The authors acknowledged that their work only focused on socio-economic drivers and did not include natural predictors that include meteorological data, topographic data, or fuel models (Vilar et al., 2016). The current study included these meteorological data obtained from satellite data and triangulated to create a clearer picture of factors that affect fire frequency. Oliveira et al. (2012) explored fire occurrence in Mediterranean Europe and modeled its spatial patterns using random forest and multiple regression techniques. The authors used fire density (number of fires/km^2) as their dependent variable in the model (Oliveira et al., 2012). The model data spanned only the main fire season (June-September), which limits the scope of the work and influence of historical data on fire regimes. The current work conducted modeling on data from 18 years to investigate and include the potential effects of seasonality on the predictions.

Considering that the number of fires within a given period is count data, some researchers settled on using Poisson processes to make predictions (Boubeta et al., 2020, 2019; Kim et al., 2021; Lima et al., 2018; Marchal et al., 2017). Boubeta et al. (2019) used the number of forest fires by forest area as their response variable for the period of 2007-2008. This was 24 months, which introduced volatility as they acknowledged that their models gave underpredictions of the numbers of fires in months 2, 3, 8, and 9 of 2009, where there was an unusually high number of fires (Boubeta et al., 2019). The current study acknowledged that predictions will only be accepted if the future behaved as the past has and more data improved the ability of the model to make predictions.

Marchal et al. (2017) exploited Poisson additivity to predict fire frequency from maps of fire weather and land cover in Canada. The authors used piecewise linear functions to model non-linear relations between fire weather and fire frequency for each cover type simultaneously (Marchal et al., 2017). The temporal breadth of the authors' analysis was constrained by the availability of digital data necessary to derive a time series of land-cover maps. The current study extended this possibility as it incorporates data from 2000-2018.

On the other hand, traditional regression techniques have also been applied to fire prediction with different success levels (Kwak et al., 2012). Su et.al.(2019) applied a geographically weighted Negative Binomial regression model which performed better than a Negative Binomial model (Su et al., 2019). A Negative binomial model is more appropriate in modeling in the presence of overdispersion (i.e. when the variance is larger than or equal to the mean (Dong et al., 2016).

Cao et al.(2021) compared four regression techniques to explore factors governing the number of forest fires in Southeast, China (Cao et al., 2021). Besides these, other studies focused on different methods to deal with the fire prediction problem. For instance, Woo et al.,(2017) applied a point process modeling of fire occurrence and Monte Carlo fire simulation (Woo et al., 2017). Li et al.(2020) applied the technique of Artificial Neural Network and Support Vector Machines (Li et al., 2020). Pimont et al.,(2021) predicted regional wildfire activity using probabilistic Bayesian techniques, which is also mentioned by Halim et al.(2021) (Halim et al., 2021; Pimont et al., 2021).

2.4 Constructing prediction intervals

To deal with count data features such as over-dispersion and autocorrelation, several methods are used (Avanzi et al., 2021). Using a Markov-modulated non-homogeneous Poisson process framework, Avanzi et al. (2021) modeled count processes. The authors extend the standard Markov-modulated Poisson process (MMNPP) model by using a more flexible approach in which both cyclical and non-recurring trends can be captured. The authors agree that consideration of larger datasets to be of merit as is attempted in this study. In their study, Yadav et al. (2021) consider if and how generalized Poisson models can substitute any other discrete count models. They conclude that generalized Poisson models provide a better fit for overdispersed data due to excess zeros, consistently in real-time and simulated with varying sample sizes. However, they note that Negative Binomial models can be restricted or re-evaluated against the generalized Poisson model (Yadav et al., 2021). The current study re-evaluated the Negative Binomial model against the Generalized Poisson model as the real world data did not contain zeros.

Flexibility in modeling count data was introduced by relaxing the intrinsic equi-dispersion assumption of Poisson regression (Hilbe, 2011). Considering these Poisson-based models and the negative binomial model, a motivating problem is forecasting the number of satellite-detected fires in a certain geographical region, given previous and current data on fires and weather data. Henceforth, prediction intervals (PIs) are the desired procedures to predict one or more future observations based on existing data (Kim et al., 2021). Kim et al. (2021) point out that PIs are more informative than point predictions since they contain directly an uncertainty quantification regarding the forecast through their confidence coefficients and their interval sizes (Kim et al., 2021).

Important decisions, such as when to establish health interventions or introduce policies to combat irresponsible forest fires, are generally informed by such PIs, therefore proper uncertainty quantification is vital. If intervals are presented without sufficient uncertainty quantifications, erroneous conclusions may be made, leading to a loss of public trust in science.

According to Olive, Rathnayake, and Haile, (2021) there are not many references for prediction intervals for GLMs and GAMs (Olive et al., 2021). In existing work, the prediction intervals often have complex correction factors, are not implementable in some software, and are often only applicable to the full GLM model when $n \geq p$. Here, n refers to sample size and p refers to the number of explanatory variables. The PIs tend to be constructed using Chebyshev's inequality, percentiles of $\hat{D}$, or Bayesian predictive distributions (Olive et al., 2021). The current study sought to add to this body of work by optimizing the PIs for negative binomial models using simulated data and testing on fire frequency data.

2.5 Gaps identified

The literature showed that a variety of count models have been formulated and proposed to model count data. However, minimal research work has been done on the statistical modeling of count response data that contains spatial and temporal features. In addition, while some ways have been proposed to account for and consider overdispersion in count regression models, there was no comprehensive solution on how best to do it. The study also took a step in the future direction of modeling that considers prior distributions through Bayesian techniques. Combining these with optimized prediction intervals will be useful in making more accurate predictions and prediction intervals. Thus, this study focused on hypothetical data simulated for various statistical count models and then the best model was fitted on real fire and climate data downloaded from various secondary sources and aggregated to a format easier to analyze.

Chapter 3

Methodology

3.1 Introduction

This chapter was divided into three sections. First, the study looked at data collection methods and outlined the real-world data collected. These included fire, temperature, and rainfall data. Second, the study outlined the simulation studies carried out. It included the aims of the simulation, data generation mechanisms, target of analysis, methods to be evaluated, and the performance measures. Finally, the study looked at the statistical count models fitted and explained how they will be estimated and implemented in software.

3.2 Data collection

3.2.1 Fire data

The fire data used in this study were downloaded from the NASA FIRMS (NASA, 2021).

3.2.2 Rainfall data

The rainfall data were downloaded from WorldClim, which is a database of high spatial resolution global weather and climate data prepared by the Climatic Research Unit, University of East Anglia (Fick and Hijmans, 2017; Harris et al., 2014). The data contains records between 2000-2018.

The table below shows the description of the variables in the rainfall dataset used in this study.

Table 3.1: Description of rainfall dataset variables.

Name	Units	Description
month	-	Month of year
year	-	Year
rainfall	mm	Rainfall amount in mm

3.2.3 Temperature data

The temperature data were also downloaded from WorldClim, which is a database of high spatial resolution global weather and climate data prepared by theClimatic Research Unit, University of East Anglia (Fick and Hijmans, 2017; Harris et al., 2014). The data contained records between 2000-2018.

The table below shows the description of the variables in the temperature datasets used in this study.

Table 3.2: Description of temperature dataset variables.

Name	Units	Description
month	-	Month of year
year	-	Year
tmin	Celsius	Minimum temperature
tmax	Celsius	Maximum temperature

3.3 Simulation studies

3.3.1 Aims of the simulation

By comparing two model implementations, this simulation study sought to achieve two things. First, it investigated large and small sample bias by varying sample sizes of the simulated data sets that included the time informative component. Second, it evaluated the performance of these model implementations relative to each other in the presence of different degrees of overdispersion.

3.3.2 Data generating mechanisms

For this study, data was generated through parametric draws from a basic structural time series model discussed in (Harvey, 1990). The state-space representation of the fundamental structural time series model was defined as:

$$\textbf{observed series}: y_t = \mu_t + \gamma_t + \varepsilon_t, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2); \tag{3.1}$$

$$\textbf{latent level}: \mu_t = \mu_{t-1} + \beta_{t-1} + \xi_t, \quad \xi_t \sim NID(0, \sigma_\xi^2); \tag{3.2}$$

$$\textbf{latent drift}: \beta_t = \beta_{t-1} + \zeta_t, \quad \zeta_t \sim \text{NID}(0, \sigma_\zeta^2); \tag{3.3}$$

$$\textbf{latent seasonal}: \gamma_t = \sum_{j=1}^{s-1} -\gamma_{t-j} + \omega_t, \ \omega_t \sim NID(0, \sigma_\omega^2), \tag{3.4}$$

for $t = s, \ldots, n; s$ is the periodicity of the data (e.g $s = 6$ for semi-annual data).

As such, the matrix of three explanatory variables X_i was generated using this mechanism with the specified sample size and variances for each component.

The function *datagen.stsm* in package **stsm** generated data from this model (de Lacalle, 2016).

On the other hand, response data Y_i was generated to follow a negative binomial distribution where the θ, representing the overdispersion parameter can vary.

Both data were simulated on $n = 60, 120, 240, 360$, which represented 5 years, 10 years, 20 years, and 30 years worth of monthly data. The study simulated 1000 datasets for each combination of parameters using R software version 4.1.0 and set the seed at "76568". Overdispersion in the simulated dataset was modified through the use of different values of $\theta = 1.5, 5, 10, 100$. The table below shows the simulation scenarios:

Table 3.3: Description of number of datasets for each simulation combination.

Sample size (n)	$\theta = 1.5$	$\theta = 5$	$\theta = 10$	$\theta = 100$
60	1000	1000	1000	1000
120	1000	1000	1000	1000
240	1000	1000	1000	1000
360	1000	1000	1000	1000

3.3.3 Target of analysis

The target of analysis was the selection of the best model in predicting the discrete count variable in the presence of an informative time component and overdispersion.

3.3.4 Methods to be evaluated

In this study, two models were examined: a standard Negative Binomial model, and a Bayesian Negative Binomial MCMC model. The simulation studies here were used to evaluate and compare the two statistical models (Morris et al., 2019).

Data were simulated using a user-defined function and the stsm package, then exported the simulated data and compared the different methods using the same datasets. Each dataset was split into a training and a testing dataset in a non-random and sequential 80:20 ratio

because the data was a time series. Then, each model was fitted onto the training dataset, and its predictive performance was evaluated on the training dataset and on the test dataset.

3.3.5 Performance measures

This study assessed several metrics to evaluate model performance. These included the Root Mean Square Error (RMSE), the bias, and the Mean Absolute Scaled Error (MASE) given that the data was a time series. The model performance metrics were calculated using the `Metrics` package (Hamner and Frasco, 2018). The results of model performance were then graphed using the `ggplot2` package in R (Wickham, 2016).

3.4 Statistical models and estimation

3.4.1 Standard Negative Binomial Model

Suppose that (1) given λ, y has a Poisson(λ) distribution, and (2) λ has the gamma distribution. It is known that the gamma distribution has $E(y) = \mu$ and var(λ) = μ^2/θ for a shape parameter $\theta > 0$, so the standard deviation is proportional to the mean. This yields the gamma mixture of the Poisson distributions called the *negative binomial distribution* for y. Its probability mass function is

$$p(y;\mu,\theta) = \frac{\Gamma(y+\theta)}{\Gamma(\theta)\Gamma(y+1)}\left(\frac{\mu}{\mu+\theta}\right)^{y}\left(\frac{\theta}{\mu+\theta}\right)^{\theta}, y \geq 0. \tag{3.5}$$

With θ fixed, this is a member of an exponential dispersion family appropriate for discrete variables, with natural parameter $\log[\mu/(\mu+\theta)]$. μ is the location or shape parameter while θ is the dispersion parameter.

The highest temperature, minimum temperature, and rainfall per month were all taken into account as explanatory variables. Prior to the formal modeling procedure, correlations among

each set of explanatory variables were investigated. Pearson correlation was used to evaluate the correlations.

This model was implemented in R software using the glm.nb function in the MASS package (Venables and Ripley, 2002). The function performed a maximum likelihood estimation of the generalized Negative Binomial linear model.

3.4.2 Bayesian Negative Binomial MCMC Model

In this type of model, it was considered that information on the data that influences the parameter values exists, which is called the prior distribution. At each loop stage of the estimation process, the respective prior distribution updates the posterior distribution for each predictor (Hilbe, 2014). The posterior distribution is updated by multiplying the likelihood by the prior:

$$p(\theta|y) \propto L(\theta)\pi(\theta) \tag{3.6}$$

$p(\theta|y)$ is the posterior distribution that explains the predictors; $L(\theta)$ is the likelihood function and $\pi(\theta)$ is the prior distribution (Hilbe, 2014).

Assuming the Negative Binomial characteristics of the previous model in (3.5), the Bayesian Negative Binomial model takes the following form:

$$y_i \sim \mathscr{P}oisson(\nu_i, \mu_i) \tag{3.7}$$

The inverse link function can be expressed as follows:

$$\mu_i = \exp(x_i'\beta) \tag{3.8}$$

It is proposed that β contains two elements:

$$\beta = (\beta_a, \beta_b) \tag{3.9}$$

Then, the multivariate Normal prior on β_a is:

$$\beta_a \sim \mathcal{N}(b_0, B_0^{-1}) \tag{3.10}$$

where b_0 is the prior mean of β_a and B_0 is the prior precision of β_a. This includes the original predictor variables included in the model.

On the other hand, the prior on β_b is:

$$\beta_b = \bar{\Theta} \tag{3.11}$$

$\bar{\Theta}$ includes prior means of the response variable for every time period in the historical data.

In this case, the random effect that handles overdispersion is assumed to be distributed Gamma:

$$\nu_i \sim \mathcal{G}amma(\rho, \rho) \tag{3.12}$$

The overdispersion parameter has a prior with the following form:

$$p(\rho|e, f, g) \propto \rho^{e-1}(\rho + g)^{-(e+f)} \tag{3.13}$$

where e, f and g are hyperpriors for the distribution ρ

This model was implemented in R software using the stanglm.nb function in the rstanarm package (Goodrich et al., 2020). The function performed a full Bayesian estimation of the generalized Negative Binomial linear model via MCMC and priors were defined on the coefficients of the GLM based on the standard model's performance.

Bayesian Prediction Intervals

A time series that has n observations can be denoted as $x_1, x_2, \ldots, x_n$. In the event that a researcher want to forecast the value of the series k steps ahead, it would mean that they want to predict the observed value at time $(n+k)$. The integer k is defined as the forecasting horizon or lead time **?**. The value of the point forecast at time $(n+k)$ is denoted by $\hat{x}_n(k)$. When the observed value is available, the forecast error, denoted by $e_n(k)$ can be calculated as follows:

$$e_n(k) = x_{n+k} - \hat{x}_n(h) \tag{3.14}$$

The theoretical formula for the calculation of prediction intervals (PIs) is generally of the same form. A $100(1-\alpha)\%$ prediction interval for the value of k steps ahead is given by

$$\hat{x}_n(k) \pm z_{\alpha/2)} \sqrt{\text{Var}[e_n(k)]} \tag{3.15}$$

where appropriate formula for $\hat{x}_n(k)$ and for $\text{Var}[e_n(k)]$ exist for the model.

In R, these intervals were calculated using the rstanarm package and the predictiveinterval function. The function computes Bayesian predictive intervals for models fit using MCMC. The upper and lower intervals were then extracted from the function output and plotted together with the actual values.

Chapter 4

Analysis and Interpretation of Results

4.1 Introduction

This chapter is divided into three sections. First, the work looks at the descriptive statistics of the data used in this study. Next, the results of the simulation studies are presented. Lastly, the output of fitting the final model on the real world data is shown.

4.2 Descriptive statistics

4.2.1 Fire and climate data

Table 4.1 shows the summary statistics of the data and the selected variables where the sample size is 218. The overall minimum number of fires over the period (2000-2018) was 10 while the maximum number was 1661. In the period of 218 months, the mean and median are 277.13 and 155.00, respectively, with a standard deviation of 304.91.

Table 4.1: Summary of real world dataset variables.

Variable	**Min**	**Max**	**Median**	**Mean**	**SD**
Fire frequency	10.00	1661.00	155.00	277.13	304.91
Maximum temperature	23.44	34.82	29.27	29.19	2.30
Minimum temperature	11.26	22.34	18.36	18.17	2.14
Rainfall	8.43	297.86	70.659	84.04	51.69

On the other hand, the mean and median mean maximum temperatures were 29.19 and 29.29 respectively. For the mean minimum temperature, the mean was 18.17 degrees Celsius with a standard deviation of 2.14. Lastly, the mean amount of rainfall was 84.04mm with a standard deviation of 51.69.

The figure 4.1 below shows the trend and patterns observed on the data and its four variables between the year 2000 and 2018. The irregular patterns in each graph show the presence of seasonality in the data, which is to be expected as fires, rainfall and temperature change with season of the year.

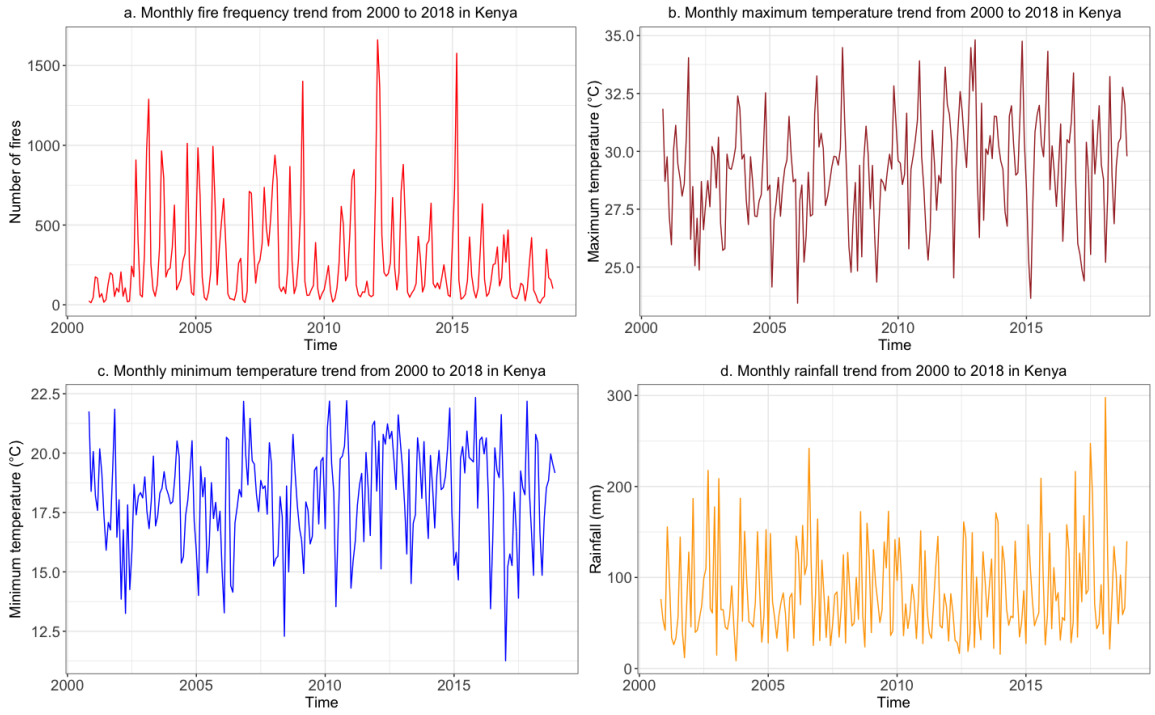

Figure 4.1: Time series of the four variables in the study from the real world data. Graph **a** shows the time series of monthly fire frequency, **b** shows the monthly maximum temperature, **c** shows the monthly minimum temperature and **d** shows the monthly rainfall amounts

4.3 Simulation studies

This section outlines the four scenarios under which the Bayesian Negative Binomial model was compared to the standard Negative Binomial model.

The study compared two model implementations (Standard Negative Binomial and Bayesian Negative Binomial) and their performance on 1000 datasets per parameter combination. The results show the output of four metrics:

1. The bias of the model on the test datasets (*biastest*)
2. The mean absolute scale error (MASE) on the test datasets (*masetest*)
3. The root mean squared error (RMSE) on the test datasets (*rmsetest*) and,
4. The root mean squared error (RMSE) on the training datasets (*rmsetrain*).

4.3.1 Scenario 1: θ = 1.5

Here, the study compared the results when the dispersion parameter $\theta = 1.5$ and the sample size is 60, 120, 240 or 360.

Sample size n = 60 (5 years)

The BNB only outperformed the NB on the RMSE on training set metric. For the rest, the NB was the better model as shown in Table 4.2.

Table 4.2: Model performance metrics at n = 60 when θ = 1.5.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	216.07	196.11
RMSE on testing set	227.06	266.23
MASE on testing set	0.84	0.98
Bias on testing set	2.29	-31.58

There seems to be more variance in the bias of the BNB model, with the NB model showing lower deviation from zero in Figure 4.2. However, the BNB shows consistently higher MASE and RMSE on testing data. On the other hand, the NB has higher RMSE on the training data.

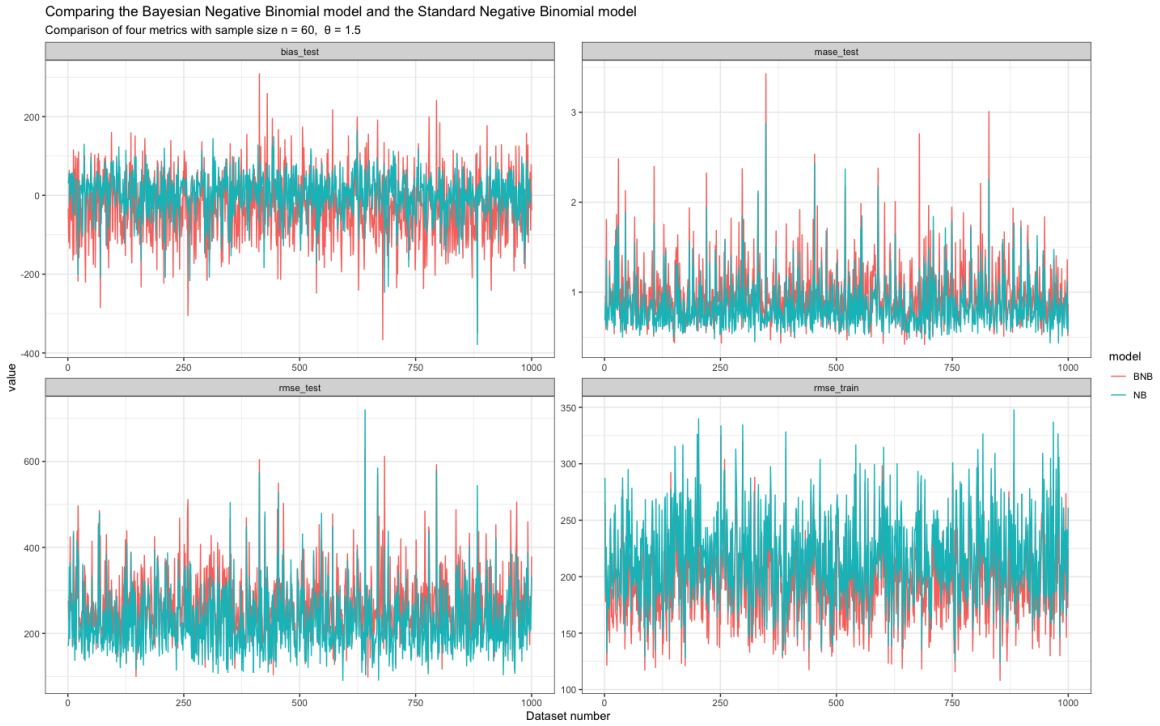

Figure 4.2: Model performance on the same 1000 datasets when sample size = 60 and θ = 1.5. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 120 (10 years)

The observations on Figure 4.3 are confirmed in Table 4.3, with visible lower margins between the model performance. However, the NB model still outperforms the BNB model.

Table 4.3: Model performance metrics at n = 120 when θ = 1.5.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	220.42	208.64
RMSE on testing set	228.46	244.61
MASE on testing set	0.77	0.83
Bias on testing set	4.43	-11.88

There seems to be more variance in the bias of the BNB model, with the NB model showing lower deviation from zero in Figure 4.3. However, the NB shows consistently lower MASE and RMSE on testing data. On the other hand, the NB has higher RMSE on the training data. It was noted that there was a slight decrease in the variance between the two models especially on the MASE and RMSE on test datasets compared to when the sample size was smaller at 60.

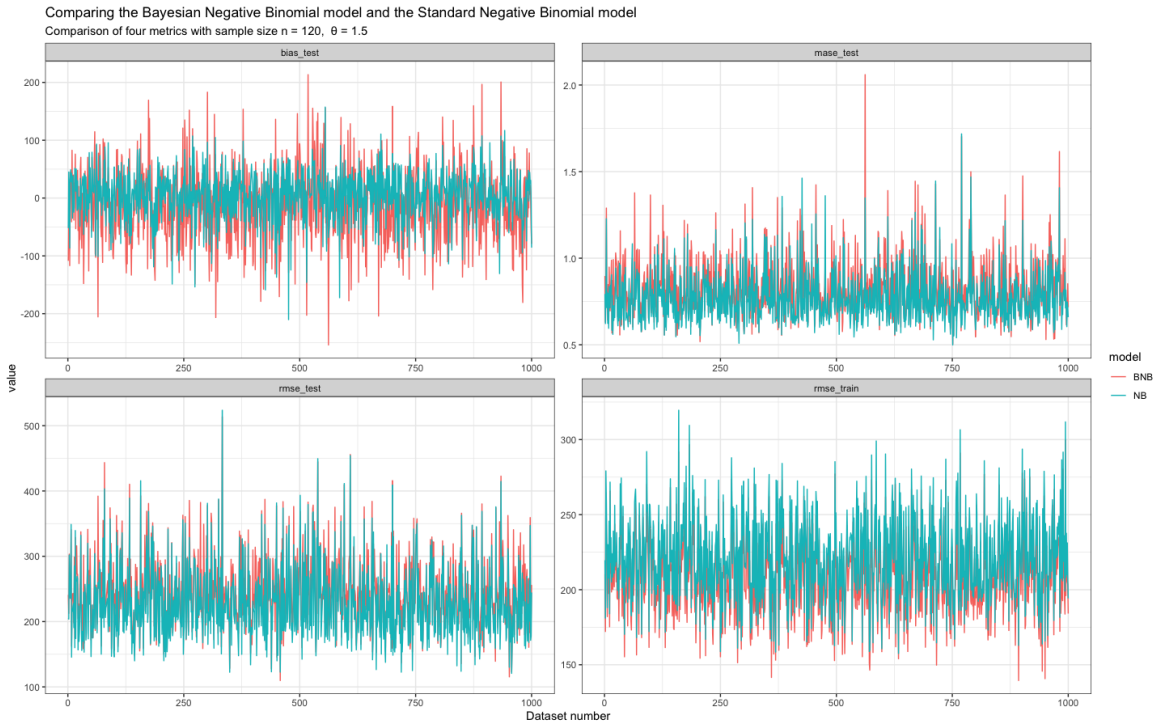

Figure 4.3: Model performance on the same 1000 datasets when sample size = 120 and θ = 1.5. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 240 (20 years)

As before, Table 4.4 shows the reduced margin in the MASE on testing data as the sample size increased. However, the NB still outperforms the BNB.

Table 4.4: Model performance metrics at n = 240 when θ = 1.5.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	223.31	217.27
RMSE on testing set	227.87	234.51
MASE on testing set	0.75	0.78
Bias on testing set	6.83	-7.08

Figure 4.4 shows that the BNB has a lot of negative bias values, suggesting it predicts higher values compared to the actual. As seen in the previous sample size of 120, the margins in performance between the BNB and NB for the metrics MASE and RMSE (on testing data) seem to be diminishing in this large sample.

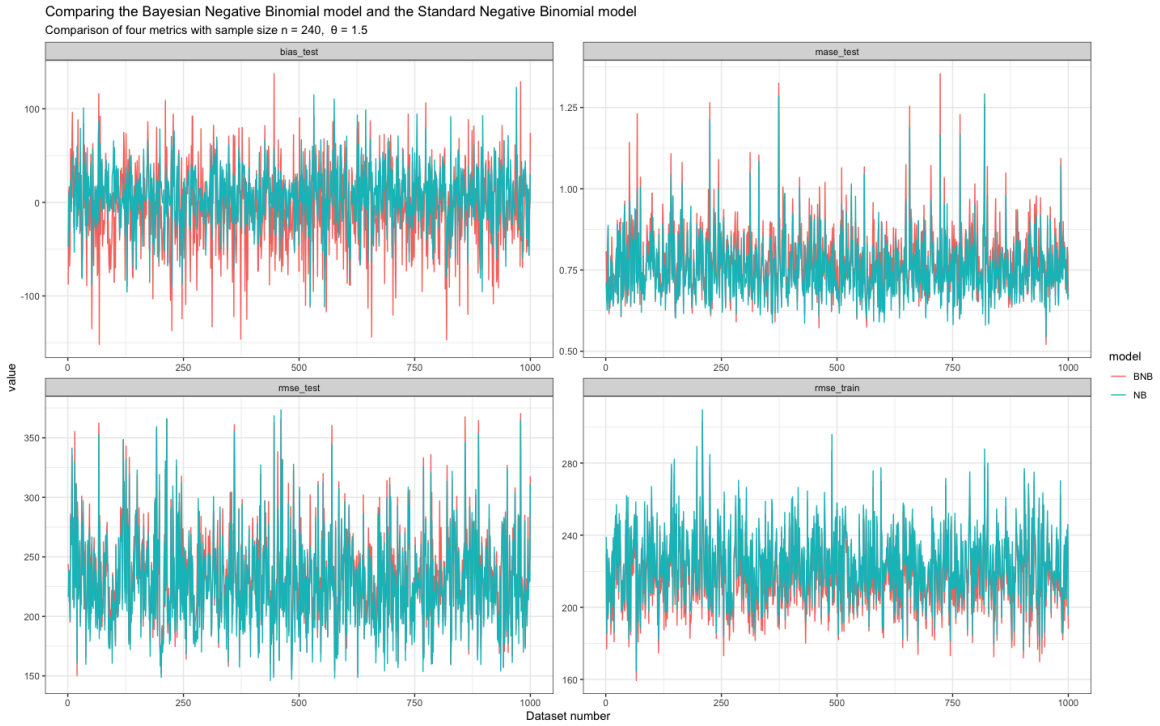

Figure 4.4: Model performance on the same 1000 datasets when sample size = 240 and θ = 1.5. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 360 (30 years)

Using the mean values of the two models, the study makes comparisons as shown in Table 4.5.

Table 4.5: Model performance metrics at n = 360 when θ = 1.5.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	225.11	221.08
RMSE on testing set	228.30	232.34
MASE on testing set	0.74	0.76
Bias on testing set	-1.42	-4.91

Although the NB had lower metric values (MASE, RMSE on testing data), it is noted that it was only a slight advantage over the BNB.

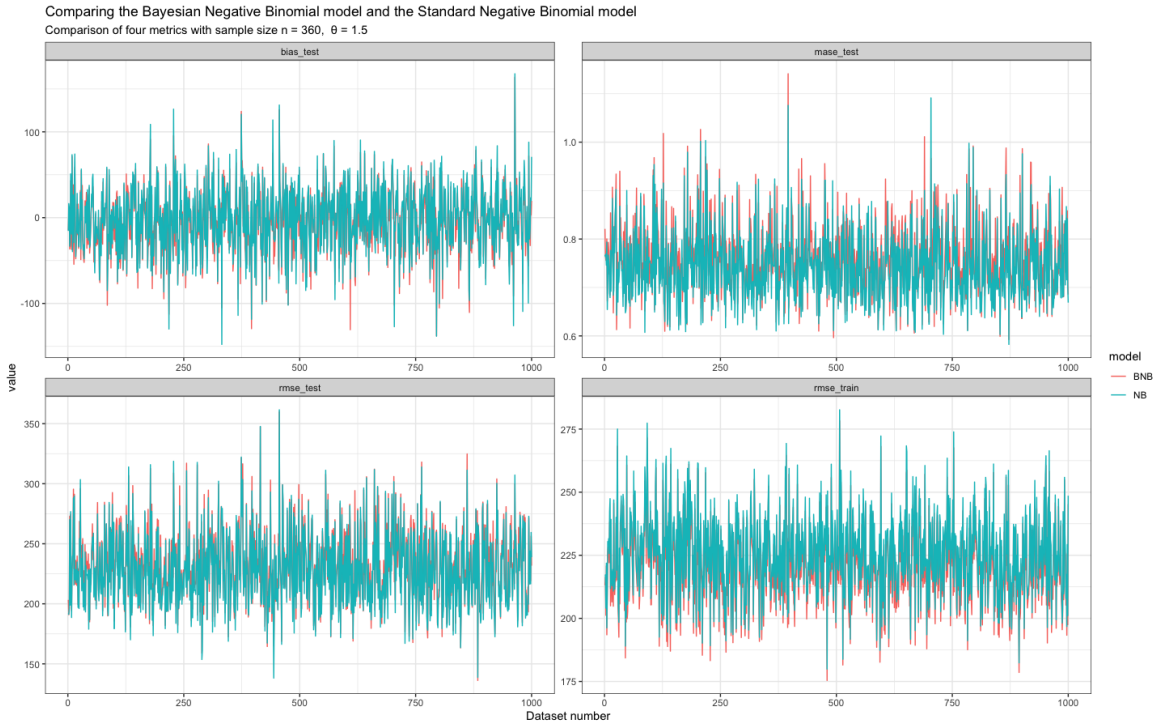

Figure 4.5: Model performance on the same 1000 datasets when sample size = 360 and θ = 1.5. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Here, as shown in Figure 4.5, there is a similar performance in the two models compared and it is not easy to tell, which model performed better than the other.

4.3.2 Scenario 2: $\theta = 5$

Here, the study compared the results when the dispersion parameter $\theta = 5$ and the sample size is 60, 120, 240 or 360.

Sample size n = 60 (5 years)

It is seen that the NB outperformed the BNB in all metrics on the testing datasets and was therefore superior.

Table 4.6: Model performance metrics at n = 60 when $\theta = 5$.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	119.15	104.95
RMSE on testing set	129.47	143.44
MASE on testing set	0.78	0.87
Bias on testing set	-1.05	-6.57

First, Figure 4.6 shows that it is difficult to differentiate the bias in the two models. The graph suggests that the NB had lower values of MASE and RMSE on the test datasets. However, it is seen that the BNB had lower RMSE on the training data.

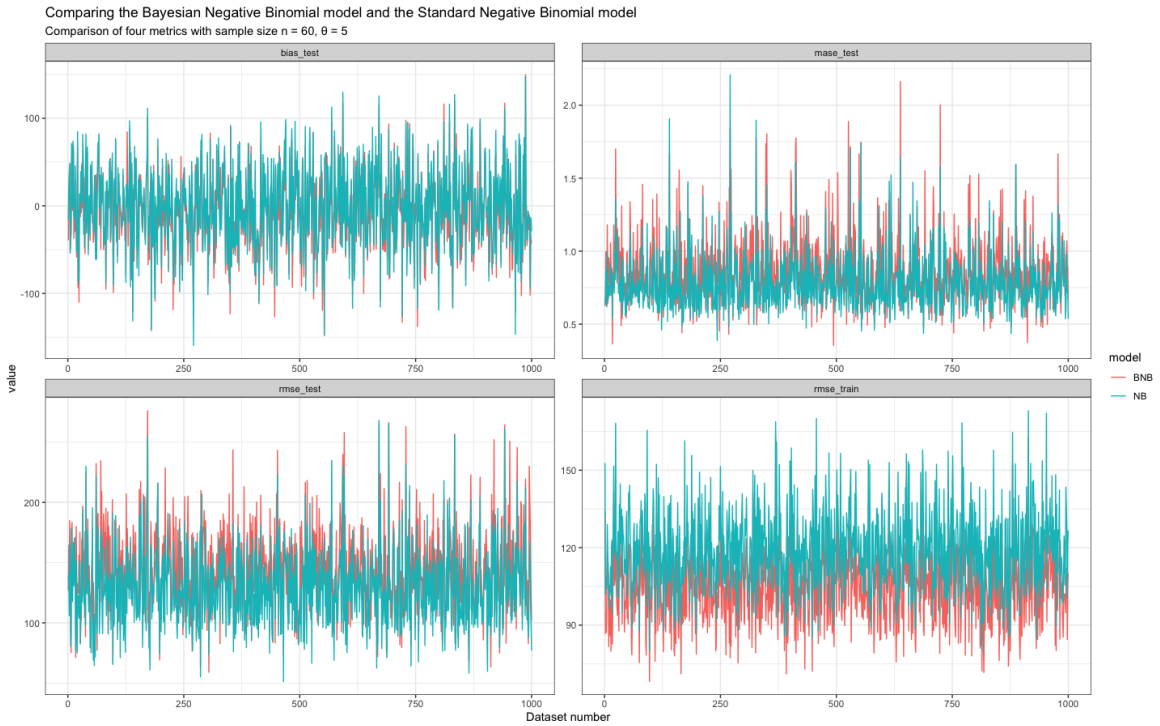

Figure 4.6: Model performance on the same 1000 datasets when sample size = 60 and $\theta = 5$. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 120 (10 years)

It is seen that the BNB outperformed the NB in terms of the bias, with its mean bias closer to zero than the NB as shown in Table 4.7.

Table 4.7: Model performance metrics at n = 120 when $\theta = 5$.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	121.45	114.69
RMSE on testing set	126.41	132.96
MASE on testing set	0.75	0.79
Bias on testing set	6.71	-3.51

According to Figure 4.7, NB had more stable bias values, slightly lower MASE and RMSE than the BNB on testing data and higher RMSE on the training data.

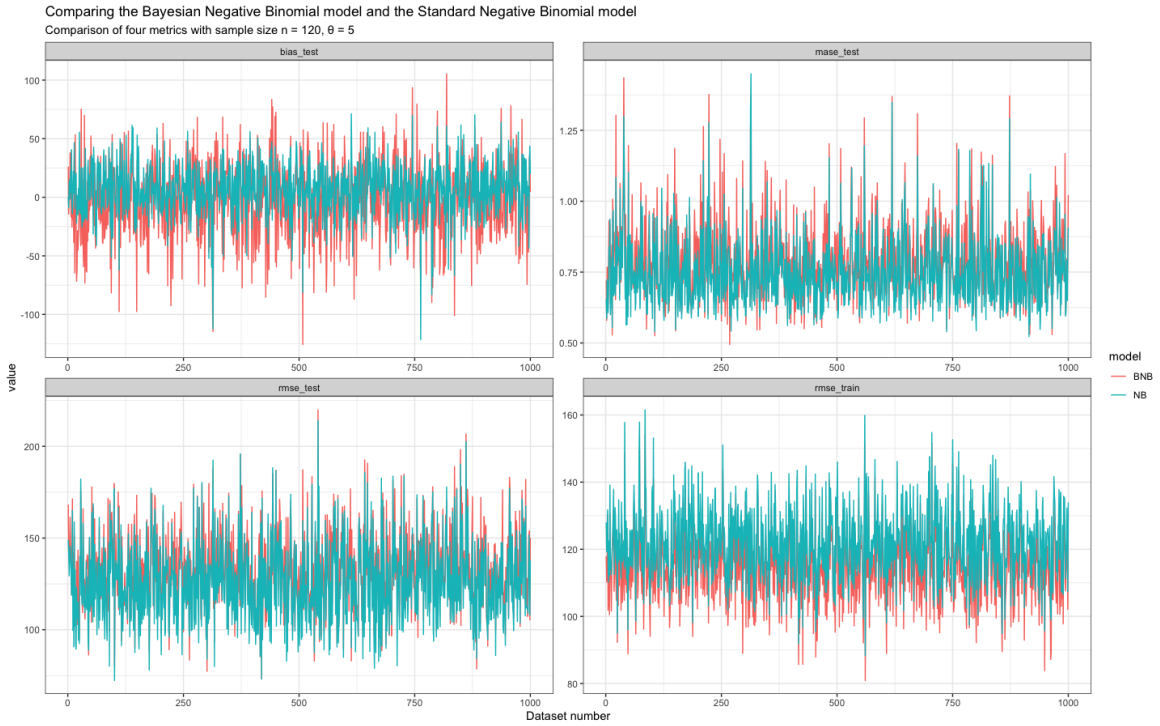

Figure 4.7: Model performance on the same 1000 datasets when sample size = 120 and $\theta = 5$. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 240 (20 years)

Table 4.8 shows that the MASE on the testing datasets margin between the two models reduced in this simulation and the bias of the BNB was closer to zero than that of the NB.

Table 4.8: Model performance metrics at n = 240 when $\theta = 5$.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	123.51	120.06
RMSE on testing set	126.27	129.52
MASE on testing set	0.74	0.76
Bias on testing set	6.26	-2.93

Compared to the previous simulations, it is seen that the differences between the two models are only obviously visible in the bias. There is increased similarity in the performance for RMSE on the test and training datasets and MASE on the test datasets.

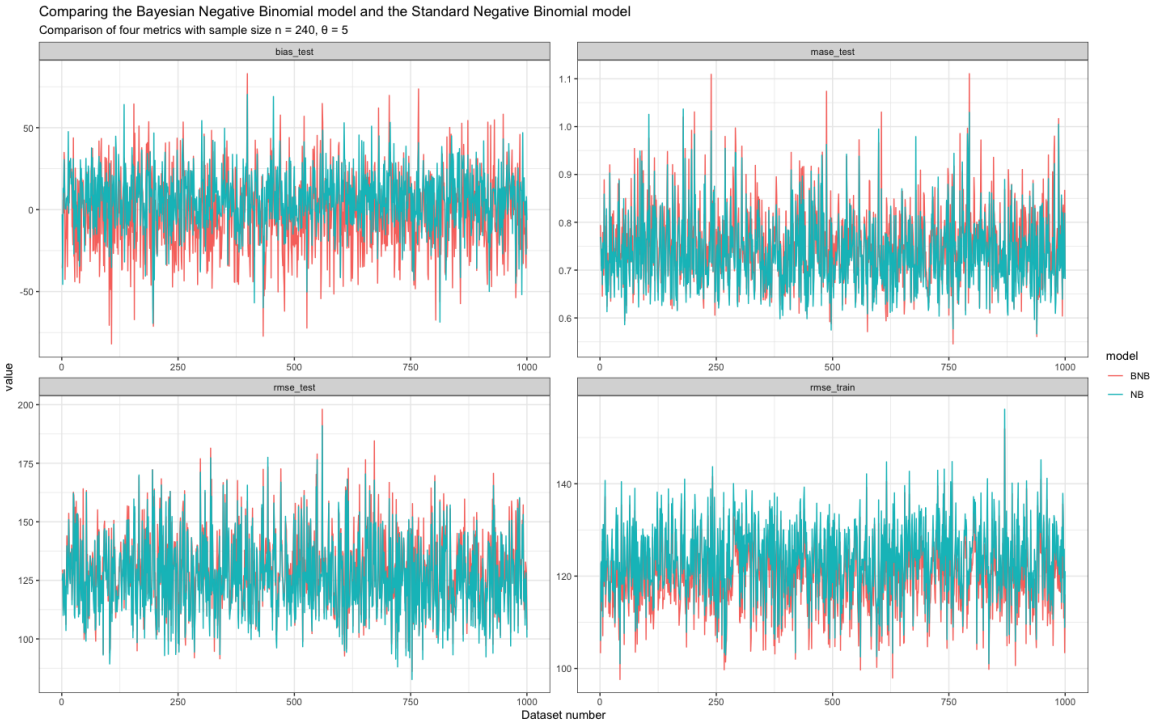

Figure 4.8: Model performance on the same 1000 datasets when sample size = 240 and $\theta = 5$. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 360 (30 years)

Unlike the previous two sample sizes, NB had a mean bias closer to zero here and thus, better than the BNB (Table 4.9). However, the margin between the MASE on the testing datasets was the lowest and indicates a similar performance even though NB's value was lower.

Table 4.9: Model performance metrics at n = 360 when $\theta = 5$.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	124.07	121.78
RMSE on testing set	126.22	128.47
MASE on testing set	0.73	0.74
Bias on testing set	-0.83	-1.64

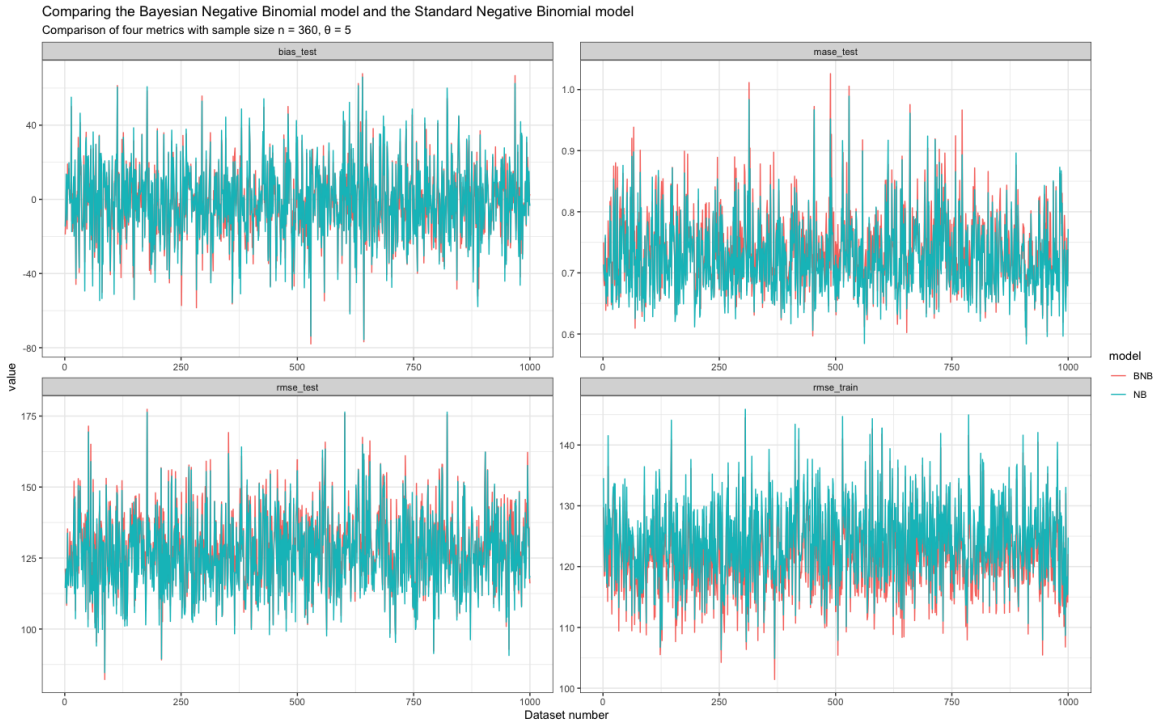

Figure 4.9: Model performance on the same 1000 datasets when sample size = 360 and $\theta = 5$. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

From Figure 4.9, it is seen that there is no visible major difference between all the metrics of the two models. Table 4.9 gives more details to allow comparison between the NB and the BNB.

4.3.3 Scenario 3: θ = 10

Here, the results are compared when the dispersion parameter $\theta = 10$ and the sample size is 60, 120, 240 or 360.

Sample size n = 60 (5 years)

According to Table 4.10, NB outperformed the BNB on the MASE and RMSE on the testing set. However, the BNB was the better performer in terms of the bias and the RMSE on the training datasets.

Table 4.10: Model performance metrics at n = 60 when θ = 10.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	84.85	74.69
RMSE on testing set	91.66	100.46
MASE on testing set	0.79	0.87
Bias on testing set	7.01	-3.43

According to Figure 4.10, BNB had lower bias values, slightly higher MASE on testing datasets, slightly higher RMSE on the testing datasets and lower RMSE on the training dataset.

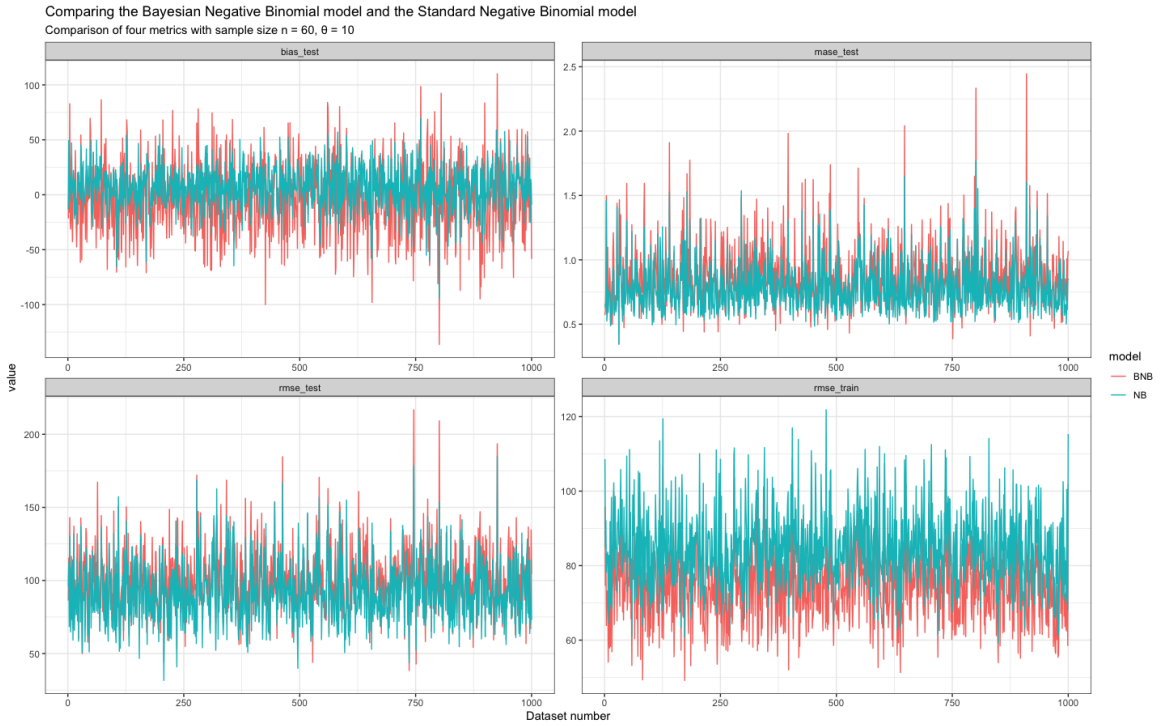

Figure 4.10: Model performance on the same 1000 datasets when sample size = 60 and θ = 10. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 120 (10 years)

Table 4.11 shows that the BNB outperformed the NB only on the bias and RMSE on the training datasets.

Table 4.11: Model performance metrics at n = 120 when θ = 10.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	87.23	82.18
RMSE on testing set	90.93	95.30
MASE on testing set	0.75	0.79
Bias on testing set	5.98	-2.46

Similar to the previous simulation, there is a similar pattern of lower bias for the BNB, slightly higher MASE and RMSE on testing datasets and lower RMSE on training dataset as shown in Figure 4.11.

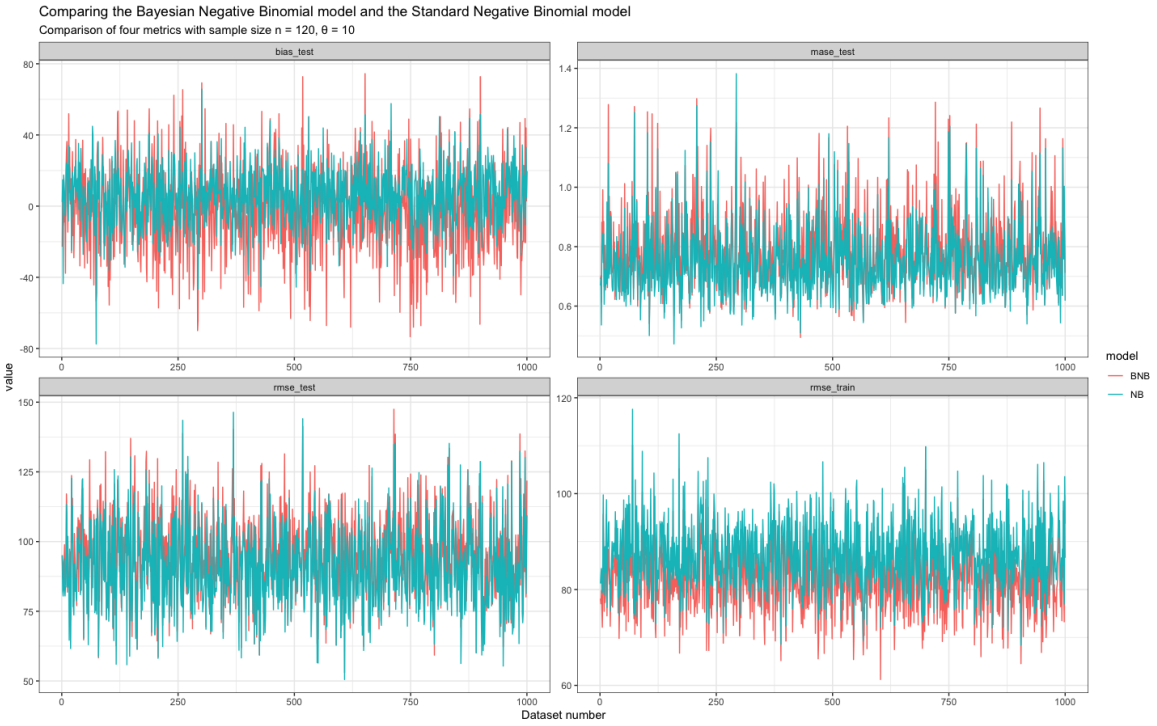

Figure 4.11: Model performance on the same 1000 datasets when sample size = 120 and $\theta = 10$. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 240 (20 years)

The summary in Table 4.12 gives a good picture on difference in performance between the models. It is seen that the MASE and RMSE on testing datasets margins are lower despite the NB performing better. On the other hand, the BNB was a better performer in terms of bias, with its mean bias closer to zero.

Table 4.12: Model performance metrics at n = 240 when θ = 10.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	88.10	85.54
RMSE on testing set	90.19	92.43
MASE on testing set	0.73	0.75
Bias on testing set	7.02	-0.65

Figure 4.12 shows BNB had lower bias and lower RMSE on training datasets compared to the NB. However, it is difficult to tell the differences in performance on the MASE and RMSE on the testing datasets.

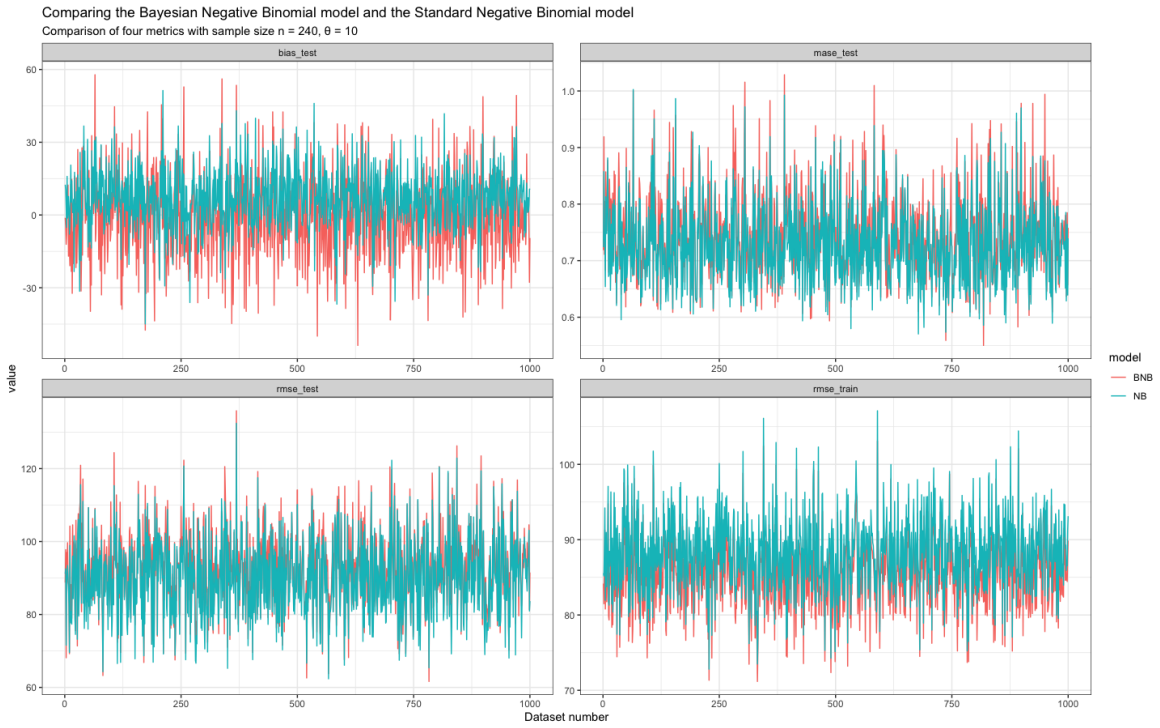

Figure 4.12: Model performance on the same 1000 datasets when sample size = 240 and θ = 10. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 360 (30 years)

It is seen that the NB outperformed BNB on the RMSE and MASE on testing set, and on bias. However, it is noted that the margins are quite small.

Table 4.13: Model performance metrics at n = 360 when θ = 10.

Metric	Negative Binomial	Bayesian Negative Binomial
RMSE on training set	88.51	86.92
RMSE on testing set	90.40	91.92
MASE on testing set	0.72	0.74
Bias on testing set	0.02	-0.51

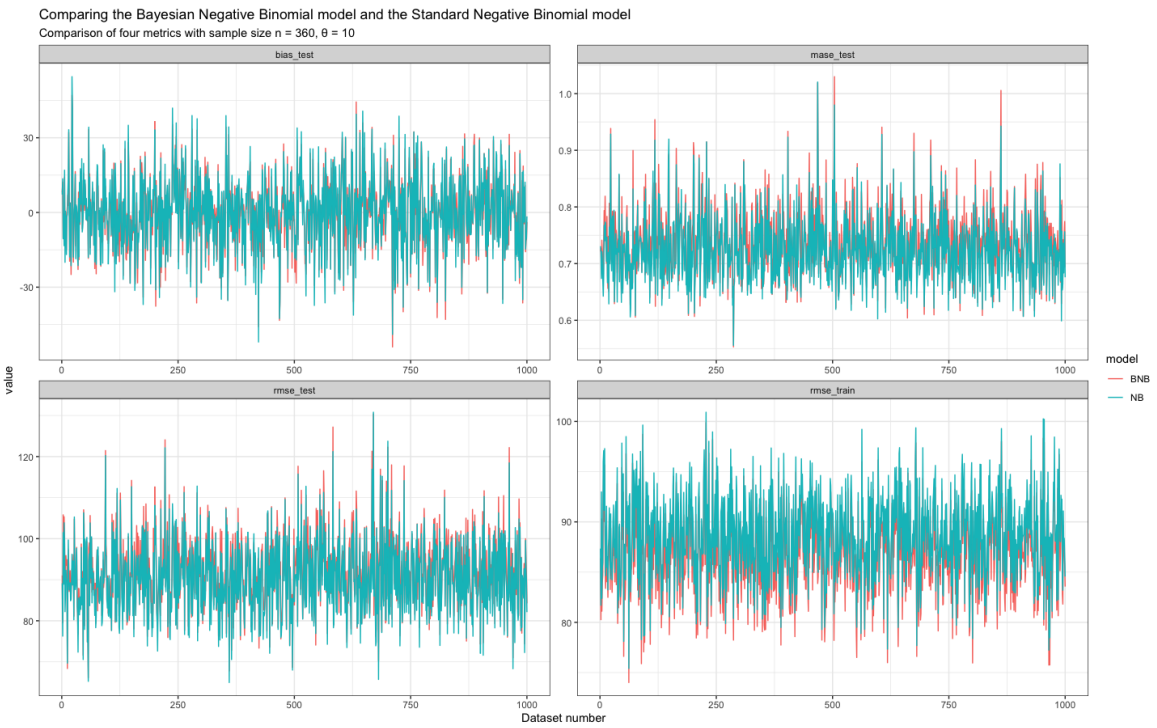

Figure 4.13: Model performance on the same 1000 datasets when sample size = 360 and θ = 10. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

At a glance, Figure 4.13 shows that the two models performed quite similar to each other. The differences are best shown in Table 4.13.

4.3.4 Scenario 4: θ = 100

Here, the results were compared when the dispersion parameter $\theta = 100$ and the sample size is 60, 120, 240 or 360.

Sample size n = 60 (5 years)

Table 4.14 shows that the NB outperformed the BNB on MASE and RMSE on testing datasets. However, BNB had bias closer to zero and thus was the better performer.

Table 4.14: Model performance metrics at n = 60 when θ = 100.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	30.73	27.00
RMSE on testing set	33.64	36.56
MASE on testing set	0.78	0.85
Bias on testing set	6.79	-1.81

According to Figure 4.14, the BNB had lower bias and RMSE on trainng data compared to the NB. In terms of the MASE and RMSE on testing data, there seems to be some similarity but the NB has lower values and a few outliers.

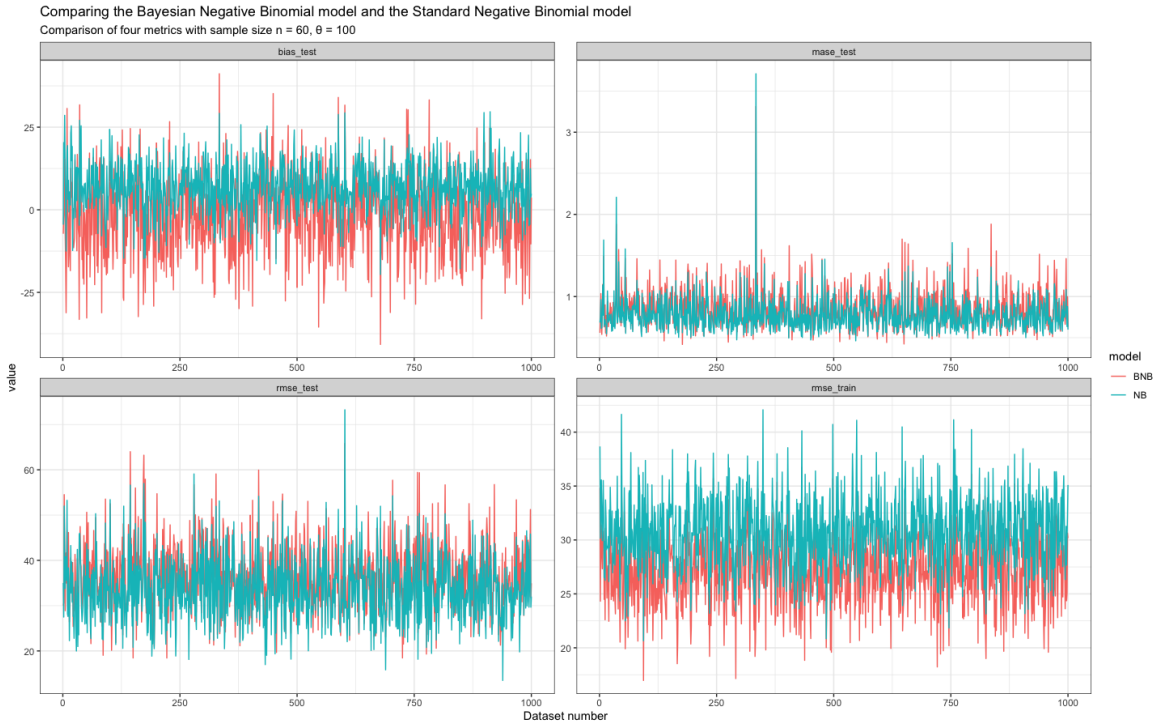

Figure 4.14: Model performance on the same 1000 datasets when sample size = 60 and θ = 100. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 120 (10 years)

According to Table 4.15, the NB outperformed the BNB in terms of the MASE and RMSE on testing datasets. However, the BNB was a better performer with a bias closer to zero than the NB.

Table 4.15: Model performance metrics at n = 120 when θ = 100.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	31.62	29.78
RMSE on testing set	32.84	34.49
MASE on testing set	0.74	0.78
Bias on testing set	6.88	-0.99

Figure 4.15 also shows that the BNB had lower bias and RMSE on training datasets. The two models performed quite similarly in the MASE and RMSE on testing datasets.

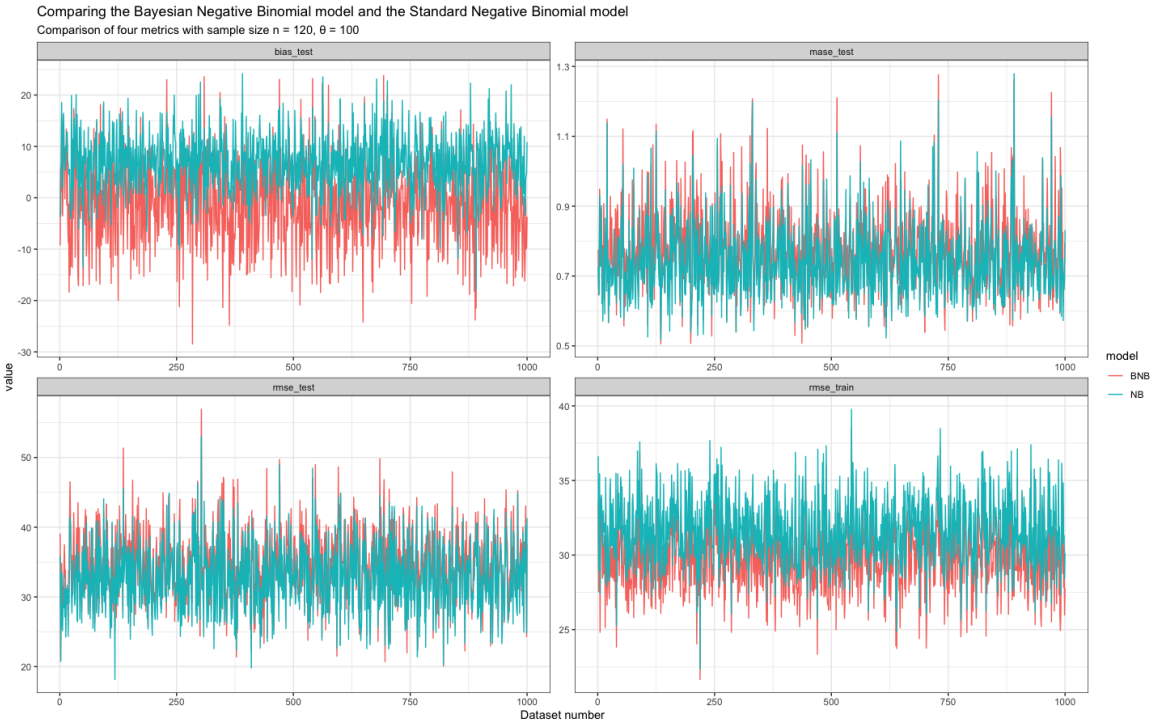

Figure 4.15: Model performance on the same 1000 datasets when sample size = 120 and θ = 100. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 240 (20 years)

Table 4.16 shows that the NB performed slightly better than the BNB on the MASE and RMSE on testing datasets. However, the BNB was a better performer on bias and RMSE on training datasets.

Table 4.16: Model performance metrics at n = 240 when θ = 100.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	88.10	85.54
RMSE on testing set	90.29	92.43
MASE on testing set	0.73	0.75
Bias on testing set	7.02	-0.65

Figure 4.16 shows that the BNB had lower bias, and slightly lower RMSE on the training datasets. The MASE and RMSE on the testing datasets for the two models are close.

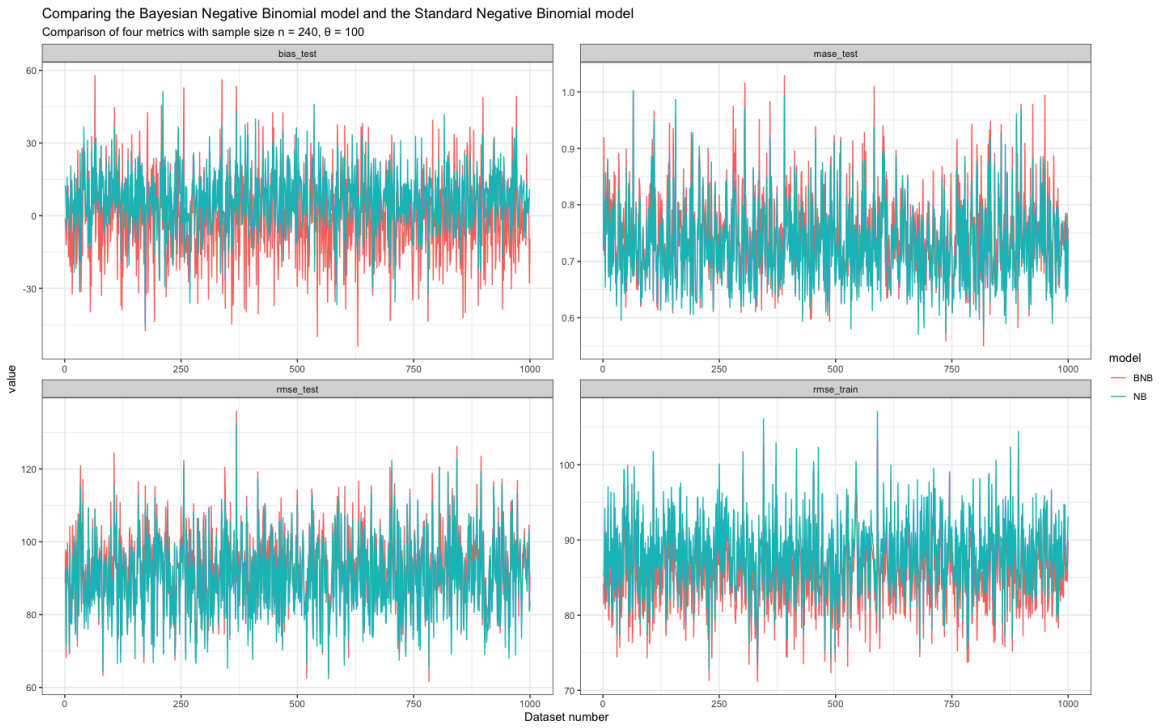

Figure 4.16: Model performance on the same 1000 datasets when sample size = 240 and θ = 100. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Sample size n = 360 (30 years)

Table 4.17 shows the mean performance metric values of the two models. NB outperformed the BNB on bias, MASE on testing data, and RMSE on testing data. Notably, the difference in margin was low.

Table 4.17: Model performance metrics at n = 360 when θ = 100.

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	32.09	31.49
RMSE on testing set	32.60	33.14
MASE on testing set	0.72	0.73
Bias on testing set	-0.04	-0.14

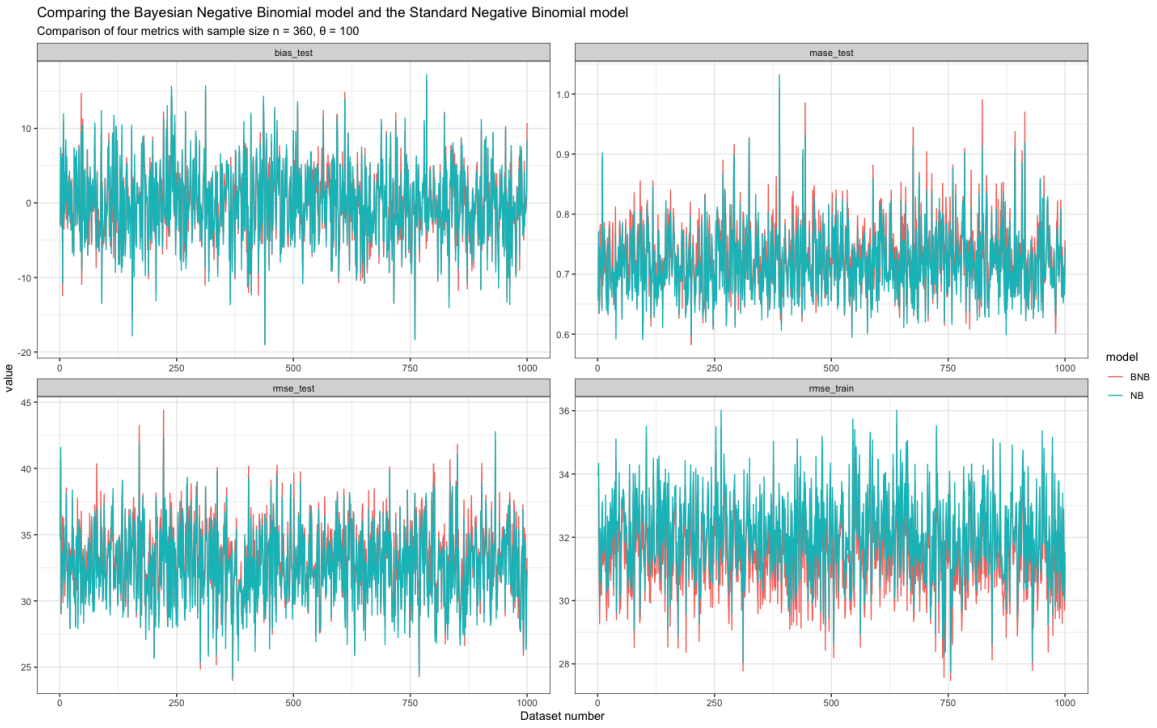

Figure 4.17: Model performance on the same 1000 datasets when sample size = 360 and θ = 100. Models compared are Bayesian Negative Binomial (BNB) and standard Negative Binomial (NB).

Figure 4.17 shows close performance between the two models. Aside from the RMSE on the training dataset, it is hard to tell, which model performed better.

4.3.5 Performance comparison of models

In many instances and on the measures of MASE and RMSE, the NB was a better model. However, it is noted how the BNB had lower bias than the NB model in most cases, which makes it a better model in that aspect.

4.4 Model estimation on actual data

Both models were fitted on real world data and compared the results as shown in Table 4.18.

4.4.1 Model performance metrics

The results show that the NB outperformed the BNB on the RMSE and bias on testing set. However, the BNB was the better performer on RMSE on the training set and the MASE on the testing set as shown in Table 4.18.

Table 4.18: Model performance metrics at n = 218

Metric	**Negative Binomial**	**Bayesian Negative Binomial**
RMSE on training set	308.76	227.94
RMSE on testing set	180.69	207.53
MASE on testing set	1.27	1.16
Bias on testing set	-118.23	-122.55

4.4.2 Prediction intervals

To understand the importance of prediction intervals, the prediction intervals on the test dataset were calculated using the BNB model. The results are as shown in Figure 4.18. It can be seen that for the most part, the actual values are within the prediction interval.

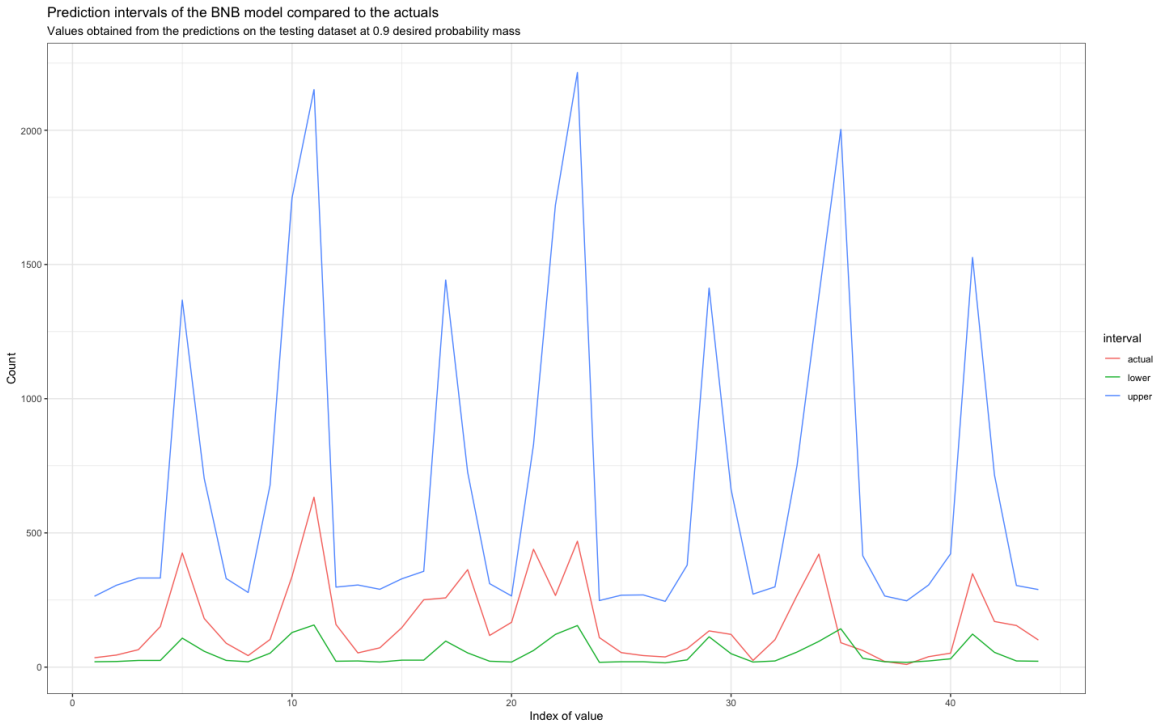

Figure 4.18: The prediction interval of the BNB model compared to actual figures. Desired probability is 0.9 and figure shows upper interval, lower interval and the actual value

The peak values shows that the model can be used to identify possibility of a peak fire season based on historical data.

Chapter 5

Discussion, Conclusion and Recommendations

5.1 Introduction

This chapter summarizes the key findings of this study, explains what the results mean, and outlines why the results matter. Furthermore, we look at the limitations of the study and give recommendations on what practical actions and studies can be done to further work in the field.

5.2 Key findings of the study

As expected, the plotting of the time series of the four variables of fire frequency, minimum temperature, maximum temperature, and rainfall shows variation due to the seasonal cycle. The simulation results show that both models work well on overdispersed data, with the BNB model on average, scoring better in terms of the RMSE on training dataset and bias on the testing datasets. However, the standard NB model still did better than the BNB model in many of the simulation runs especially on RMSE on testing datasets and the MASE on the testing datasets. The BNB model, when applied to the real world data performed better in terms of the MASE on the testing dataset. Further, the BNB model provided prediction intervals, which enveloped the actual data points, allowing it more flexibility when predicting monthly fire frequency.

5.3 What the results mean and why they matter

A discrete probability distribution, such as a Poisson or Negative Binomial likelihood, is the most appropriate way to represent count data; otherwise, the model would be biased and misspecified (De Souza et al., 2015). Considering that this type of data is tied to time, time-trend analysis is crucial in understanding how the system operates and changes. We acknowledge that prediction of the number of fires is tough as a result of the many factors at play thus affecting the system (Halim et al., 2018). However, due to the randomness of the number of fire occurrences each month, application of the Bayesian model was more appropriate (de Oliveira et al., 2012; Grzenda, 2015). We compared the differences of the models at different sample sizes and the results show that Bayesian techniques can also handle smaller sample sizes ($n < 120$) (Yu et al., 2007). The value offered by the Bayesian technique was the assumption that the parameter of interest is random, and that it lies within the prediction intervals (Grzenda, 2015). In many instances, the BNB model generated results similar to that of the frequentist NB model (de Oliveira et al., 2012; Grzenda, 2015). The intention was to allow for more precise predictions through this method as reported by (BahooToroody et al., 2020), which we have achieved in the results.

A problem this study sought to solve was on the dimensions of data used to predict fire frequency. Unlike the shorter time spans used in (Boubeta et al., 2019; Marchal et al., 2017; Oliveira et al., 2012), the study simulated data under different scenarios and thus had sample sizes of 60, 120, 240 or 360 months. As is utilised in literature, the application of the Negative Binomial distribution as in (Dong et al., 2016; Kwak et al., 2012; Su et al., 2019) allowed better modeling of overdispersed count data.

To tackle the issue of underpredictions as in (Boubeta et al., 2019) , the implementation of the Bayesian Negative Binomial model allowed for the calculation of prediction intervals. This is coming up as a useful technique as presented by (Pimont et al., 2021) and (Halim et al., 2021) who acknowledge the importance of Bayesian inference in fire prediction.

Regarding the inclusion of components of serial autocorrelation that are inherent to the sort of data being simulated, the simulations utilized in the study have limitations. This

restricts the study's scope while allowing for further investigation to examine the existence of autocorrelations and the integration of the outcome and predictor variables that would cause spurious regressions.

The study assumed that the number of fire incidences followed a Poisson process, and even though the results are informative, there is limited relevance in understanding exactly when the next event will occur. Further, we did not include any outliers in the simulation runs, which would be expected in a real world system. Despite this, the fitted models give a direction on how the real world variables would behave. Lastly, the fire regime system is complicated and is influenced by many environmental and human variables and is therefore, not completely explained by the variables.

5.4 Conclusion

In conclusion, the study showed that a variety of count models have been formulated and proposed to model count data but minimal research work has been done on the statistical modeling of count response data that contains spatial and temporal features. We proposed a Bayesian Negative Binomial model that accounts for and consider overdispersion and implemented two forms of the Negative Binomial model (Standard and Bayesian). While they may not be the comprehensive solution on how best to do it, the work sheds light on how the models behave under different circumstances.

Through the implementation of the "ADEMP" simulation structure, we were able to set aims of the simulation, generate data, set the estimand/target of analysis, use clear methods and evaluate model performance measures.

Considering the prior distributions of the data through Bayesian techniques, we showed instances where the Bayesian model outperformed the Negative Binomial Model. Combining these with optimized prediction intervals was useful in making more inclusive predictions, to allow room for planning and mitigation of fires in Kenya.

5.5 Recommendations for future work

The authors suggest the following steps for future work;

1. Extend the models in this proposed framework to account for more variables that may affect the fire regime

2. Incorporating a few outliers in the simulated datasets. This would be achieved through the inclusion of aspects of autocorrelation that exists in the type of data that are to be simulated.

3. Prediction of monthly fire frequency n-steps ahead where n is less than 6.

4. Refine the proposed models to optimize the performance metrics such as MASE and RMSE on the testing datasets. For example, include polynomials in the model specification. Also, describing the conditional distribution of the dependent variable given outcomes at previous time points will improve the work.

References

Avanzi, B., Taylor, G., Wong, B., and Xian, A. (2021). Modelling and understanding count processes through a markov-modulated non-homogeneous poisson process framework. *European Journal of Operational Research*, 290(1):177–195.

BahooToroody, A., Abaei, M. M., Arzaghi, E., Song, G., De Carlo, F., Paltrinieri, N., and Abbassi, R. (2020). On reliability challenges of repairable systems using hierarchical bayesian inference and maximum likelihood estimation. *Process Safety and Environmental Protection*, 135:157–165.

Boubeta, M., Lombardía, M., Marey-Pérez, F., and Morales, D. (2020). Area-level spatio-temporal poisson mixed models for predicting domain counts and proportions. *arXiv preprint arXiv:2012.00069*.

Boubeta, M., Lombardía, M. J., Marey-Pérez, M., and Morales, D. (2019). Poisson mixed models for predicting number of fires. *International journal of wildland fire*, 28(3):237–253.

Cai, Z., Jönsson, P., Jin, H., and Eklundh, L. (2017). Performance of smoothing methods for reconstructing ndvi time-series and estimating vegetation phenology from modis data. *Remote Sensing*, 9(12):1271.

Cao, Q., Zhang, L., Su, Z., Wang, G., Sun, S., and Guo, F. (2021). Comparing four regression techniques to explore factors governing the number of forest fires in southeast, china. *Geomatics, Natural Hazards and Risk*, 12(1):499–521.

Cracknell, M. J. and Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, 63:22–33.

de Lacalle, J. L. (2016). *stsm: Structural Time Series Models*. R package version 1.9.

de Oliveira, M. D., Colosimo, E. A., and Gilardoni, G. L. (2012). Bayesian inference for power law processes with applications in repairable systems. *Journal of Statistical Planning and Inference*, 142(5):1151–1160.

De Souza, R., Hilbe, J., Buelens, B., Riggs, J., Cameron, E., Ishida, E. E. d. O., Chies-Santos, A. L., and Killedar, M. (2015). The overlooked potential of generalized linear models in astronomy–iii. bayesian negative binomial regression and globular cluster populations. *Monthly Notices of the Royal Astronomical Society*, 453(2):1928–1940.

Dong, K., Zhao, H., Tong, T., and Wan, X. (2016). Nblda: negative binomial linear discriminant analysis for rna-seq data. *BMC bioinformatics*, 17(1):1–10.

Fick, S. E. and Hijmans, R. J. (2017). Worldclim 2: new 1-km spatial resolution climate surfaces for global land areas. *International journal of climatology*, 37(12):4302–4315.

Flannigan, M. D., Stocks, B. J., and Wotton, B. M. (2000). Climate change and forest fires. *Science of the total environment*, 262(3):221–229.

Fornacca, D., Ren, G., and Xiao, W. (2017). Performance of three modis fire products (mcd45a1, mcd64a1, mcd14ml), and esa fire_cci in a mountainous area of northwest yunnan, china, characterized by frequent small fires. *Remote Sensing*, 9(11):1131.

Giglio, L., Schroeder, W., and Justice, C. O. (2016). The collection 6 modis active fire detection algorithm and fire products. *Remote Sensing of Environment*, 178:31–41.

Goodrich, B., Gabry, J., Ali, I., and Brilleman, S. (2020). rstanarm: Bayesian applied regression modeling via stan. *R package version*, 2(1).

Grzenda, W. (2015). The advantages of bayesian methods over classical methods in the context of credible intervals. *Information systems in management*, 4.

Halim, S. Z., Janardanan, S., Flechas, T., and Mannan, M. S. (2018). In search of causes behind offshore incidents: Fire in offshore oil and gas facilities. *Journal of Loss Prevention in the Process Industries*, 54:254–265.

Halim, S. Z., Quddus, N., and Pasman, H. (2021). Time-trend analysis of offshore fire incidents using nonhomogeneous poisson process through bayesian inference. *Process Safety and Environmental Protection*, 147:421–429.

Hamner, B. and Frasco, M. (2018). *Metrics: Evaluation Metrics for Machine Learning*. R package version 0.1.4.

Hardin, J. W., Hardin, J. W., Hilbe, J. M., and Hilbe, J. (2007). *Generalized linear models and extensions*. Stata press.

Harris, I., Jones, P. D., Osborn, T. J., and Lister, D. H. (2014). Updated high-resolution grids of monthly climatic observations–the cru ts3. 10 dataset. *International journal of climatology*, 34(3):623–642.

Harvey, A. C. (1990). Forecasting, structural time series models and the kalman filter.

Hayat, M. J. and Higgins, M. (2014). Understanding poisson regression. *Journal of Nursing Education*, 53(4):207–215.

Hilbe, J. M. (2011). *Negative Binomial Regression*. Cambridge University Press.

Hilbe, J. M. (2014). *Modeling Count Data*. Cambridge University Press.

Huang, A. and Kim, A. (2021). Bayesian conway–maxwell–poisson regression models for overdispersed and underdispersed counts. *Communications in Statistics-Theory and Methods*, 50(13):3094–3105.

Kim, T., Lieberman, B., Luta, G., and Peña, E. A. (2021). Prediction intervals for poisson-based regression models. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1568.

Kwak, H., Lee, W.-K., Saborowski, J., Lee, S.-Y., Won, M.-S., Koo, K.-S., Lee, M.-B., and Kim, S.-N. (2012). Estimating the spatial pattern of human-caused forest fires using a generalized linear mixed model with spatial autocorrelation in south korea. *International Journal of Geographical Information Science*, 26(9):1589–1602.

Li, Y., Feng, Z., Chen, S., Zhao, Z., and Wang, F. (2020). Application of the artificial neural network and support vector machines in forest fire prediction in the guangxi autonomous region, china. *Discrete Dynamics in Nature and Society*, 2020.

Lim, C.-H., Kim, Y. S., Won, M., Kim, S. J., and Lee, W.-K. (2019). Can satellite-based data substitute for surveyed data to predict the spatial probability of forest fire? a geostatistical approach to forest fire in the republic of korea. *Geomatics, Natural Hazards and Risk*, 10(1):719–739.

Lima, C. H., AghaKouchak, A., and Randerson, J. T. (2018). Unraveling the role of temperature and rainfall on active fires in the brazilian amazon using a nonlinear poisson model. *Journal of Geophysical Research: Biogeosciences*, 123(1):117–128.

Lindén, A. and Mäntyniemi, S. (2011). Using the negative binomial distribution to model overdispersion in ecological count data. *Ecology*, 92(7):1414–1421.

Liu, D., Xu, Z., and Fan, C. (2019). Predictive analysis of fire frequency based on daily temperatures. *Natural Hazards*, 97(3):1175–1189.

Marchal, J., Cumming, S. G., and McIntire, E. J. (2017). Exploiting poisson additivity to predict fire frequency from maps of fire weather and land cover in boreal forests of québec, canada. *Ecography*, 40(1):200–209.

Morris, T. P., White, I. R., and Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in medicine*, 38(11):2074–2102.

NASA (2021). Firms.

NASA (2022). Fire information for resource management system (firms).

Olive, D. J., Rathnayake, R. C., and Haile, M. G. (2021). Prediction intervals for glms, gams, and some survival regression models. *Communications in Statistics-Theory and Methods*, pages 1–15.

Oliveira, S., Oehler, F., San-Miguel-Ayanz, J., Camia, A., and Pereira, J. M. (2012). Modeling spatial patterns of fire occurrence in mediterranean europe using multiple regression and random forest. *Forest Ecology and Management*, 275:117–129.

Payne, E. H., Hardin, J. W., Egede, L. E., Ramakrishnan, V., Selassie, A., and Gebregziabher, M. (2017). Approaches for dealing with various sources of overdispersion in modeling count data: Scale adjustment versus modeling. *Statistical methods in medical research*, 26(4):1802–1823.

Pereira, J. and Turkman, K. (2019). Statistical models of vegetation fires: Spatial and temporal patterns. In *Handbook of Environmental and Ecological Statistics*, pages 401–420. Chapman and Hall/CRC.

Pimont, F., Fargeon, H., Opitz, T., Ruffault, J., Barbero, R., Martin-StPaul, N., Rigolot, E., Rivière, M., and Dupuy, J.-L. (2021). Prediction of regional wildfire activity in the probabilistic bayesian framework of firelihood. *Ecological applications*, page e02316.

Ramapriyan, H. K., Behnke, J., Sofinowski, E., Lowe, D., and Esfandiari, M. A. (2010). *Evolution of the Earth Observing System (EOS) Data and Information System (EOSDIS)*, pages 63–92. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sayad, Y. O., Mousannif, H., and Al Moatassime, H. (2019). Predictive modeling of wildfires: A new dataset and machine learning approach. *Fire Safety Journal*, 104:130–146.

Schroeder, W., Oliva, P., Giglio, L., and Csiszar, I. A. (2014). The new viirs 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, 143:85–96.

Su, Z., Hu, H., Tigabu, M., Wang, G., Zeng, A., and Guo, F. (2019). Geographically weighted negative binomial regression model predicts wildfire occurrence in the great xing'an mountains better than negative binomial model. *Forests*, 10(5):377.

Trauernicht, C. (2019). Vegetation-rainfall interactions reveal how climate variability and climate change alter spatial patterns of wildland fire probability on big island, hawaii. *Science of The Total Environment*, 650:459–469.

Trocóniz, I. F., Plan, E. L., Miller, R., and Karlsson, M. O. (2009). Modelling overdispersion and markovian features in count data. *Journal of pharmacokinetics and pharmacodynamics*, 36(5):461–477.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Verdin, A., Funk, C., Peterson, P., Landsfeld, M., Tuholske, C., and Grace, K. (2020). Development and validation of the CHIRTS-daily quasi-global high-resolution daily temperature data set. 7(1):303.

Vilar, L., Gómez, I., Martínez-Vega, J., Echavarría, P., Riaño, D., and Martín, M. P. (2016). Multitemporal modelling of socio-economic wildfire drivers in central spain between the 1980s and the 2000s: comparing generalized linear models to machine learning algorithms. *PLoS One*, 11(8):e0161344.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Woo, H., Chung, W., Graham, J. M., and Lee, B. (2017). Forest fire risk assessment using point process modelling of fire occurrence and monte carlo fire simulation. *International journal of wildland fire*, 26(9):789–805.

Wooster, M. J., Roberts, G., Perry, G., and Kaufman, Y. (2005). Retrieval of biomass combustion rates and totals from fire radiative power observations: Frp derivation and calibration relationships between biomass consumption and fire radiative energy release. *Journal of Geophysical Research: Atmospheres*, 110(D24).

World Bank, G. (2021). World bank climate change knowledge portal: Kenya.

Xia, Y., Morrison-Beedy, D., Ma, J., Feng, C., Cross, W., and Tu, X. (2012). Modeling count outcomes from hiv risk reduction interventions: A comparison of competing statistical models for count responses. *AIDS research and treatment*, 2012.

Yadav, B., Jeyaseelan, L., Jeyaseelan, V., Durairaj, J., George, S., Selvaraj, K., and Bangdiwala, S. I. (2021). Can generalized poisson model replace any other count data models? an evaluation. *Clinical Epidemiology and Global Health*, page 100774.

Yu, J.-W., Tian, G.-L., and Tang, M.-L. (2007). Predictive analyses for nonhomogeneous poisson processes with power law using bayesian approach. *Computational Statistics & Data Analysis*, 51(9):4254–4268.

Appendix A

R Code

The R code used for graphing and simulations in Chapter 3 and 4.

A.1 Simulation code

A.1.1 Data generation

```
# Simulate climate data
library(truncnorm)
library(MASS)
#temper <-
 # function(n = 218, n2 = 2, thet = 1.5){
    # Set empty list
  #  daf = list()


   # for(i in 1:n2){
    #  x1 = rtruncnorm(n, a = 0, b = 140, mean = 50, sd = 3)
     # x2 = rtruncnorm(n, a = 15, b = 40, mean = 18, sd = 2)
      #x3 = rtruncnorm(n, a = 8, b = 24, mean = 15, sd = 0.9)
      # seq
      #timer = c(11,12,1,2,3,4,5,6,7,8,9,10)
      # create data
      #df <- data.frame(
       # rainfall = round(x1,1),
```

```
        #min_temp = round(x2,1),
        #max_temp = round(x3,1),
        #month = rep(timer, n/12, length.out = n)
      #)
      #daf[[i]] <- df}
    #daf
  #}

# library(purrr)
# rdunif(12, 12, 1)
#-----------------------------------------------------
# Generate time series
timeGad <-
  function(nt = 24, frequencyt = 12, thet = 1.5){
    library(stsm)
    # generate a  series from a local level plus seasonal model
    pars <- c(var1 = 60)
    parsl <- c(var1 = 0)
    parsu <- c(var1 = 150)
    m <- stsm.model(model = "llm+seas", y = ts(seq(nt),
     frequency = frequencyt),
                    pars = pars, nopars = NULL, lower = parsl,
                    upper = parsu)
    ss <- char2numeric(m)

    y <- datagen.stsm(n = nt, model = list(Z = ss$Z, T = ss$T,
    H = ss$H, Q = ss$Q),
                      n0 = 20, freq = frequencyt,
                       old.version = TRUE)$data
    #plot(y, main = "data generated from the local-level
```

```
 plus seasonal component")
# Add a large constant to remove negative values
rain  = as.data.frame(y+50+abs(min(y)))

#-----------------------------------------------------
# generate a monthly series for minimum temperature

pars_m <- c(var1 = 15)
parsl_m <- c(var1 = 12)
parsu_m <- c(var1 = 22)
m_m <- stsm.model(model = "llm+seas", y = ts(seq(nt),
 frequency = frequencyt),
                  pars = pars_m, nopars = NULL, lower = parsl_m,
                  upper = parsu_m)
ss_m <- char2numeric(m_m)

y_m <- datagen.stsm(n = nt, model = list(Z = ss_m$Z,
                                         T = ss_m$T, H = ss_m$H,
                                         Q = ss_m$Q),
                    n0 = 20, freq = frequencyt,
                     old.version = TRUE)$data

# Add a large constant to remove negative values
min_temp  = as.data.frame(y_m+12+abs(min(y_m)))

#-----------------------------------------------------

# generate a monthly series for minimum temperature

pars_mx <- c(var1 = 25)
```

```
  parsl_mx <- c(var1 = 20)
  parsu_mx <- c(var1 = 40)
  m_mx <- stsm.model(model = "llm+seas", y = ts(seq(nt),
   frequency = frequencyt),
                     pars = pars_mx, nopars = NULL, lower = parsl_mx,
                     upper = parsu_mx)
  ss_mx <- char2numeric(m_mx)

  y_mx <- datagen.stsm(n = nt, model = list(Z = ss_mx$Z,
                                            T = ss_mx$T, H = ss_mx$H,
                                            Q = ss_mx$Q),
                       n0 = 20, freq = frequencyt,
                        old.version = TRUE)$data

  # Add a large constant to remove negative values
  max_temp  = as.data.frame(y_mx+20+abs(min(y_mx)))

  # generate a series following a NB
  y_count <- rnegbin(nt, mu = 277.1284, theta = thet)

  # Combine all data
  dataclean = cbind(rainfall = rain,
                    min_temp = min_temp,
                    max_temp = max_temp,
                    y_count = y_count)
  # Result data
  colnames(dataclean) <- c("rainfall","min_temp", "max_temp", "count")
  dataclean
}
```

```
# Simulate climate data

temper <-
  function(n = 10, n2 = 60, theta = 1.5, pathway = "simulated_data/"){
    # Set empty list
    daf = list()

    for(i in 1:n){
      # create data
      df <- timeGad(nt=n2, thet = theta)
      daf[[i]] <- df}
    # Write out list as csv files
    for(i in 1:length(daf)){
      write.csv(data.frame(daf[[i]]),
                file = paste0(pathway, n2,"_",i, '.csv'))
    }

  }

# Generate data (theta = 1.5,)
## 5 year data
set.seed(76568)
temper(n=1000, n2 = 60, theta = 1.5,
pathway = "simulated_data/d5year/theta_1.5/")

## 10 year data
temper(n=1000, n2 = 120, theta = 1.5,

pathway = "./simulated_data/d10year/theta_1.5/")
```

```
## 20 year data
temper(n=1000, n2 = 240, theta = 1.5,
 pathway = "simulated_data/d20year/theta_1.5/")

## 30 year data
temper(n=1000, n2 = 360, theta = 1.5,
pathway = "simulated_data/d30year/theta_1.5/")

# Generate data (theta = 5)
## 5 year data
set.seed(76568)
temper(n=1000, n2 = 60, theta = 5,
 pathway = "simulated_data/d5year/theta_5/")

## 10 year data
temper(n=1000, n2 = 120, theta = 5,
 pathway = "simulated_data/d10year/theta_5/")

## 20 year data
temper(n=1000, n2 = 240, theta = 5,
 pathway = "simulated_data/d20year/theta_5/")

## 30 year data
temper(n=1000, n2 = 360,  theta = 5,
pathway = "simulated_data/d30year/theta_5/")

# Generate data (theta = 10)
## 5 year data
set.seed(76568)
```

```
temper(n=1000, n2 = 60, theta = 10,
 pathway = "simulated_data/d5year/theta_10/")

## 10 year data
temper(n=1000, n2 = 120, theta = 10,
 pathway = "simulated_data/d10year/theta_10/")

## 20 year data
temper(n=1000, n2 = 240, theta = 10,
 pathway = "simulated_data/d20year/theta_10/")

## 30 year data
temper(n=1000, n2 = 360, theta = 10,
 pathway = "simulated_data/d30year/theta_10/")

# Generate data (theta = 100)
## 5 year data
set.seed(76568)
temper(n=1000, n2 = 60, theta = 100,
 pathway = "simulated_data/d5year/theta_100/")

## 10 year data
temper(n=1000, n2 = 120, theta = 100,
 pathway = "simulated_data/d10year/theta_100/")

## 20 year data
temper(n=1000, n2 = 240, theta = 100,
 pathway = "simulated_data/d20year/theta_100/")

## 30 year data
```

```
temper(n=1000, n2 = 360, theta = 100,
 pathway = "simulated_data/d30year/theta_100/")
```

A.2 Analysis code

```
%#*************************#
# Script to fit models on data
library(MASS)
library(stsm)
# Load libraries ----
library(rstanarm)
library(brms)  # for models
library(bayesplot)
library(ggplot2)
library(dplyr)
library(tidybayes)
library(modelr)
#library(tidyverse)
library(caret)
library(readr)
library(purrr)
library(parallel)

#############################################################################
# FITTING STANDARD NEGATIVE BINOMIAL MODEL
#############################################################################
```

```
negbinner <- function(x, theta = 1.5, n = 60){

  # create training and test sets
  # set seed
  set.seed(456)



  trainIndex <- round(0.8*length(x$count))
  # Create the data sets
  fireTrain <- x[1:trainIndex,]
  fireTest  <- x[(trainIndex+1):n,]

  # Fit model on training set

  glmNB <- MASS::glm.nb(count ~ max_temp +
                          min_temp + rainfall, data = fireTrain,
                        link = "log")
  # Predict on training set
  predictions_train <- predict(glmNB,
                          newdata = fireTrain, type = "response")
  # Predict on testing set
  predictions_test <- predict(glmNB,
                          newdata = fireTest, type = "response")
  # get the rmse
  train_rmse <- caret::RMSE(round(predictions_train),fireTrain$count)

  test_rmse <- caret::RMSE(round(predictions_test),fireTest$count)

  # get the MASE
  test_mase <- Metrics::mase(actual = fireTest$count,
```

```
                              predicted = round(predictions_test))

  # Get the bias
  test_bias <- Metrics::bias(actual = fireTest$count,
                              predicted = round(predictions_test))
  # Dispersion parameter

  cbind(rmse_train = train_rmse,
rmse_test = test_rmse, mase_test = test_mase,
        bias_test = test_bias,
        theta = theta, n = n)

}

##############################################################################
# FITTING BAYESIAN NEGATIVE BINOMIAL MODEL
##############################################################################

stanbinner <- function(x, theta = 1.5, n = 60){

  # create training and test sets
  # set seed
  set.seed(456)

  # Add time by month index
  x$time <- rep(1:12, length.out = n)


  trainIndex <- round(0.8*length(x$count))
  # Create the datasets
```

```
fireTrain <- x[1:trainIndex,]
fireTest  <- x[(trainIndex+1):n,]

# Add prior means

get_prior_means <- function(x){
  library(dplyr)
  x %>% group_by(time) %>%
    summarize(count_mean = mean(count)) %>%
    data.frame()
}

p_means = get_prior_means(fireTrain)

# Join means to train data
fireTrain2 <- fireTrain %>%
  inner_join(p_means, by = "time")

# Join means to test data
fireTest2 <- fireTest %>%
  inner_join(p_means, by = "time")



# Fit model on training set

stanNB <- rstanarm::stan_glm.nb(count ~  max_temp+
                                  min_temp + rainfall + count_mean,
                                data = fireTrain2,
                                link = "log")
# Predict on training set
```

```
  predictions_train <- predict(stanNB,
                               newdata = fireTrain2,
                                type = "response")
  # Predict on testing set
  predictions_test <- predict(stanNB,
                              newdata = fireTest2,
                               type = "response")
  # get the rmse
  train_rmse <- caret::RMSE(round(predictions_train),fireTrain2$count)

  test_rmse <- caret::RMSE(round(predictions_test),fireTest2$count)

  # get the MASE
  test_mase <- Metrics::mase(actual = fireTest2$count,
                             predicted = round(predictions_test))
  # Get the bias
  test_bias <- Metrics::bias(actual = fireTest2$count,
                             predicted = round(predictions_test))
  # Dispersion parameter

  cbind(rmse_train = train_rmse,
rmse_test = test_rmse, mase_test = test_mase,
        bias_test = test_bias,
        theta = theta, n = n)

}
##############################################################################

# Script to run models on 5 year data (Interchangeable)
```

```
# Source model formulas

source('final_models.R')

# Set seed
set.seed(76568)

# Run standard Negative Binomial model
##############################################################################
# Results for 5 year - 1.5
five_year1.5 <- list.files(path = "./simulated_data/d5year/theta_1.5",
                           # Identify all csv files in folder
                           pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%             # Store all files in list
  map(negbinner, n = 60)
five_year1.5 <- do.call(rbind, five_year1.5)
# Combine data sets into one data set
# check the data
head(five_year1.5)
# write.csv
write.csv(five_year1.5, "./model_results/five_year_1.5_metrics.csv")
#_____________________________________________________________________

# Results for 5 year - 5
five_year5 <- list.files(path = "./simulated_data/d5year/theta_5",
                         # Identify all csv files in folder
                         pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%             # Store all files in list
  map(negbinner, n = 60, theta = 5)
five_year5 <- do.call(rbind, five_year5)
```

```
# Combine data sets into one data set
# check the data
head(five_year5)
# write.csv
write.csv(five_year5, "./model_results/five_year_5_metrics.csv")

#------------------------------------------------------------------------

# Results for 5 year - 10
five_year10 <- list.files(path = "./simulated_data/d5year/theta_10",
                          # Identify all csv files in folder
                          pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%              # Store all files in list
  map(negbinner, n = 60, theta = 10)
five_year10 <- do.call(rbind, five_year10)
# Combine data sets into one data set
# check the data
head(five_year10)
# write.csv
write.csv(five_year10, "./model_results/five_year_10_metrics.csv")

#------------------------------------------------------------------------

# Results for 5 year - 100
five_year100 <- list.files(path = "./simulated_data/d5year/theta_100",
                           # Identify all csv files in folder
                           pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%            # Store all files in list
  map(negbinner, n = 60, theta = 100)
five_year100 <- do.call(rbind, five_year100)
```

```
# Combine data sets into one data set
# check the data
head(five_year100)
# write.csv
write.csv(five_year100, "./model_results/five_year_100_metrics.csv")##

##### BAYESIAN MODEL

# Run Bayesian Negative Binomial model
#############################################################################
# Results for 5 year - 1.5
five_year1.5b <- list.files(path = "./simulated_data/d5year/theta_1.5",
                            # Identify all csv files in folder
                            pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%             # Store all files in list
  map(stanbinner, n = 60)
five_year1.5b <- do.call(rbind, five_year1.5b)
# Combine data sets into one data set
# check the data
head(five_year1.5b)
# write.csv
write.csv(five_year1.5b, "./model_results/five_year_1.5b_metrics.csv")
#_____________________________________________________________________

# Results for 5 year - 5
five_year5b <- list.files(path = "./simulated_data/d5year/theta_5",
                          # Identify all csv files in folder
                          pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>% # Store all files in list
  map(stanbinner, n = 60, theta = 5)
```

```
five_year5b <- do.call(rbind, five_year5b)
# Combine data sets into one data set
# check the data
head(five_year5b)
# write.csv
write.csv(five_year5b, "./model_results/five_year_5b_metrics.csv")

#------------------------------------------------------------------------

# Results for 5 year - 10
five_year10b <- list.files(path = "./simulated_data/d5year/theta_10",
                           # Identify all csv files in folder
                           pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%     # Store all files in list
  map(stanbinner, n = 60, theta = 10)
five_year10b <- do.call(rbind, five_year10b)
# Combine data sets into one data set
# check the data
head(five_year10b)
# write.csv
write.csv(five_year10b, "./model_results/five_year_10b_metrics.csv")

#------------------------------------------------------------------------

# Results for 5 year - 100
five_year100b <- list.files(path = "./simulated_data/d5year/theta_100",
                            # Identify all csv files in folder
                            pattern = "*.csv", full.names = TRUE) %>%
  mclapply(read_csv) %>%             # Store all files in list
  map(stanbinner, n = 60, theta = 100)
```

```
five_year100b <- do.call(rbind, five_year100b)
# Combine data sets into one data set
# check the data
head(five_year100b)
# write.csv
write.csv(five_year100b, "./model_results/five_year_100b_metrics.csv")
##############################################
```

A.3 Visualizaton code

```
# Script to start out the results section
library(tidyverse)
# read data
fire_clim <- read_csv('fire_data_2000-18.csv')
```r}
# Find the summary stats
library(rstatix)
library(flextable)
fire_clim[,-1] %>%
  # remove unwanted variables
  select(-month, -year, -average_temp, -mean_bright31,
   -mean_brightness,
         -mean_frp, -anomaly) %>%
  get_summary_stats() %>%
  select(-q1,-q3,-iqr,-mad,-ci,-se)
```
```

```
```{r}
# number of fires
## Create a time series object
count_ts <- fire_clim |>
  mutate(Time = paste(year,month,sep = "-")) |>
  mutate(Time = zoo::as.yearmon(Time))

# Plot the time series
plot_count <-
ggplot(count_ts, aes(Time, count )) +  geom_line(col = "red") +
  scale_x_continuous(breaks = seq(2000,2018,5)) + theme_bw() +
  labs(title = "a. Monthly fire frequency trend from 2000
   to 2018 in Kenya")+
  ylab("Number of fires")+
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14),
        legend.position = "none")

plot_count
```
```{r}
# max temperature

# Plot the time series
plot_max <-
ggplot(count_ts, aes(Time, mean_max_temp)) +  geom_line( col = "brown") +
```
```

```
  scale_x_continuous(breaks = seq(2000,2018,5)) + theme_bw() +
  labs(title =
"b. Monthly maximum temperature trend from 2000 to 2018
in Kenya")+
  ylab("Maximum temperature (\u00B0C)")+
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14),
        legend.position = "none")
plot_max
```
```
```{r}
# min temperature

# Plot the time series
plot_min <-
ggplot(count_ts, aes(Time, mean_min_temp)) +  geom_line(col = "blue") +
  scale_x_continuous(breaks = seq(2000,2018,5)) + theme_bw() +
  labs(title = "c. Monthly minimum temperature trend from 2000 to 2018
   in Kenya")+
  ylab("Minimum temperature (\u00B0C)")+
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14),
        legend.position = "none")
plot_min
```
```

```
```{r}
# rainfall

# Plot the time series
plot_rain <-
ggplot(count_ts, aes(Time, mean_rainfall)) +  geom_line(col = "orange") +
  scale_x_continuous(breaks = seq(2000,2018,5)) + theme_bw() +
  labs(title = "d. Monthly rainfall trend from 2000 to 2018 in Kenya")+
  ylab("Rainfall (mm)")+
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14))

plot_rain
```
```{r, fig.width=8, fig.height=8}
# Combine the plots
library(patchwork)
(plot_count + plot_max) / (plot_min + plot_rain)
```
Simulation Results

Scenario 1 (Theta = 1.5, 5, 10, 100)

Sample size n = 60, 120, 240, 360 (Reproducible by changing names)

```{r}
```
```

```
# Read in the four time periods
five_a <- read_csv("./model_results/five_year_1.5_metrics.csv")
# add mdtype column
five_a$model <- "NB"

five_b <- read_csv("./model_results/five_year_1.5b_metrics.csv")
# add mdtype
five_b$model <- "BNB"
# combine the two
five_pnt1.5 <- rbind(five_a, five_b)
head(five_pnt1.5)
```

```{r, fig.width=8, fig.height=6}
# Long format
library(tidyr)
five_pnt1.5 %>%
  pivot_longer(cols = rmse_train:bias_test, names_to = "metric",
               values_to = "value") %>%
  # Plot line graphs
  ggplot(aes(y = value, x = ...1, group = model, col = model))+
  geom_line() +
  facet_wrap(~ metric, scales = "free") +
  # add theme
  theme_bw()+
  # add labels
  labs(title = "Comparing the Bayesian Negative Binomial
   model and the Standard Negative Binomial model",
       subtitle = "Comparison of four metrics with
       sample size n = 60,  \u03B8 = 1.5",
```

```
       x = "Dataset number")
```

```{r}
library(flextable)
five_pnt1.5 %>%
  pivot_longer(cols = rmse_train:bias_test, names_to = "metric",
               values_to = "value") %>%
  group_by(metric,model) %>%
  summarise(mean_val = mean(value)) %>%
  spread(key = model, value = mean_val) %>%
  mutate_if(is.numeric,round, 2) %>%
  regulartable()
```

```{r}
# On real data
# Negbinner
negbinner2 <- function(x, prop = 0.8){

  # create training and test sets
  # set seed
  set.seed(456)

  n = length(x$count)
  trainIndex <- round(prop*n)
  # Create the data sets
  fireTrain <- x[1:trainIndex,]
  fireTest  <- x[(trainIndex+1):n,]

  # Fit model on training set
```

```
glmNB <- MASS::glm.nb(count ~ mean_max_temp +
                        mean_min_temp + mean_rainfall,
                         data = fireTrain,
                      link = "log")
# Predict on training set
predictions_train <- predict(glmNB,
                        newdata = fireTrain, type = "response")
# Predict on testing set
predictions_test <- predict(glmNB,
                        newdata = fireTest, type = "response")
# get the rmse
train_rmse <- caret::RMSE(round(predictions_train),fireTrain$count)

test_rmse <- caret::RMSE(round(predictions_test),fireTest$count)

# get the MASE
test_mase <- Metrics::mase(actual = fireTest$count,
                           predicted = round(predictions_test))

# Get the bias
test_bias <- Metrics::bias(actual = fireTest$count,
                           predicted = round(predictions_test))
# Dispersion parameter

df = cbind(rmse_train = train_rmse,
 rmse_test = test_rmse, mase_test = test_mase,
      bias_test = test_bias, n = n, prop = prop)

df
```

```
}
```
```

```{r}
stanbinner2 <- function(x, prop = 0.8){

  # create training and test sets
  # set seed
  set.seed(456)
  n = length(x$count)

  # Add time by month index
  x$time <- rep(1:12, length.out = n)



  trainIndex <- round(prop*n)
  # Create the datasets
  fireTrain <- x[1:trainIndex,]
  fireTest  <- x[(trainIndex+1):n,]

  # Add prior means

  get_prior_means <- function(x){
    library(dplyr)
    x %>% group_by(month) %>%
      summarize(count_mean = mean(count)) %>%
      data.frame()
  }

  p_means = get_prior_means(fireTrain)
```
```

```
# Join means to train data
fireTrain2 <- fireTrain %>%
  inner_join(p_means, by = "month")

# Join means to test data
fireTest2 <- fireTest %>%
  inner_join(p_means, by = "month")



# Fit model on training set

stanNB <- rstanarm::stan_glm.nb(count ~  mean_max_temp+
                                  mean_min_temp +
                    mean_rainfall + count_mean,
                                data = fireTrain2,
                                link = "log")
# Predict on training set
predictions_train <- predict(stanNB,
                             newdata = fireTrain2, type = "response")
# Predict on testing set
predictions_test <- predict(stanNB,
                            newdata = fireTest2, type = "response")
# get the rmse
train_rmse <- caret::RMSE(round(predictions_train),fireTrain2$count)

test_rmse <- caret::RMSE(round(predictions_test),fireTest2$count)

# get the MASE
test_mase <- Metrics::mase(actual = fireTest2$count,
```

```
                              predicted = round(predictions_test))
  # Get the bias
  test_bias <- Metrics::bias(actual = fireTest2$count,
                              predicted = round(predictions_test))
  # Dispersion parameter

  df = cbind(rmse_train = train_rmse,
   rmse_test = test_rmse, mase_test = test_mase,
        bias_test = test_bias, n = n, prop = prop)

  df
}
```

```{r, message=FALSE}
# Script to fit models on data
library(MASS)
library(stsm)
# Load libraries ----
library(rstanarm)
library(brms)  # for models
library(bayesplot)
library(ggplot2)
library(dplyr)
library(tidybayes)
library(modelr)
#library(tidyverse)
library(caret)
library(readr)
library(purrr)
```

```
library(parallel)
# read in the data
series_data <- read_csv("fire_data_2000-18.csv")

# Set seed
set.seed(76568)

# Standard NB
nb_result80 <- negbinner2(series_data, prop = 0.8)
nb_result90 <- negbinner2(series_data, prop = 0.9)
nb_result95 <- negbinner2(series_data, prop = 0.95)

# Bayesian NB
bnb_result80 <- stanbinner2(series_data, prop = 0.8)
bnb_result90 <- stanbinner2(series_data, prop = 0.9)
bnb_result95 <- stanbinner2(series_data, prop = 0.95)
```

```{r}
rbind(data.frame(nb_result80),data.frame(bnb_result80)) %>%
  data.frame() %>%
  mutate_if(is.numeric, round, 2)
```

```{r}
rbind(bnb_result80,bnb_result90, bnb_result95) %>%
  data.frame()
```

```{r}
# prediction intervals
intervals_bnb <- predictive_interval(stanNB, newdata = fireTest2)
```

```
# add test data
preds_intervals <- cbind(intervals_bnb, fireTest2$count)
# Write
write.csv(preds_intervals, "prediction_intervals.csv")
```

```{r}
# Review prediction intervals
pred_int <- read.csv("prediction_intervals.csv")
# Rename variables
pred_int$lower <-  pred_int$X5.
pred_int$upper <- pred_int$X95.
pred_int$actual <-  pred_int$X.1
pred_int <- pred_int[,-c(2,3,4)]
head(pred_int)
```

```{r}
# plot a prediction interval curve
pred_int %>%
  pivot_longer(lower:actual, names_to = "interval",
               values_to = "Count") %>%
  ggplot(aes(x = X, y = Count, group = interval, col = interval)) +
  geom_line() +
  labs(title = "Prediction intervals of the BNB model
 compared to the actuals",
subtitle = "Values obtained from the predictions on the
testing dataset at 0.9 desired probability mass")+
  theme_bw()
```

Appendix B

Timelines

B.1 Sep 2021-Jan 2022

1. Conduct a constant, in-depth assessment of the literature to discover knowledge gaps and experts in the topic. s
2. Determine the project's precise goals based on my research vision, plan, preliminary data, and literature review findings.

B.2 Feb 2022 - March 2022

3. Prepare a proposal draft.
4. Set aside the proposal draft for a while, then revise it.
5. Get independent readers to review and comment on my proposal draft. This includes my supervisors
6. Rewrite and rewrite the proposal in response to reader feedback (process continues until close to proposal submission)
7. Complete the proposal budget and reasoning.
8. Write abstract/summary and share with supervisors for further comments
9. Submit proposal to SIMS after approval and signatures by supervisors

B.3 March 2022 - May 2022

10. Conduct analyses and write up the results section
11. Get supervisors to review and comment on my thesis draft.
12. Write and rewrite thesis as per comments from supervisors
13. Write discussion and conclusion section. Make any updates and changes to other chapters
14. Get supervisors to review and comment on my thesis draft.
15. Write and rewrite thesis as per comments from supervisors
16. Submit the approved thesis to SIMS

Appendix C

Budget

Table C.1: Project budget.

Item	Total
Stationery	5,000
Internet	16,400
Printing	2,000
Lunch	3,000
Total	**26,400**

Appendix D

Ethical Approval

Strathmore UNIVERSITY

24th May 2022

Mr Owiti Levi,
leviorero.owiti@strathmore.edu

Dear Mr Owiti,

RE: Modeling of Count Data with an Informative Time Component in the Presence of Overdispersion

This is to inform you that SU-IERC has reviewed and **approved** your above **SU Masters'** research proposal. Your application reference number is **SU-IERC1338/22.** The approval period is **24th May 2022 to 23rd May 2023.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

Appendix E

Turnitin Similarity Report

Curiginal

Document Information

Analyzed document	Owiti Thesis.pdf (D138302638)
Submitted	2022-05-28T15:20:00.0000000
Submitted by	
Submitter email	Leviorero.Owiti@strathmore.edu
Similarity	1%
Analysis address	library.strath@analysis.urkund.com

Sources included in the report

W	URL: https://worldwidescience.org/topicpages/n/negative+binomial+regression.html Fetched: 2019-12-15T20:09:24.0100000	2
W	URL: https://forestry.ubc.ca/faculty-profile/guangyu-wang/ Fetched: 2021-11-19T05:53:21.3730000	2
W	URL: https://www.frontiersin.org/articles/10.3389/fmicb.2021.711861/full Fetched: 2022-05-28T15:20:42.0970000	1
W	URL: https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=osu1543573678017356&disposition=inline Fetched: 2022-05-09T19:56:11.0300000	4

Curiginal

Entire Document

Modeling of Count Data with an Informative Time Component in the Presence of Overdispersion Owiti, Levi Alfred Orero Submitted in partial fulfillment of the requirements for the degree of Master of Science in Statistical Science of Strathmore University Institute of Mathematical Sciences Strathmore University Nairobi, Kenya May 2022 This thesis is available for Library use through open access on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgment.

Declaration I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself. © No part of this thesis may be reproduced without the permission of the author and Strathmore University. Name: ..... Owiti, Levi Alfred Orero ..... Signature: ..... Date: ..... May 28, 2022 ..... Approval The thesis of Owiti, Levi Alfred Orero was reviewed and approved by the following: Dr. Evans Omondi Supervisor, Institute of Mathematical Sciences, Strathmore University. Professor Benard Omolo Supervisor, Institute of Mathematical Sciences, Strathmore University. Dr. Godfrey Madigu Dean, Institute of Mathematical Sciences, Strathmore University. Dr. Bernard Shibwabo Director, Office of Graduate Studies, Strathmore University. ii

Abstract In real-world count data, several methods have been applied to handle the common problem of overdispersion. However, these methods have not comprehensively considered unique features that may exist in the data. This study sought to address robust statistical modeling of count response data that contains temporal features. We proposed a Bayesian Negative Bino- mial model that will handle overdispersion while taking into account the temporal features of the data. Two count data models were compared and extended to incorporate an informative time component. To test the various models, this study conducted simulation studies under specified parameters to examine how the models behave under certain conditions. We used a data generation mechanism that ensured our simulated data had seasonality as is with our real-world data on fire frequency, temperature, and rainfall. Further, we examined the effect of the additional components on prediction intervals of the simulation studies for the different count models. The introduction of Bayesian techniques into our modeling was intended to create more accurate prediction intervals that take account of the prior distribution of the data. Our Bayesian Negative Binomial model was better than the Negative Binomial model in terms of model bias. When validated on real data to confirm its effectiveness, the Bayesian model had better MASE and the prediction intervals enveloped the actual data in the testing dataset of fires in Kenya between the year 2000 and 2018.
Keywords: overdispersion, Bayesian, Negative Binomial distribution, count data, informa- tive component, spatiotemporal data iii

Table of contents List of figures vii List of tables x List of abbreviations xi Acknowledgement xii Dedication xiii 1 Introduction 1 1.1 Background to the study . . . . . 1 1.2 Example of count data . . . . . 2 1.3 Statement of the problem . . . . . 2 1.4 Objectives of the study . . . . . 4 1.4.1 General objective . . . . . 4 1.4.2 Specific objectives . . . . . 4 1.5 Scope of the study . . . . . 4 1.6 Significance of the study . . . . . 5 1.7 Limitations of the study . . . . . 5 1.8 Thesis outline . . . . . 5 2 Literature review 7 2.1 Introduction . . . . . 7 2.2 Applications in fire management systems . . . . . 7 2.2.1 MODIS Fire Products 8

2.2.2 Rainfall and Temperature . . . . . 8 2.3 Informative time and spatial components . . . . . 8 2.4 Constructing prediction intervals . . . . . 11 2.5 Gaps identified . . . . . 12 3 Methodology 13 3.1 Introduction . . . . . 13 3.2 Data collection . . . . . 13 3.2.1 Fire data . . . . . 13 3.2.2 Rainfall data . . . . . 13 3.2.3 Temperature data . . . . . 14 3.3 Simulation studies . . . . . 15 3.3.1 Aims of the simulation . . . . . 15 3.3.2 Data generating mechanisms . . . . . 15 3.3.3 Target of analysis . . . . . 16 3.3.4 Methods to be evaluated . . . . . 16 3.3.5 Performance measures . . . . . 17 3.4 Statistical models and estimation . . . . . 17 3.4.1 Standard Negative Binomial Model . . . . . 17 3.4.2 Bayesian Negative Binomial MCMC Model . . . . . 18 4 Results 20 4.1 Introduction . . . . . 20 4.2 Descriptive statistics . . . . . 20 4.2.1 Fire and climate data . . . . . 20 4.3 Simulation studies . . . . . 22 4.3.1 Scenario 1: θ = 1.5 . . . . . 22 4.3.2 Scenario 2: θ = 5 28