A Nonlinear Regression Model for Modeling Autocorrelated and Overdispersed Count Data

A Thesis Submitted to the Department of Statistics, University of Dhaka, as a Partial Fulfillment of the Requirements for the Degree of Master of Science

M.S. Thesis

Submitted by

Registration Number: 2017-113-960

Examination Roll: 144210

Session: 2021-22

Department of Statistics

University of Dhaka

September, 2024

Acknowledgement

All praise is due to Allah, the Almighty, the Most Merciful, who deserves all credit for any good that has come my way. I am deeply thankful to Allah for blessing me with the opportunity, health, and ability to successfully complete this thesis.

I would like to express my gratitude to my research supervisor, **Dr. Anamul Haque Sajib**, Associate Professor, Department of Statistics, University of Dhaka. His unwavering guidance, insightful critiques, and constant encouragement to explore new ideas have been instrumental in the completion of this thesis and in my growth as a researcher. I am truly appreciative of his dedication to making time for me despite his demanding schedule.

I also extend my heartfelt thanks to **Dr. Jafar Ahmed Khan**, Professor, Department of Statistics, University of Dhaka, for graciously permitting me to use the seminar facilities that were essential for the effective conduct of this research.

To my dearest friends, your moral support, companionship, and shared moments of both challenge and joy have made this journey far more manageable and rewarding. I am deeply grateful for your presence in my life.

Finally, I wish to convey my profound love to my mother, **Mahbuba Khanam**, and my sister, **Mahjabin Monisha**, whose prayers and unwavering encouragement have been my source of strength during the most challenging times.

Mahjaerin Onnesha
September, 2024.

Abstract

Count data collected over time are autocorrelated as factors which affect the count are autocorrelated. For example, dengue cases of a specific day depend not only on the factors but also the dengue cases of preceding days. Furthermore, the factors may be linked with response in a nonlinear way and count may be overdispersed due to autocorrelation. The generalized additive model (GAM) is the generalization of GLM which handles all types of relationships but not autocorrelation (overdispersion). Lei et al. (2012) extended GAM by incorporating autocorrelation terms of both factors and response as independent variable in GAM framework known as GAMAR for dealing autocorrelated and overdispersed count data. The performance of their model was investigated through simulation study where one single lag and one single nonlinear functional form were considered. However, how their model performs for other different lag values and functional forms is not explored yet. Motivated by these lacks, an extensive simulation study is conducted in this thesis where all scenarios are covered. From the simulation study, it is observed that GAMAR offers better performance compared to GAM irrespective of the lag values and functional forms. Finally, the relationship between dengue cases and environmental factors in Dhaka, Bangladesh is analyzed using the GAMAR.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Count data emerge from observations obtained through counting, such as, the number of road accidents in a day, the number of dengue patients admitted to the clinic in a week, and so on. Time series count data refers to a type of data where observations are collected sequentially over time, and each observation represents a count of events or occurrences within a specified time interval. Time series of observed counts can be found in many different contexts, such as, studies concerning the prevalence of a certain disease [1, 2] or of discrete transaction price movements on financial markets [3, 4].

Count data are usually analyzed by using the Poisson regression model under the framework of Generalized Linear Models (GLMs) [5]. This approach assumes that its expected value is a linear combination of predictor variables linked through a logarithmic function. However, in fields like economics and environmental sciences, non-linear relationships often exist between predictors and the response variable. To address nonlinear relationships, Trevor et al. (1990) introduced the Generalized Additive Model (GAM), which allows for flexible modeling of non-linear effects using smooth functions of explanatory variables while maintaining interpretability [6]. Time series count data exhibit temporal correlation, where counts at adjacent time points are often dependent. For instance, the number of dengue cases or deaths on a particular day may depend on the previous day's counts. Additionally, environmental factors such as temperature and humidity, which influence disease incidence, also exhibit temporal correlation over consecu-

tive time periods. Overdispersion is another common issue, often related to sources of autocorrelation [7]. When analyzing time series count data, it is crucial to incorporate this correlation to obtain robust estimates. Furthermore, autocorrelation poses challenges in estimating GLM and GAM because these models depend on the assumption that each observation is independently distributed. When this assumption is violated, it can result in problematic estimates, even in straightforward scenarios. For instance, if the error terms in a linear regression model are actually positively autocorrelated, failing to address this issue may lead to underestimating the standard errors of the estimated regression coefficients [8]. In many studies, the response variable may also exhibit autocorrelation, which must be accounted for in modeling. Standard GLM and GAM models relate the response variable to explanatory variables but do not consider its dependence on past values [9].

Lei et al. (2012) expanded the GAM model with autoregressive terms (GAMAR) by incorporating the auto regressive correlation structure of both the response and explanatory variables, using Generalized Autoregressive Moving Average (GARMA) models [10]. In contrast to GARMA, GAMAR does not include moving average terms and generalizes the linear components to natural splines. GAMAR provides two key advantages over GAM: It offers generalized time series analysis rather than probabilistic modeling, and its autoregressive (AR) component effectively models and explains the autocorrelation within observations. This results in GAMAR's Pearson residuals resembling white noise more closely than those of GAM, leading to more reliable estimations.

In their simulation study, Lei et al. examined a specific lag and a single functional form without exploring how varying sample sizes and autoregressive (AR) orders might influence the model's performance. An extensive simulation study has been conducted to examine the performance of GAM and GAMAR.

Dengue fever is an infectious disease transmitted through the bite of an infected Aedes mosquito and is caused by one of four distinct serotypes of the dengue

virus (DENV 1-4) [11]. Dengue fever occurs in both urban and semi-urban areas throughout the tropics and sub-tropics, putting more than half of the global population at risk. Annually, there are over 400 million documented cases of dengue virus infection and 22,000 deaths. Dengue can range from causing a mild fever to leading to the life-threatening dengue hemorrhagic fever (DHF) or dengue shock syndrome (DSS). These severe forms are characterized by a drop in platelets and white blood cells, as well as increased vascular permeability. In Dhaka city, dengue fever is a significant contributor to severe illness and hospitalizations. The temporal and geographical spread of this vector-borne disease is influenced by weather conditions [12]. Aedes mosquitoes are sensitive to changes in temperature. It serves as the principal host for dengue and yellow fever virus amplification and transmission [13]. Hence, climate stands out as a crucial factor within the epidemiological framework [14].

Weather conditions such as temperature, humidity, and rainfall often exhibit persistence over time [15], influencing the transmission dynamics of dengue [16]. Similarly, dengue incidence may depend on past values due to the incubation period of the virus and the persistence of environmental conditions favorable for mosquito breeding [17]. Besides that the relationship between dengue cases and weather variables such as temperature, rainfall, and humidity often exhibits nonlinear patterns. Modeling dengue data requires addressing both nonlinearity in the relationship between dengue cases and weather variables, as well as autocorrelation due to temporal dependencies in the data.

Many studies have been carried out to determine the association between dengue cases and meteorological factors in Bangladesh. For example, a study by Hossain et al. used Poisson, zero-inflated regression, and negative binomial models to explore the relationship between daily dengue counts and climate factors [12]. Islam et al. fitted the Poisson regression model of dengue cases with each of the climatic factors reveal the relationships between dengue cases and climatic factors [18]. But none of the above models can cover nonlinearity as well as autocorrelation. Islam et al. used a GAM to analyze interaction of climatic factors with dengue cases

[19]. Although GAM can incorporate nonlinearity but it has failed to incorporate autocorrelation structure in the data [9].

To the best of our knowledge no study has considered both the nonlinearity as well as autocorrelation structure. Our main goal of this thesis is to use GAMAR model proposed by Lei et al. to investigate the effects of weather variables on dengue data.

1.1 Objectives of the study

Motivated by the issues mentioned in the preceding section, this study has the following objectives:

- To conduct a simulation study to explore how different sample sizes, lag values and different functional forms influence the performance of GAM and GAMAR.
- To apply GAMAR to analyze the relationship between dengue incidence and weather variables, focusing on capturing temporal dependencies and nonlinear relationships in the data.
- To investigate the efficiency of GAMAR over GAM for modeling dengue infected cases and weather variables.
- To recommend various strategies to the policy makers based on the result of our thesis.

1.2 Layout of the Study

The contents of this study are organized as follows:

Chapter 1: Introduces the idea of time series count data and initiates a comprehensive discussion of the GAMAR model, including its motivations for use. The objectives of the study are outlined.

Chapter 2: Discusses Generalized Linear Models (GLM), Generalized Additive Models (GAM), and Generalized Additive Models with Autoregressive Terms (GAMAR), highlighting how the latter integrates autoregressive components into GAMs to effectively capture temporal dependencies in data. Additionally, the estimation techniques for GAMAR are presented.

Chapter 3: Presents the findings and discussions from our simulation study comparing how GAM and GAMAR perform across various sample sizes and autoregressive orders.

Chapter 4: Provides a detailed description of the data and essential variables used in this study, and explore the application of GAMAR in analyzing the correlation between dengue incidence and weather variables.

Chapter 5: Concludes the study and presents major findings from the study.

Chapter 2

Methodology

When the response variable represents counts, the model often takes the form of a Poisson distribution. Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) are the primary models used for modeling count data. GLMs are suitable when the relationship between predictors and the response variable is expected to be approximately linear. In contrast, GAMs are preferred when there are evident nonlinear relationships between the predictors and the response variable. However, for modeling time series count data, both of these models often fall short as they cannot adequately account for the autocorrelation structure of the observations. Lei et al. addressed this limitation by proposing the Generalized Additive Model with Autoregressive Terms (GAMAR), which can incorporate autocorrelation.

This chapter begins with a brief overview of GLMs and GAMs. It then provides a detailed discussion of GAMAR. Finally, the chapter explores the computational techniques supporting our study.

2.1 Generalized Linear Model

Generalized linear models (GLMs) extend traditional linear models to allow for a broader range of response variable distributions and relationships between the response and predictor variables. They offer the flexibility to model response variables with error distributions beyond the typical assumption of normality. This

versatility allows for a more realistic representation of data, especially when the response variable's distribution deviates from the normal. Nelder and Wedderburn developed the Generalized Linear Model (GLM) [20]. Suppose that $Y_1, Y_2, \ldots, Y_n$ are the n independent counts of response variable, $\mathbf{x}_i^T = (x_{i1}, x_{i2}, \ldots, x_{ik})$ be a vector of k covariates and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ be the vector of regression parameters. Under GLM setup, the $\boldsymbol{\beta}$'s are the main parameter of interest and need to be estimated. GLM mainly consists of the following three components:

1. random component
2. linear predictor
3. link function

The conditional distribution of the response variable, given the values of the explanatory variables in the model, is specified by a random component. Nelder and Wedderburn's formulation states that the probability density function of the response variable Y_i belongs to the exponential family of distributions.

Systematic component represents the linear predictor η_i, which is a linear function of covariate $\mathbf{x_i}$ and regression parameter $\boldsymbol{\beta}$. It is defined as,

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik},$$

where η_i ranges from $-\infty$ to ∞. Link function defines the relationship between the mean of the response variable and the linear predictors. Let $h\,(.)$ be the link function, then the generalised linear model can be expressed as

$$h\,(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}.$$

The Poisson regression model is a type of generalized linear model. In this model, it is assumed that the dependent variables follow a Poisson distribution with a mean μ_i. This mean, μ_i, is associated with the explanatory variables and is typically modeled using a log link function. This relationship is mathematically expressed as:

$$\log(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}.$$

In the classical approach, the model parameters are primarily estimated using the maximum likelihood estimation (MLE) method. This method involves maximizing the likelihood function. The likelihood function can be expressed as follows:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{e^{-\mu_i} \mu_i^{y_i}}{y_i!}.$$

For mathematical convenience, we take the logarithm of the above likelihood known as the log likelihood function, which can be expressed as,

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \Big(y_i \log(\mu_i) - \mu_i - \log(y_i!) \Big).$$

Next, we need to find the score function. In statistics, the score function is the partial derivative of the log-likelihood function with respect to the vector of parameters, denoted by $U(\boldsymbol{\beta})$ and can be expressed as

$$U_{(k\times 1)}(\boldsymbol{\beta}) = \begin{pmatrix} \frac{\delta l(\boldsymbol{\beta})}{\delta \beta_1} \\ \frac{\delta l(\boldsymbol{\beta})}{\delta \beta_2} \\ \vdots \\ \frac{\delta l(\boldsymbol{\beta})}{\delta \beta_k} \end{pmatrix}.$$

The observed information matrix, $I^*(\boldsymbol{\beta})$, is negative of the first derivative of the score function or negative of the second derivative of the log-likelihood function. Thus,

$$I^*_{(k\times k)}(\boldsymbol{\beta}) = - \begin{pmatrix} \frac{\delta U_1(\boldsymbol{\beta})}{\delta \beta_1} & \frac{\delta U_1(\boldsymbol{\beta})}{\delta \beta_2} \cdots & \frac{\delta U_1(\boldsymbol{\beta})}{\delta \beta_k} \\ \frac{\delta U_2(\boldsymbol{\beta})}{\delta \beta_1} & \frac{\delta U_2(\boldsymbol{\beta})}{\delta \beta_2} \cdots & \frac{\delta U_2(\boldsymbol{\beta})}{\delta \beta_k} \\ \vdots & & \\ \frac{\delta U_k(\boldsymbol{\beta})}{\delta \beta_1} & \frac{\delta U_k(\boldsymbol{\beta})}{\delta \beta_2} \cdots & \frac{\delta U_k(\boldsymbol{\beta})}{\delta \beta_k} \end{pmatrix}.$$

Now the maximum likelihood estimating equations are

$$U(\boldsymbol{\beta}) = \mathbf{0}$$

$$\sum_{i=1}^{n} \Big(y_i - \exp\left(\mathbf{x}_i^T \boldsymbol{\beta}\right) \Big) \mathbf{x}_i^T = \mathbf{0},$$

where $\mathbf{0} = (0, 0, \ldots, 0)$. The above equations can be solved by Newton-Raphson's iterative procedure and the m^{th} iteration estimates are:

$$\hat{\boldsymbol{\beta}}^{(m)} = \hat{\boldsymbol{\beta}}^{(m-1)} + \Big[I^*(\boldsymbol{\beta}) \Big]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{(m-1)}} U(\boldsymbol{\beta}) \Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}.$$

Stop the iteration when $\left| \hat{\boldsymbol{\beta}}^{(m+1)} - \hat{\boldsymbol{\beta}}^{(m)} \right| < \epsilon$, where ϵ is a very small positive quantity.

2.2 Generalized Additive Model

Generalized Additive Model (GAM) is a flexible extension of Generalized Linear Model (GLM) that allows for non-linear relationships between the predictors and the response variable. Introduced by Hastie and Tibshirani in 1990, GAMs combine the interpretability of linear models with the flexibility of non-parametric models [21]. However, semiparametric GAMs that combine nonparametric and parametric modeling are also available. In a regression model, the term nonparametric indicates an interest in the shape of the relationship (such as its wiggliness or nonlinear pattern) between the response variable(s) and the covariate(s). This relationship is often explained through visualization. This differs from parametric regression, where specific parameters are required to describe any linear or nonlinear pattern [22]. GAMs offer the flexibility to accommodate both types of modeling. We have considered the non parametric part of GAM for our thesis. GAMs are effective tools for capturing complex relationships without requiring explicit specification of nonlinear functions for predictors. GAMs operate under two main assumptions: additivity, where predictors have an additive effect on the response, and smoothness, implying a continuous relationship. The decision to use GAMs can be based on prior knowledge suggesting nonlinear patterns in the data or on visual inspection revealing irregularities. Overall, GAMs offer flexibility

in modeling nonlinear relationships and are particularly useful when traditional linear models may not adequately capture the complexity of the data. A logical extension of the generalized linear model

$$\log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}.$$

Each linear component $\beta_k x_{ik}$ is replaced with a smooth nonlinear function $f_j\left(x_{ij}\right)$. Under this setup, the above equation reduces to the following form:

$$\log(\mu_i) = \beta_0 + \sum_{j=1}^{k} f_j(x_{ij})$$

$$\log(\mu_i) = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_k(x_{ik}).$$

In this formulation, μ_i represents the expected value of the response variable Y for the i-th observation. The model involves k smooth functions $f_j\left(x_{ij}\right)$ one for each feature x_{ij}. These smooth functions capture the noninear relationships between each feature and the response. This additive structure allows us to calculate a separate smooth function f_j for each feature x_{ij}, and then we add together all of their contributions to model the log of the expected value of the response. This is why it is called an additive model.

2.2.1 Basis Function

The concept of basis functions is to have a set of functions or transformations readily available that can be applied to a variable X. These functions, which can be denoted as $b_1(X), b_2(X), \ldots, b_k(X)$, form a basis. Instead of fitting a linear model directly in terms of X, we fit the model using these basis functions. The model takes the following form

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i,$$

where ϵ_i represents the error term, and the basis functions $b_1(\cdot), b_2(\cdot), \ldots, b_K(\cdot)$ are predetermined and known functions of the variable X. Essentially, we are expressing the relationship between X and Y as a linear combination of these basis functions, with coefficients $\beta_0, \beta_1, \ldots, \beta_K$. For example, in polynomial regression the basis functions $b_j(x_i)$ are simply powers of x_i, such that $b_j(x_i) = x_i^j$, where j denotes the degree of the polynomial.

The basis function approach provides a flexible framework for representing smooth functions within Generalized Additive Models (GAMs). Several smoothers are available in the literature. Some of the most well known smoothers are the natural cubic spline, penalized cubic regression spline etc. [23].

2.2.2 Knots and Location of Knots

When fitting a spline, the placement of knots is crucial for flexibility and accuracy. A regression spline is more adaptable in areas with a higher concentration of knots, as the polynomial coefficients can change more rapidly in those regions. Therefore, a practical approach is to position more knots in areas where the function is expected to change quickly and fewer knots in areas where the function is more stable. This can be achieved by examining the scatterplot of the dataset for the continuous response variable or by utilizing any prior knowledge. However, they should be in a modest amount, i.e neither too much nor too little.

In this study, we utilize the natural cubic spline as a smoother. Natural splines use a set of smooth piecewise polynomial functions as basis functions to model the relationship between the predictor variable X and the response variable Y. The general form of the cubic spline regression model with K knots (excluding the two endpoints) [24–28] is given by

$$f(x) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3$$

$$= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \theta_1 (x - \xi_1)_+^3 + \theta_2 (x - \xi_2)_+^3 + \cdots + \theta_K (x - \xi_K)_+^3, \quad (1)$$

where

$$(x - \xi_k)_+^3 = \begin{cases} (x - \xi_k)^3 & \text{if } x > \xi_k, \\ 0 & \text{if } x \leq \xi_k. \end{cases}$$

For the natural spline, it is necessary to ensure that the line before the first knot and the line after the last knot are linear. These constraints lead to the following conditions:

- For the first restriction, when x is less than the first knot, we need

$$\beta_2 = 0 \quad \text{and} \quad \beta_3 = 0. \tag{2}$$

- For the second restriction, when x is greater than the last knot, we also need

$$\sum_{k=1}^{K} \theta_k = 0 \quad \text{and} \quad \sum_{k=1}^{K} \xi_k \theta_k = 0. \tag{3}$$

From equation (1), when $x < \xi_1$, $f(x)$ must be linear. This means that $f(x)$ can only be expressed as a combination of x to the power of 0 or 1, i.e., it should not contain terms like x^2, x^3, etc.

When $x < \xi_1$, we have

$$f(x) = \sum_{j=0}^{3} \beta_j x^j = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3. \tag{4}$$

Therefore, by the linearity, both β_2 and β_3 have to be equal to zero. For the second restriction, when $x > \xi_K$ (i.e. when x is greater than the last knot), we can omit the + sign for the truncated function. This is because when $x > \xi_k$, $(x - \xi_k)_+^d$ is equivalent to $(x - \xi_k)^d$. Consequently $f(x)$ can be expressed as

$$f(x) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)^3. \tag{5}$$

To demonstrate the final restriction, we can expand $f(x)$ and eliminate x^2 and x^3 terms. This process will confirm that $\sum_{k=1}^{K} \theta_k = 0$ and $\sum_{k=1}^{K} \xi_k \theta_k = 0$.

The second derivative of $f(x)$ is:

$$f''(x) = 6 \sum_{k=1}^{K} \theta_k (x - \xi_k) = 0 \Rightarrow \left(\sum_{k=1}^{K} \theta_k \right) x - \sum_{k=1}^{K} \theta_k \xi_k = 0, \quad \forall x > \xi_K.$$

Since $\forall x$ that $x > \xi_K$, we have $f''(x) = 0$, therefore, both

$$\sum_{k=1}^{K} \theta_k = 0 \quad \text{and} \quad \sum_{k=1}^{K} \theta_k \xi_k = 0.$$

i.e.,

$$\sum_{k=1}^{K} \theta_k = 0 \quad \text{and} \quad \sum_{k=1}^{K} \xi_k \theta_k = 0.$$

As we introduce constraints (2) and (3) to equation (1), we have to develop a new representation of the linear regression model. Now when we put the restriction (3) on (1)

$$\sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3 = \sum_{k=1}^{K-1} \theta_k (x - \xi_k)_+^3 + \theta_K (x - \xi_K)_+^3. \tag{6}$$

Given the restriction $\sum_{k=1}^{K} \theta_k = 0$, we can infer that

$$\sum_{k=1}^{K-1} \theta_k + \theta_K = 0 \Rightarrow \theta_K = -\sum_{k=1}^{K-1} \theta_k,$$

we can replace θ_K in equation (6) with $-\sum_{k=1}^{K-1} \theta_k$ and we get

$$\begin{aligned} \sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3 &= \sum_{k=1}^{K-1} \theta_k (x - \xi_k)_+^3 - \sum_{k=1}^{K-1} \theta_k (x - \xi_K)_+^3 \\ &= \sum_{k=1}^{K-1} \theta_k \left[(x - \xi_k)_+^3 - (x - \xi_K)_+^3 \right]. \end{aligned} \tag{7}$$

Note that we have not yet utilized the condition $\sum_{k=1}^{K} \xi_k \theta_k = 0$ for the restriction (3). Now, we will incorporate this condition. Given

$$\sum_{k=1}^{K} \theta_k = 0 \quad \text{and} \quad \sum_{k=1}^{K} \xi_k \theta_k = 0 \Rightarrow \Big(\sum_{k=1}^{K} \theta_k \Big) \xi_K - \sum_{k=1}^{K} \xi_k \theta_k = 0. \tag{8}$$

From equation (7), we have

$$\begin{aligned} &\Big(\sum_{k=1}^{K} \theta_k \Big) \xi_K - \sum_{k=1}^{K} \xi_k \theta_k = 0 \\ &\Rightarrow \sum_{k=1}^{K} \theta_k (\xi_K - \xi_k) = 0 \\ &\Rightarrow \sum_{k=1}^{K-1} \theta_k (\xi_K - \xi_k) + \theta_K (\xi_K - \xi_K) = 0 \end{aligned}$$

$$\Rightarrow \sum_{k=1}^{K-1} \theta_k(\xi_K - \xi_k) = 0$$

$$\Rightarrow \sum_{k=1}^{K-2} \theta_k(\xi_K - \xi_k) + \theta_{K-1}(\xi_K - \xi_{K-1}) = 0$$

$$\Rightarrow \theta_{K-1} = -\frac{\sum_{k=1}^{K-2} \theta_k(\xi_K - \xi_k)}{\xi_K - \xi_{K-1}}. \tag{9}$$

Combining equations (7) and (9)

$$\sum_{k=1}^{K} \theta_k(x - \xi_k)_+^3 = \sum_{k=1}^{K-1} \theta_k \Big[(x - \xi_k)_+^3 - (x - \xi_K)_+^3\Big]$$

$$= \sum_{k=1}^{K-2} \theta_k \Big[(x - \xi_k)_+^3 - (x - \xi_K)_+^3\Big] + \theta_{K-1}\Big[(x - \xi_{K-1})_+^3 - (x - \xi_K)_+^3\Big]$$

$$= \sum_{k=1}^{K-2} \theta_k \Big[(x-\xi_k)_+^3 - (x-\xi_K)_+^3\Big] - \sum_{k=1}^{K-2} \theta_k \frac{\xi_K - \xi_k}{\xi_K - \xi_{K-1}} \Big[(x-\xi_{K-1})_+^3 - (x-\xi_K)_+^3\Big]. \tag{10}$$

Let us define a function

$$d_k(x) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k}.$$

$$d_{K-1}(x) = \frac{(x - \xi_{K-1}) +^3 - (x - \xi_K) +^3}{\xi_K - \xi_{K-1}}.$$

Replace truncated functions in (10) by $d_k(x)$ and $d_{K-1}(x)$ and we get

$$\begin{aligned}
\sum_{k=1}^{K} \theta_k(x - \xi_k)_+^3 &= \sum_{k=1}^{K-1} \theta_k \Big[(x - \xi_k)_+^3 - (x - \xi_K)_+^3\Big] \\
&= \sum_{k=1}^{K-2} \theta_k \Big[(x - \xi_k)_+^3 - (x - \xi_K)_+^3\Big] - \sum_{k=1}^{K-2} \theta_k a_k \Big[(x - \xi_{K-1})_+^3 - (x - \xi_K)_+^3\Big] \\
&= \sum_{k=1}^{K-2} \theta_k(\xi_K - \xi_k) d_k(x) - \sum_{k=1}^{K-2} \theta_k a_k (\xi_K - \xi_{K-1}) d_{K-1}(x) \\
&= \sum_{k=1}^{K-2} \theta_k(\xi_K - \xi_k) \Big[d_k(x) - d_{K-1}(x)\Big],
\end{aligned}$$

where $a_k = \dfrac{\xi_K - \xi_k}{\xi_K - \xi_{K-1}}$. Now putting $\sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3$ and (2) back to (1) we get the final natural cubic spline regression model.

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K-2} \theta_k \left(\xi_K - \xi_k\right) \Big[d_k(x) - d_{K-1}(x) \Big].$$

For $f(x)$, a standard linear regression model can be applied according to the basis function utilized in natural spline regression. Conventional techniques like maximum likelihood estimation and ordinary least squares can be used to estimate the coefficients. However, when applying this in the context of a Generalized Additive Model (GAM), the process involves fitting the spline terms within the GAM framework. This approach leverages the flexibility of GAMs to model nonlinear relationships while maintaining the interpretability and robustness of spline-based regression.

The degrees of freedom for a natural spline equal the number of subintervals separated by these knots. When the degrees of freedom (df) are specified, the standard practice is to place df-1 number of interior knots at evenly spaced intervals within the data.

2.2.3 Estimation Procedure

There are essentially two estimation procedure for GAM:

- **Local Scoring Procedure with Backfitting Algoritm:** This procedure is a generalization of Fisher Scoring Procedure followed by the Backfitting Algorithm. It was proposed by Hastie and Tibshirani (1986). It has the advantage of ensuring convergence of a model.

- **Penalized Iteratively Re-weighted Least Squares (PIRLS) Iteration**: In the context of Generalized Additive Models (GAMs), the PIRLS algorithm is specifically tailored to fit GAMs with smooth additive components, where each component is modeled using a smooth function, such as a spline or a smoother. PIRLS for GAMs iteratively updates the smooth additive components of the model while incorporating penalties to control

their smoothness. It is a powerful algorithm for fitting GAMs that can handle complex, non-linear relationships between predictors and the response variable while avoiding overfitting through penalization.

In this study, for fitting GAM, we use the R package *mgcv* [29]. The mgcv package is highly optimized and efficient, employing various numerical and computational techniques to ensure fast and accurate model fitting.

2.2.4 Illustrative Example

To illustrate how GAM works, we have considered a very small dataset, consisting of six data points. By applying a GAM to our small dataset, we have demonstrated how it works and its ability to model complex, nonlinear relationships. This approach can be extended to larger and more complex datasets, providing a robust tool for uncovering hidden patterns and trends in data. Suppose we have the values of independent variables $x = (0.5, 1.5, 2.0, 3.0, 4.5, 5.0)$ and response variable $y = (1, 3, 6, 14, 20, 45, 70)$ respectively. We first visualize the data by creating a line graph. This provides an initial understanding of the relationship between the independent and response variable.

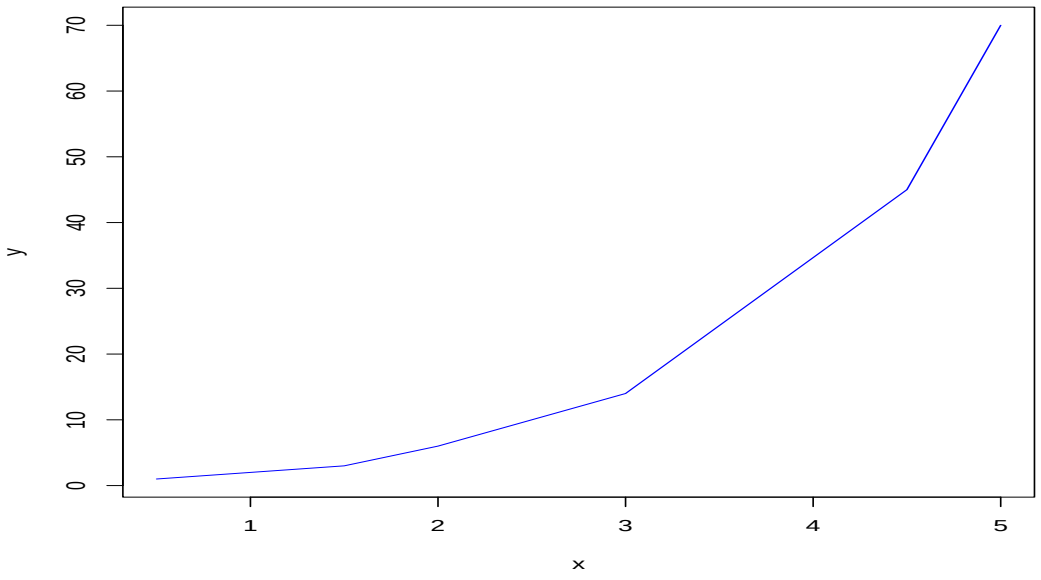


Figure 2.1: Nonlinear relationship between x and y

Based on the graphical representation, it seems appropriate to consider incorporating three knots for this dataset. We choose three knot points $\xi_1 = 2$, $\xi_2 = 3$, $\xi_3 = 4$. So, we have $3 - 2 = 1$ basis functions for each data point. Now, we will proceed to compute these basis functions.

Calculation for $x = 0.5$

$$d_{11}(x) = \frac{(0.5-2)_+^3 - (0.5-4)_+^3}{4-2} = \frac{0 - 1.5}{2} = -0.75$$

$$s_{11}(x) = (4-2) \cdot (-0.75) = -1.5$$

Calculation for $x = 1.5$

$$d_{12}(x) = \frac{(1.5-2)_+^3 - (1.5-4)_+^3}{4-2} = \frac{0 - 2.625}{2} = -1.3125$$

$$s_{12}(x) = (4-2) \cdot (-1.3125) = -2.625$$

Calculation for $x = 2.0$

$$d_{13}(x) = \frac{(2.0-2)_+^3 - (2.0-4)_+^3}{4-2} = \frac{0 - 2}{2} = -1$$

$$s_{13}(x) = (4-2) \cdot (-1) = -2$$

Calculation for $x = 3.0$

$$d_{14}(x) = \frac{(3.0-2)_+^3 - (3.0-4)_+^3}{4-2} = \frac{1 - 0}{2} = 0.5$$

$$s_{14}(x) = (4-2) \cdot (0.5) = 1$$

Calculation for $x = 4.5$

$$d_{15}(x) = \frac{(4.5-2)_+^3 - (4.5-4)_+^3}{4-2} = \frac{8.125 - 0}{2} = 4.0625$$

$$s_{15}(x) = \frac{8.125}{(4-2)^2} = \frac{8.125}{4} = 2.03125$$

Calculation for $x = 5.0$

$$d_{16}(x) = \frac{(5.0-2)_+^3 - (5.0-4)_+^3}{4-2} = \frac{27-1}{2} = 13$$

$$s_{16}(x) = (4-2) \cdot 13 = 26$$

The `rms` package in `R` has scaled the values of these basis functions, which are rather huge. Normally, we can make these values lower by dividing them by $(\xi_3 - \xi_1)^2$. Scaling a continuous predictor changes the predictor variable's coefficient but has no effect on model fit or prediction.

$$s_{11}(x) = \frac{-1.5}{(4-2)^2} = \frac{-1.5}{4} = -0.375$$

$$s_{12}(x) = \frac{-2.625}{(4-2)^2} = \frac{-2.625}{4} = -0.65625$$

$$s_{13}(x) = \frac{-2}{(4-2)^2} = \frac{-2}{4} = -0.5$$

$$s_{14}(x) = \frac{1}{(4-2)^2} = \frac{1}{4} = 0.25$$

$$s_{15}(x) = (4-2) \cdot (4.0625) = 8.125$$

$$s_{16}(x) = \frac{26}{(4-2)^2} = \frac{26}{4} = 6.5$$

Next, we will utilize the *mgcv* package to model the data. In this model, we will employ x and $s_{11} \ldots s_{16}$ as explanatory variables.

2.3 Generalized Autoregressive Moving Average Models

A new class of models called Generalized Autoregressive Moving Average (GARMA) models have been introduced by Benjamin, Rigby and Stasinopoulos (2003). These models extend the traditional univariate Gaussian ARMA time series model to accommodate non-Gaussian time series data with a flexible observation-driven approach. In GARMA models, the dependent variable is assumed to follow a conditional exponential family distribution, taking into account the past history of the process [10].

In GARMA, the conditional distribution of each observation y_t, for $t = 1, \ldots n$, given the previous information set $H_t = \{X_1, \ldots, X_t, y_1, \ldots, y_{t-1}\}$ containing past observation y_t and covariate vectors $X_t = (X_{tl}, \ldots, X_{tm})$, is assumed to follow the same exponential family distribution. As with the standard GLM, the conditional mean μ_t is related to the variables by a twice-differentiable one-to-one monotonic function g, which is called the link function. However, unlike the standard GLM, the formula here allows autoregressive moving average terms to be included additively:

$$g(\mu_t) = \sum_{i=1}^{m} X_{ti}\beta_i + \sum_{j=1}^{p} c_j \Big[g(y_{t-j}) - \sum_{i=1}^{m} X_{t-j,i}\beta_i \Big] + \sum_{j=1}^{q} d_j \Big[g(y_{t-j}) - g(\mu_{t-j}) \Big],$$

where $\sum_{j=1}^{p} c_j \Big[g(y_{t-j}) - \sum_{i=1}^{m} X_{t-j,i}\beta_i \Big]$ are autoregressive terms and $\sum_{j=1}^{q} d_j \Big[g(y_{t-j}) - g(\mu_{t-j}) \Big]$ are moving average terms. The Poisson GARMA submodel is:

$$\ln(\mu_t) = \sum_{i=1}^{m} X_{ti}\beta_i + \sum_{j=1}^{p} c_j \Big[\ln\left(y^*_{t-j}\right) - \sum_{i=1}^{m} X_{t-j,i}\beta_i \Big] + \sum_{j=1}^{q} d_j \Big[\ln\left(y^*_{t-j}/\mu_{t-j}\right) \Big],$$

where $y_t = \max(y_t, \tau)$, τ is a positive threshold parameter. Any 0 or negative values of y are replaced by τ, because ln() is not defined for 0 or negative values.

2.4 Generalized Additive Model with Autoregressive Terms

Generalized Additive Models with Autoregressive Terms (GAMAR), introduced by Lei et al., extend Generalized Additive Models (GAMs) by including autoregressive components. This enhancement allows GAMAR to effectively account for temporal dependencies in the data. The concept of GAMAR integrates the flexibility of GAMs with the ability to model time series data's inherent autocorrelation. GAMAR differs from GARMA by generalizing the linear components to natural splines and omitting the moving-average terms. This modification allows for modeling nonlinear relationships, and the absence of MA terms is justified since AR terms can approximate both MA and ARMA models. Furthermore, a higher order of AR in GAMAR does not hinder the estimation of the effects of explanatory variables. Here we use x_t instead of x since the data are time series count data. The model becomes

$$g(\mu_t) = \sum_{i=1}^{m} s_i(X_{ti}) + \sum_{j=1}^{p} c_j \Big[g(y_{t-j}) - \sum_{i=1}^{m} s_i(X_{t-j,i}) \Big], \tag{11}$$

where $\sum_{i=1}^{m} s_i(X_{ti})$ are smoothers of covariates, $\sum_{j=1}^{p} c_j \Big[g(y_{t-j}) - \sum_{i=1}^{m} s_i(X_{t-j,i}) \Big]$ are autoregressive terms. Compared to GAM, (11) allows autoregressive terms to be included additively in the link predictor.

For count data y, we use the Poisson submodel:

$$\ln(\mu_t) = \sum_{i=1}^{m} s_i(X_{ti}) + \sum_{j=1}^{p} c_j \Big[\ln\left(y^*_{t-j}\right) - \sum_{i=1}^{m} s_i(X_{t-j,i}) \Big], \tag{12}$$

where $y_t = \max(y_t, \tau)$, τ is a positive threshold parameter. By using ns as smoother in (12), the poisson GAMAR becomes:

$$\ln(\mu_t) = \sum_{i=1}^{m} ns(X_{it}, df_i) + \sum_{j=1}^{p} c_j \Big[\ln\left(y^*_{t-j}\right) - \sum_{i=1}^{m} ns(X_{t-j,i}, df_i) \Big].$$

2.4.1 Estimation Procedure

Maximum Partial Likelihood Estimator (MPLE)

For a jointly distributed time series $\{X_t, y_t\}$, $t = 1, \dots n$, the parameters of GAMAR can be estimated using the maximum partial likelihood approach. The partial likelihood based on y_t for $\{X_t, y_t\}$, $t = 1, \dots, n$ is expressed as the product of a sequence of conditional likelihood $f\left(y_t \mid X^{(t)}, y^{(t-1)}; \theta\right)$, $t = 1, \dots, n$, where $X^{(t)} = X_1, \dots X_t, y^{(t-1)} = y_1, \dots y_{t-1}$. Specifically, the partial likelihood is given by

$$PL = \prod_{t=1}^{n} f\left(y_t \mid X^{(t)}, y^{(t-1)}; \theta\right),$$

This is called a partial likelihood rather than a full likelihood because the X_t are stochastic. For a Poisson GAMAR, the partial likelihood is

$$PL = \prod_{t=1}^{n} \frac{\mu_t^{y_t}}{y_t!} e^{-\mu_t},$$

and μ_t can be expressed as

$$\ln(\mu_t) = \sum_{i=1}^{m} ns\left(X_{ti}, df_i\right) + \sum_{j=1}^{p} c_j \Big[\ln\left(y^*_{t-j}\right) - \sum_{i=1}^{m} ns\left(X_{t-j,i}, df_i\right) \Big].$$

Since a natural cubic spline is a linear combination of its B-spline basis, it behaves like linear terms in computation. Therefore,

$$\ln(\mu_t) = \eta_t = \sum_{i=1}^{m} \beta_i X_{ti} + \sum_{j=1}^{p} c_j \Big[\ln\left(y^*_{t-j}\right) - \sum_{i=1}^{m} \beta_i X_{t-j,i} \Big],$$

where $\theta = (\beta_1 \cdots, \beta_m, c_1, \cdots, c_p)^T$ is the model parameter vector.

Modified Newton's method

To maximize the partial likelihood, the parameters are estimated using a modified Newton's method. The detailed calculations are provided here for transparency and better understanding. For Newton's method, the iteration goes

$$\theta_{m+1} = \theta_m - \left(\frac{\partial^2 \ln(PL)}{\partial \theta_i \partial \theta_j} \right)^{-1} \frac{\partial \ln(PL)}{\partial \theta} \Bigg|_{\theta = \theta_m}, \text{until it convergence.}$$

For a modified Newton's method, the iteration goes:

$$\theta_{m+1} = \theta_m + \Gamma_*^{-1}(\theta_m) \left. \frac{\partial \ln(PL)}{\partial \theta} \right|_{\theta=\theta_m}, \text{until it convergence.} \tag{13}$$

where $\Gamma(\theta) = -\dfrac{\partial^2 \ln(PL)}{\partial\theta\partial\theta^T}$, which is the information matrix, and $\Gamma_*^{-1}(\theta_m)$ is a modified version of $\Gamma^{-1}(\theta_m)$. From equation (13) we have

$$\begin{aligned}
\frac{\partial \ln(PL)}{\partial \theta_i} &= \frac{\partial \sum_{t=1}^{n} \Big(y_t \ln(\mu_t) - \mu_t - \ln(y_t!) \Big)}{\partial \theta_i} \\
&= \frac{\partial \sum_{t=1}^{n} \Big(y_t \eta_t - e^{\eta_t} - \ln(y_t!) \Big)}{\partial \theta_i} \\
&= \sum_{t=1}^{n} \left(y_t \frac{\partial \eta_t}{\partial \theta_i} - \mu_t \frac{\partial \eta_t}{\partial \theta_i} \right) \\
&= \sum_{t=1}^{n} (y_t - \mu_t) \frac{\partial \eta_t}{\partial \theta_i}.
\end{aligned}$$

Now $\Gamma(\theta)$ can be calculated as

$$\Gamma(\theta) = -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \dfrac{\partial \eta_t}{\partial \theta_i}}{\partial \theta_j} = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}.$$

Since

$$\ln(\mu_t) = \eta_t = \sum_{i=1}^{m} \beta_i X_{ti} + \sum_{j=1}^{p} c_j \Big(\ln\big(y^*_{t-j}\big) - \sum_{i=1}^{m} \beta_i X_{t-j,i} \Big),$$

then

$$\frac{\partial \eta_t}{\partial \beta_i} = X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i}$$

$$\frac{\partial \eta_t}{\partial c_i} = \ln\big(y^*_{t-i}\big) - \sum_{k=1}^{m} \beta_k X_{t-j,k},$$

$$\frac{\partial^2 \eta_t}{\partial \beta_i \partial \beta_j} = 0, \frac{\partial^2 \eta_t}{\partial c_i \partial c_j} = 0, \frac{\partial^2 \eta_t}{\partial \beta_i \partial c_j} = -X_{t-j,i}.$$

Now

$$A = -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \frac{\partial \eta_t}{\partial \beta_i}}{\partial \beta_j}$$

$$= \sum_{t=1}^{n} \Big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \Big) \frac{\partial \mu_t}{\partial \beta_j}$$

$$= \sum_{t=1}^{n} \Big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \Big) \mu_t \frac{\partial \eta_t}{\partial \beta_j}.$$

$$B = -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \frac{\partial \eta_t}{\partial \beta_i}}{\partial c_j}$$

$$= -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \Big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \Big)}{\partial c_j}$$

$$= \sum_{t=1}^{n} \Big[\big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \big) \mu_t \frac{\partial \eta_t}{\partial c_j} + \sum_{t=1}^{n} (y_t - \mu_t) X_{t-j,i} \Big].$$

$$C = -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \frac{\partial \eta_t}{\partial c_i}}{\partial c_j}$$

$$= -\frac{\partial \sum_{t=1}^{n} (y_t - \mu_t) \Big(\ln \big(y^*_{t-i} \big) - \sum_{k=1}^{m} \beta_k X_{t-i,k} \Big)}{\partial c_j}$$

$$= \sum_{t=1}^{n} \Big(\ln \big(y^*_{t-i} \big) - \sum_{k=1}^{m} \beta_k X_{t-i,k} \Big) \mu_t \frac{\partial \eta_t}{\partial c_j}.$$

So

$$A = \Big(\sum_{t=1}^{n} \mu_t \big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \big) \big(X_{tj} - \sum_{r=1}^{p} c_r X_{t-r,j} \big) \Big)_{mm}.$$

$$B = \Big(\sum_{t=1}^{n} \Big(\mu_t \big(X_{ti} - \sum_{r=1}^{p} c_r X_{t-r,i} \big) \big(\ln \big(y^*_{t-j} \big) - \sum_{k=1}^{m} \beta_k X_{t-j,k} \big) + (y_t - \mu_t) X_{t-j,i} \Big) \Big)_{mp}.$$

$$C = \Big(\sum_{t=1}^{n} \big(\ln \big(y^*_{t-i} \big) - \sum_{k=1}^{m} \beta_k X_{t-i,k} \big) \big(\ln \big(y^*_{t-j} \big) - \sum_{k=1}^{m} \beta_k X_{t-j,k} \big) \Big)_{pp}.$$

And the modified inverse matrix ${\Gamma_*}^{-1}(\theta_m)$ is defined as follows:

- If $\Gamma(\theta)$ is reversible, then $\Gamma^{*-1}(\theta) = \Gamma^{-1}(\theta)$.
- If $\Gamma(\theta)$ is irreversible and its eigenvalues are $\lambda_1, \lambda_2, \ldots, \lambda_{mp}$, then we can find an orthogonal matrix P, which satisfies:

$$P^T \Gamma(\theta) P = \mathrm{diag}(\lambda_1, \ldots, \lambda_{mp}).$$

Let $\lambda^*_i = \max(\lambda_i, \delta)$, where $\delta = 0.01$, $i = 1, 2, \ldots, mp$, then:

$$\Gamma^{*-1}(\theta) = P \, \mathrm{diag}(\lambda_1^{*^{-1}}, \ldots, \lambda_{mp}^{*^{-1}}) \, P^T.$$

Such a procedure ensures $\Gamma^{*-1}(\theta_m)$ to be positive definite.

2.4.2 Illustrative Example

In this section, the working procedure of GAMAR model with natural cubic splines is discussed in details for a small dataset. The data used in this analysis has been simulated using GAMAR, with a detailed discussion of data generation technique is planned for the subsequent chapter. Since we have considered a lag of 1, there will be no y value for t=1. The following table shows the data layout:

Table 2.1: Illustrative example using hypothetical data

t	y_t	X_{t1}
1	0	27.6
2	136	28.0
3	116	32.1
4	147	27.0
5	85	37.0
6	111	32.8

For simplicity, here we have used m=1 covariate and p=1 lag. Assume the smoothers s_i are linear, i.e., $s_i(X_{ti}) = \beta_i X_{ti}$.

The model is given by

$$\ln(\mu_t) = \eta_t = \beta_1 X_{t1} + c_1\Big(\ln\left(y^*_{t-1}\right) - \beta X_{t-1,1}\Big).$$

Step 1: Using a simple linear basis for the splines, the basis functions for X_{t1} might look like this (assuming a single knot at the median):
For X_{t1}:

$$s_1(X_{t1}) = \beta_{11} X_{t1} + \beta_{12} \max(0, X_{t1} - 30).$$

Step 2: Let's start with initial parameters for the splines:

$$\beta_{11} = 0.1, \quad \beta_{12} = 0.1, \quad c_1 = 0.1.$$

Step 3: For each time point, we have to calculate μ_t:
At $t = 2$

$$s_1(X_{21}) = 0.1 \cdot 28.0 + 0.1 \cdot \max(0, 28.0 - 30) = 0.1 \cdot 28.0 + 0 = 2.8$$

$$s_1(X_{11}) = 0.1 \cdot 27.6 + 0.1 \cdot \max(0, 27.6 - 30) = 0.1 \cdot 27.6 + 0 = 2.76$$

$$\ln(\mu_2) = s_1(X_{21}) + c_1\left(\ln(y_1 + 1) - s_1(X_{11})\right)$$

(Note: y_1 is 0, adding 1 to avoid taking ln(0))

$$\ln(\mu_2) = 2.8 + 0.1\left(\ln(1) - 2.76\right)$$

$$\ln(1) = 0$$

$$\ln(\mu_2) = 2.8 + 0.1(0 - 2.76) = 2.8 - 0.276 = 2.524$$

$$\mu_2 = e^{2.524} \approx 12.49$$

At $t = 3$

$$s_1(X_{32}) = 0.1 \cdot 32.1 + 0.1 \cdot \max(0, 32.1 - 30) = 3.21 + 0.21 = 3.42$$

$$s_1(X_{22}) = 0.1 \cdot 28.0 + 0.1 \cdot \max(0, 28.0 - 30) = 0.1 \cdot 28.0 + 0 = 2.8$$

$$\ln(\mu_3) = s_1(X_{32}) + c_1\left(\ln(y_2 + 1) - s_1(X_{22})\right)$$

$$\ln(\mu_3) = 3.42 + 0.1\,(\ln(136+1) - 2.8)$$

$$\ln(137) \approx 4.92$$

$$\ln(\mu_3) = 3.42 + 0.1(4.92 - 2.8) = 3.42 + 0.212 = 3.632$$

$$\mu_3 = e^{3.632} \approx 37.86$$

Similarly, we can calculate μ_4, μ_5, and μ_6. The partial likelihood is

$$PL = \prod_{t=2}^{6} \frac{\mu_t^{y_t}}{y_t!} e^{-\mu_t},$$

For simplicity, let's illustrate it for one term:

$$PL_2 = \frac{12.49^{136}}{136!} e^{-12.49} \approx 0.267$$

Step 4: For $t = 2$ the gradient and Hessian for Newton's method:

$$\frac{\partial \ln(PL)}{\partial \beta_i} = \sum_{t=2}^{6} (y_t - \mu_t) \frac{\partial \eta_t}{\partial \beta_i}$$

$$\frac{\partial \eta_t}{\partial \beta_{11}} = X_{t1} - c_1 X_{t-1,1} = 28.0 - 0.1 \cdot 27.6 = 28.0 - 2.76 = 25.24$$

$$\frac{\partial \ln(PL)}{\partial \beta_{11}} = (136 - 12.49) \cdot 25.24 \approx 3110.5$$

We have to repeat this for β_{12}, c_1, and calculate the second derivatives similarly to form the Hessian matrix.

Step 5: Update the parameters using Newton's method:

$$\theta_{m+1} = \theta_m - \left(\frac{\partial^2 \ln(PL)}{\partial \theta \partial \theta^T} \right)^{-1} \frac{\partial \ln(PL)}{\partial \theta} \bigg|_{\theta=\theta_m}.$$

Iterate steps 3-5 until convergence.

2.5 Model Evaluation Techniques

Coverage Probability

Coverage probability is a key measure of the reliability and accuracy of the intervals produced by a statistical model or method. It quantifies the confidence in

the model's ability to capture the variability in the data. For example, if a 95% confidence interval is constructed for a parameter, the coverage probability indicates the proportion of times, in repeated sampling, that this interval will contain the true value of the parameter. If the confidence level is chosen to be 95% , the coverage of a correct model should be around 95%.

ACF and PACF

The autocorrelation function (ACF) and partial autocorrelation function (PACF) are statistical tools used in time series analysis to understand the correlation structure within a series. We have showed below when to use AR, MA, and ARMA models based on the characteristics of their ACF and PACF.

AR models have following characteristics:

- ACF: exponential series decaying to 0 or alternatively positive and negative spikes decaying to 0.
- PACF: p significant lags or cuts off after p significant lags.

MA models have following characteristics:

- ACF: q significant lags or cuts off after q significant lags.
- PACF: exponential series decaying to 0 or alternatively positive and negative spikes decaying to 0.

ARMA models have following characteristics:

- ACF: exponential series decaying to 0.
- PACF: exponential series decaying to 0.

The above characteristics can also be shown for a simulated data set which is given below:

Characteristics of AR model

We have simulated data from an AR (1) model for two different seed values.

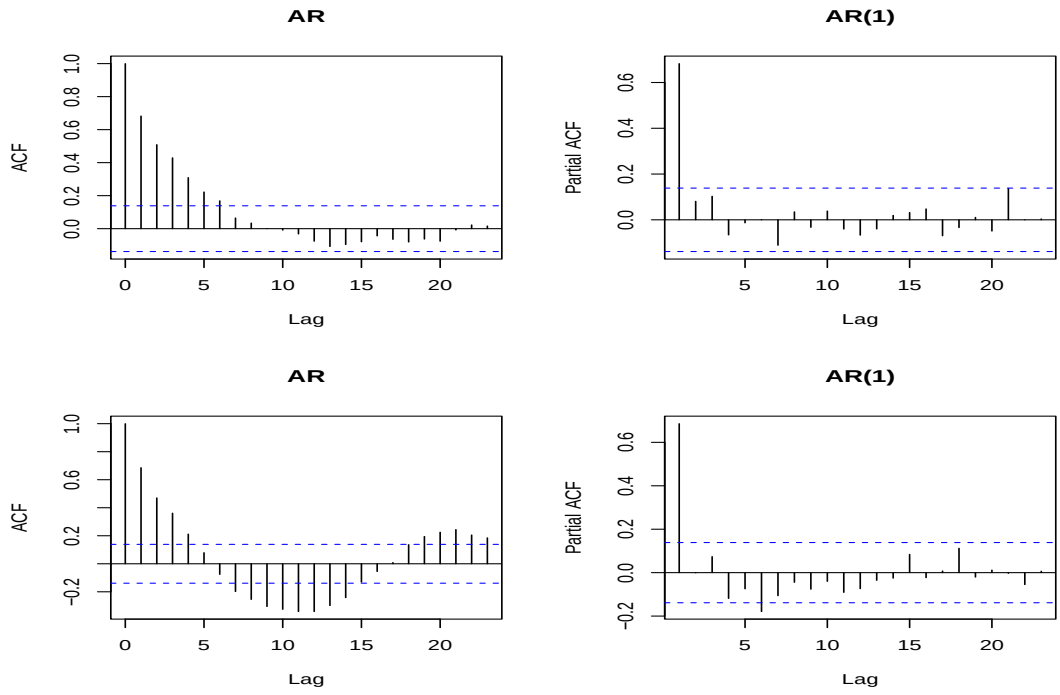


Figure 2.2: Characteristics of AR model

The results are shown in Figure 2.2. The upper two plots have the following characteristics:

- ACF function decays exponentially to zero.
- PACF function: first lag is significant, so we can consider AR (1) as a potential model.

The lower two plots have the following characteristics:

- ACF function (alternatively positive and negative spikes) decays exponentially to zero.
- PACF function: first lag is significant, so we can consider AR (1) as a potential model.

Characteristics of MA model

We have simulated data from an MA(1) model for two different seed values. The results are shown in Figure 2.3. The upper two plots have the following characteristics:

- ACF function: first lag is significant.

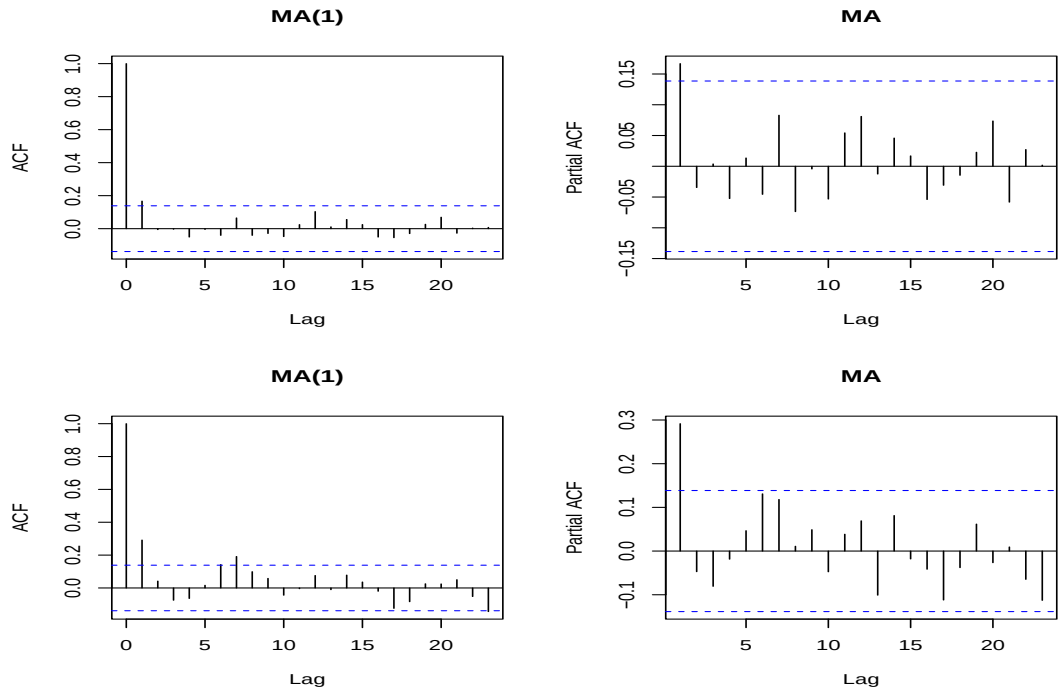


Figure 2.3: Characteristics of MA model

- PACF function: alternatively positive and negative spike decaying to 0.

So we can say that the MA (1) model can be a potential model.

The lower two plots have the following characteristics:

- ACF function: lag 1, 5 and 6 are significant.
- PACF function: alternatively positive and negative spikes decaying exponentially to zero.

So we can say that MA (1), MA (5) and MA (6) model can be a potential model. We will select an MA model when these features are shown in the ACF and PACF plots.

Characteristics of ARMA model

We have simulated data from an ARMA (1,1) model for two different seed values. The results are shown in Figure 2.4. The upper two plots have the following characteristics:

- The ACF plot typically tails off.

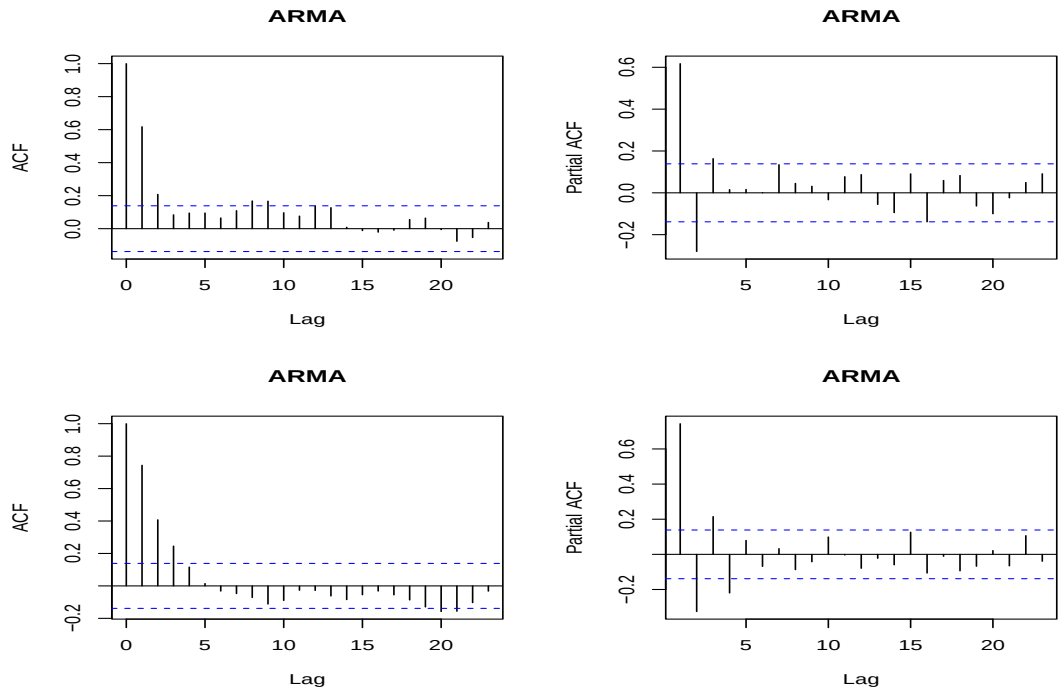


Figure 2.4: Characteristics of ARMA model

- PACF function: alternatively positive and negative spikes decaying to 0.

So potential models are AR (2), ARMA (2, 2); considering a lower order models like AR (1) and ARMA (1, 1), ARMA (1, 2), ARMA (2, 1) is a prudent way to find the best model as choosing p and q are not unique.

The lower two plots have the following characteristics:

- ACF function: alternatively positive and negative spikes decays to zero.
- PACF function: alternatively positive and negative spikes decays exponentially to zero or first q lags are significant.

Potential model are AR (2), ARMA (4, 2), ARMA (2, 4) ; considering a lower order models like AR (1) and ARMA (1, 1), ARMA (1, 2), ARMA (2, 1) etc. is a prudent way to find the best model as choosing p and q is not unique.

We will select an ARMA model when these features are shown in the ACF and PACF plots.

Chapter 3

Simulation Study

Generalized additive model with autoregressive terms for estimating time series count data has been discussed in detail in previous chapter. Lei et al. conducted simulation studies to compare the performance of GAM and GAMAR under two different setup. Responses were generated by considering a predefined set of coefficients alongside autoregressive (AR) terms under the first simulation setup while an investigation whether their suggested approach could approximate a non-linear curve was conducted under second setup. They considered one specific lag when simulated data from AR under setup 1 and one specific functional form under second setup to show the efficiency of their proposed model. However different lag values and different functional forms were not explored in their simulation study. Motivated by these lacks an extensive simulation study is conducted in this chapter by incorporating the above issues. From the result of our extensive simulation study it becomes apparent that as the lag increases, GAMAR offer more robust outcomes compared to GAM.

3.1 Simulation Setting

In our simulation setup, we have taken into account different autoregressive (AR) orders alongside varying sample sizes. Additionally, we have included a distinct set of covariates schemes and two different functional forms. The aim behind this consideration is to evaluate the consistency of partial maximum likelihood estimation and to assess the performance of both GAM and GAMAR under var-

ious scenarios. The table below shows the simulation design for which extensive simulation study is conducted in this chapter.

Table 3.1: Different sample size (n) and different lag values (AR order)

Sample size	AR Order			
(days)	1	2	3	4
730	$(1, 730)$	$(2, 730)$	$(3, 730)$	$(4, 730)$
1461	$(1, 1461)$	$(2, 1461)$	$(3, 1461)$	$(4, 1461)$
2191	$(1, 2191)$	$(2, 2191)$	$(3, 2191)$	$(4, 2191)$
2922	$(1, 2922)$	$(2, 2922)$	$(3, 2922)$	$(4, 2922)$

From the design of the simulation study shown above, it is evident that a total of 16 scenarios based on different sample sizes and AR orders have been considered in this study. The above shows different combinations of sample sizes (days) and AR order. For example, $(1, 730)$ means the sample size is 730 days and the AR order is 1. We will number the cases row-wise.

3.1.1 Scenario 1

The outcome Y_t is assumed to follow a Poisson distribution with mean (μ_t) which is linked to a smoother term and an autoregressive term. Here, natural cubic spline has been used as a smoother. The autoregressive part consists of lagged terms for both the response and the explanatory variable. We have represented the functions mathematically as follows:

Algorithm 1:

- $y_t \sim$ Poisson (μ_t)
- $\ln(\mu_t) = ns(x_t, 6) + a_t$
- $ns(x_t, 6) = \sum_{i=1}^{6} \beta_i s_{i6}(x_t)$
- $a_t = \sum_{i=1}^{p} c_i \Big(\ln\left(y^*_{t-i}\right) - ns(x_{t-i}, 6) \Big)$
  when i=1, $a_t = c_1 \Big(\ln\left(y^*_{t-1}\right) - ns(x_{t-1}, 6) \Big)$

- $y_t = \max(y, \tau), \tau = 0.5$

Here x_t represents a daily averaged temperature series obtained from the Bangladesh Meteorological Department. The x_t needs to be simulated for our simulation study but for our convenience we assume here real temperature value for x_t. The terms $s_{i6}(x_t)$, i = 1, 2, 3, 4, 5, 6 form the B-spline basis for the natural cubic spline, and τ is a threshold parameter.

Data Simulation for case 1 (1,730)

Data from the GAMAR model are needed to conduct a simulation study. Implementing the following steps yields one observation from the GAMAR model:

Step 1: Set the true values $\beta_0 = 5.02$, $\beta_1 = -0.45$, $\beta_2 = -0.46$, $\beta_3 = -0.48$, $\beta_4 = -0.43$, $\beta_5 = -0.38$, $\beta_6 = -0.25$ and $c_1 = 0.5$.
Step 2: Choose a single a_t from $a_t \sim Normal(0, 0.2)$.
Step 3: Simulate y_t by using algorithm 1.

For the other combinations mentioned in the table, we have to follow the same procedure except for the fact that the coefficient of AR terms will increase as the number of lag increases from 1 to 4, and they are $c_1 = 0.5$. $c_2 = 0.25$, $c_3 = 0.12$, and $c_4 = 0.06$, and we have to generate as many a_t as the AR order.

3.1.2 Scenario 2

We will consider two different nonlinear functions to study whether the GAMAR model can approximate nonlinear curves.

The first model
Algorithm 2:

- $y_t \sim$ Poisson (μ_t)
- $\ln(\mu_t) = 3.5 + 0.4\cos\left(\frac{20\pi(x_t+5)}{100}\right) + a_t$
- $a_t = \sum_{i=1}^{p} c_i \Big[\ln\left(y^*_{t-i}\right) - \left(3.5 + 0.4\cos\left(\frac{20\pi(x_t+5)}{100}\right)\right)\Big]$

- $y_t = \max(y, \tau), \tau = 0.5$

Here, a daily averaged temperature series is represented by x_t obtained from the Bangladesh Meteorological Department. x_t needs to be simulated for our simulation study but for our convenience we assume here real temperature value for x_t.

Data Simulation for case 1 (1,730)

To conduct a simulation study, data generated from the GAMAR model is essential. Implementing the following steps yields one observation from the GAMAR model:

Step 1: Set the true value $c_1 = 0.5$.
Step 2: Choose a single a_t from $a_t \sim Normal(0, 0.2)$.
Step 3: Simulate y_t from algorithm 2.

For the other combinations mentioned in the table, we have to follow the same procedure except the fact that the coefficient of AR terms will increase as the AR order increases from 1 to 4 and they are $c_1 = 0.5$. $c_2 = 0.25$, $c_3 = 0.12$, and $c_4 = 0.06$ and we have to generate as many a_t as the AR order.

The second model:

- $y_t \sim$ Poisson (μ_t)
- $\ln(\mu_t) = 4.8 + 0.2 \sin\left(\frac{\pi(x+3)}{28}\right) + a_t$
- $a_t = \sum_{i=1}^{p} c_i \Big[\ln\left(y^*_{t-i}\right) - \left(4.8 + 0.2 \sin\left(\frac{\pi(x+3)}{28}\right)\right) \Big]$
- $y_t = \max(y, \tau), \tau = 0.5$

The simulation of y_t can be performed in the same manner as described for the first model.

3.2 Results and Discussion

To evaluate residual autocorrelation, the ACF and PACF [30] of the Pearson residuals [5] were plotted over various lag periods for both GAM and GAMAR models.

Plotting estimates of temperature effects from the real model against the actual effect showed which model fits the data better.

3.2.1 Scenario 1

Our analysis aimed to determine the bias, relative error (RelErr), and coverage for each model configuration. Across all four cases (case 1 - case 4), the ACF and PACF plots of the GAM Pearson residuals revealed clear sign of autocorrelation Figures (3.1 - 3.4). For GAM we observed that the ACF tails off and the PACF cuts off after lag p (p = 1, 2, 3, 4) indicating that an AR (p) model would be reasonable. On the other hand, the ACF and PACF of GAMAR (p) were quite near to 0 for the similar data. Furthermore, Figures (3.5 - 3.8) showed that the predictive spline functions derived from GAMAR models align much more closely with the actual model compared to those from GAM.

Additionally, GAMAR models effectively control overdispersion which arose because of autocorrelationin of the data. The Pearson estimates of the dispersion parameter for both the GAM and GAMAR are presented for each of the four cases below:

Table 3.2: Pearson estimates of the dispersion parameter for GAM and GAMAR

Case	GAM	GAMAR
1	1.976	1.001
2	2.403	1.099
3	4.490	1.111
4	2.800	1.061

Since the dispersion parameter estimates for GAMAR are close to 1, it indicates that the model adequately captures the variability present in the data, without the need for additional adjustments to account for overdispersion. Autocorrelation produced overdispersion in the data since it was generated from GAMAR. GAM could not incorporate autocorrelation so the dispersion parameter from GAMs are greater than 1.

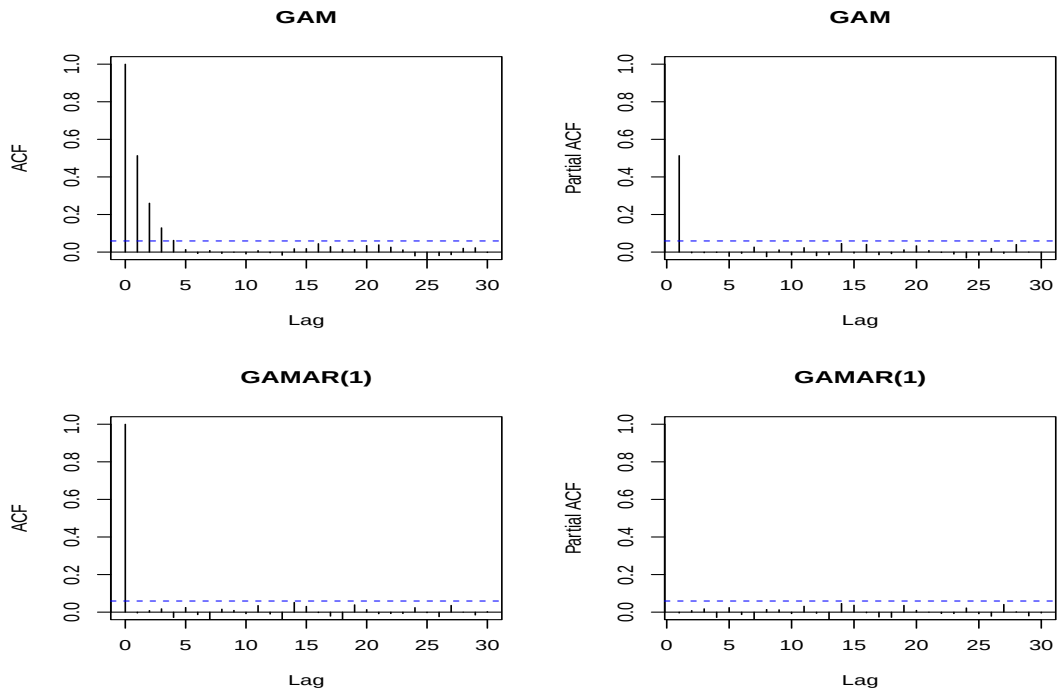


Figure 3.1: ACF and PACF of GAM and GAMAR (1) for case 1 from scenario 1

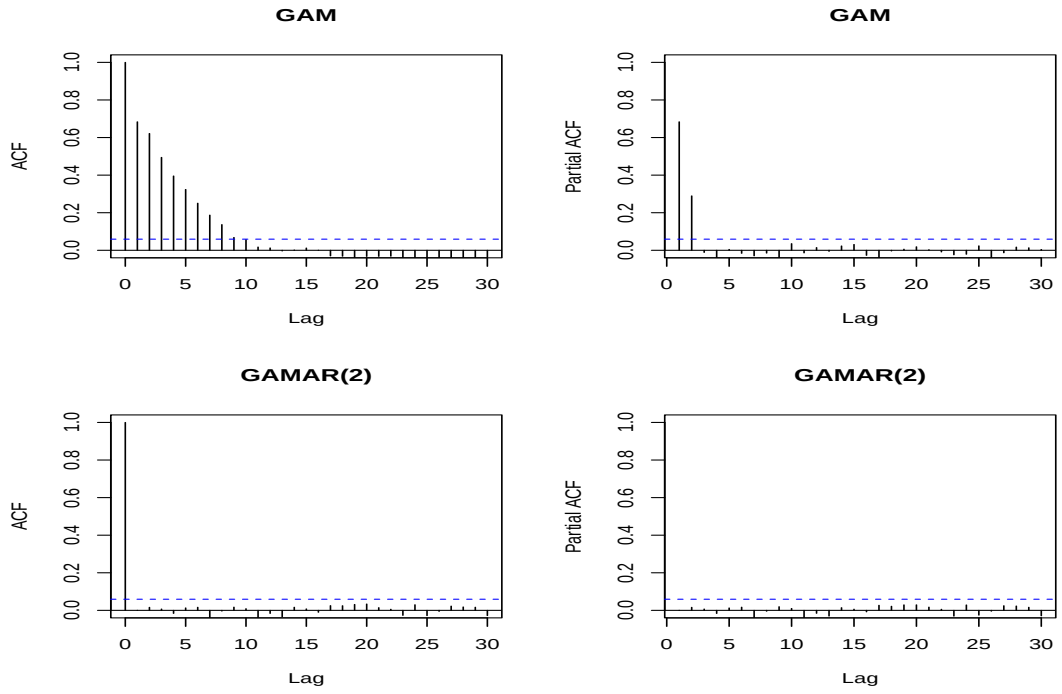


Figure 3.2: ACF and PACF of GAM and GAMAR (2) for case 2 from scenario 1

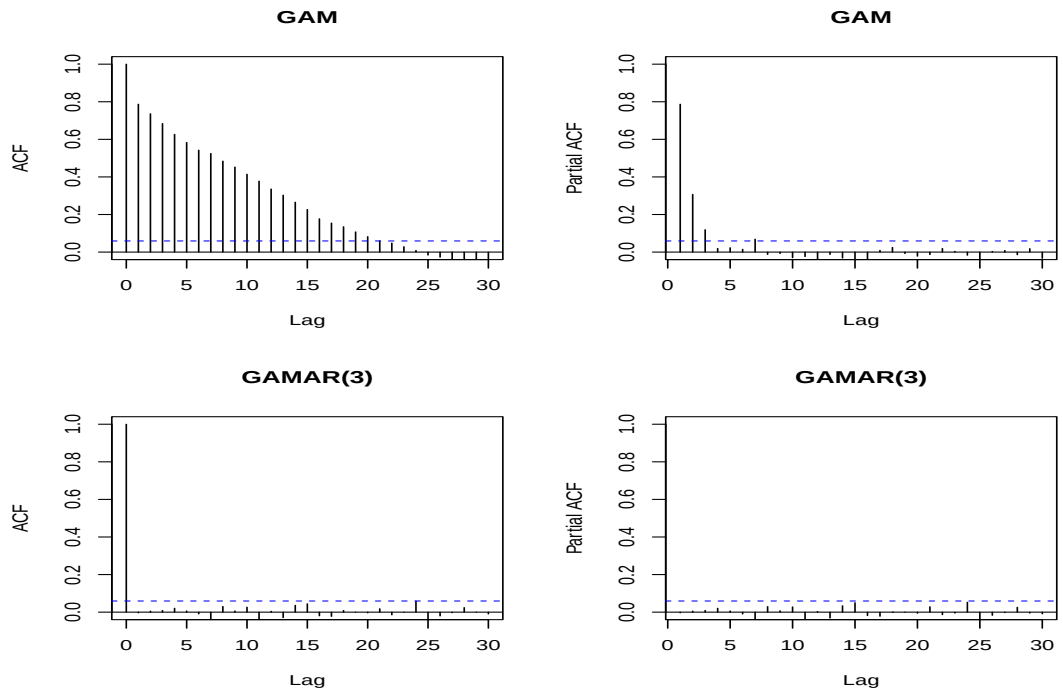


Figure 3.3: ACF and PACF of GAM and GAMAR (3) for case 3 from scenario 1

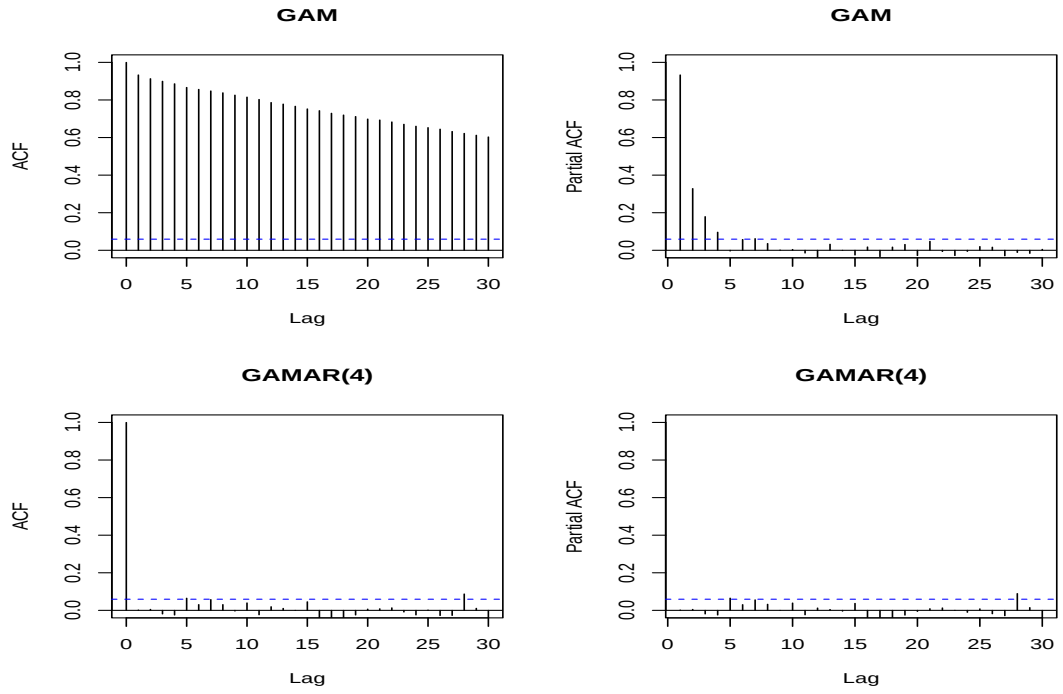


Figure 3.4: ACF and PACF of GAM and GAMAR (4) for case 4 from scenario 1

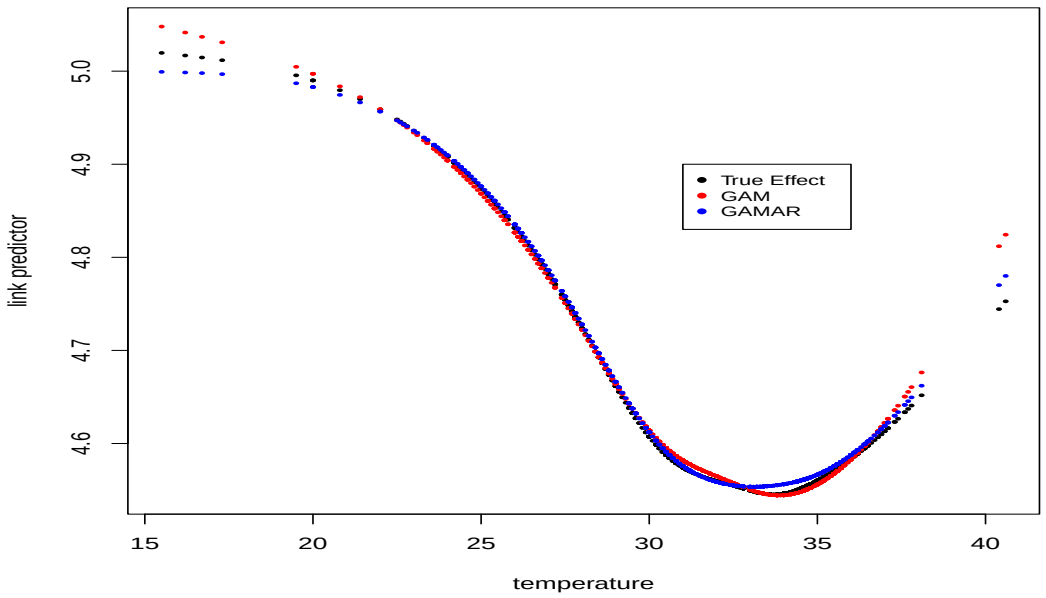


Figure 3.5: The temperature effects in link scale for case 1. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR (1)

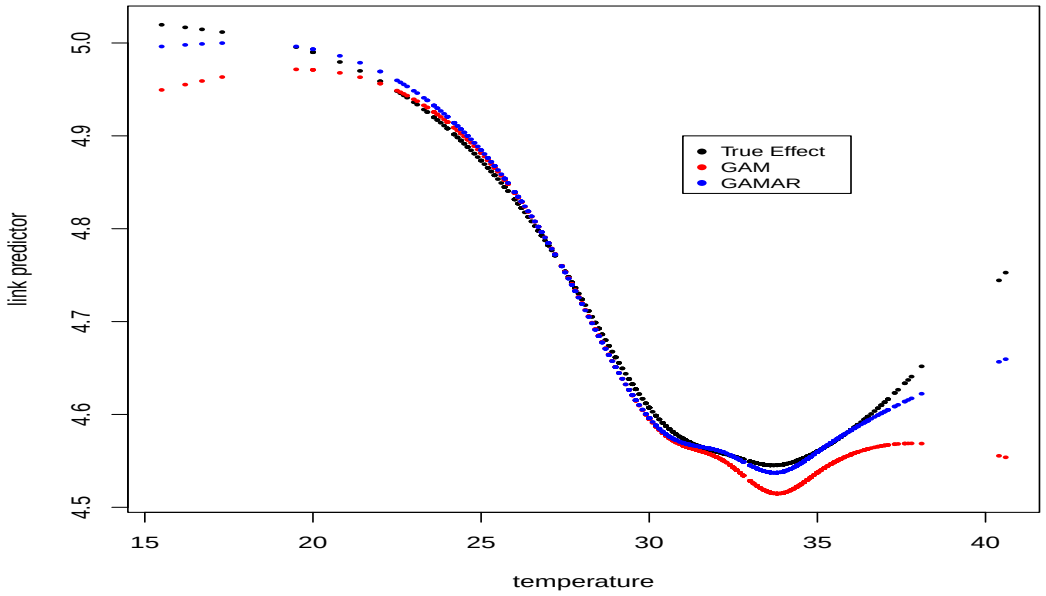


Figure 3.6: The temperature effects in link scale for case 2. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR (2)

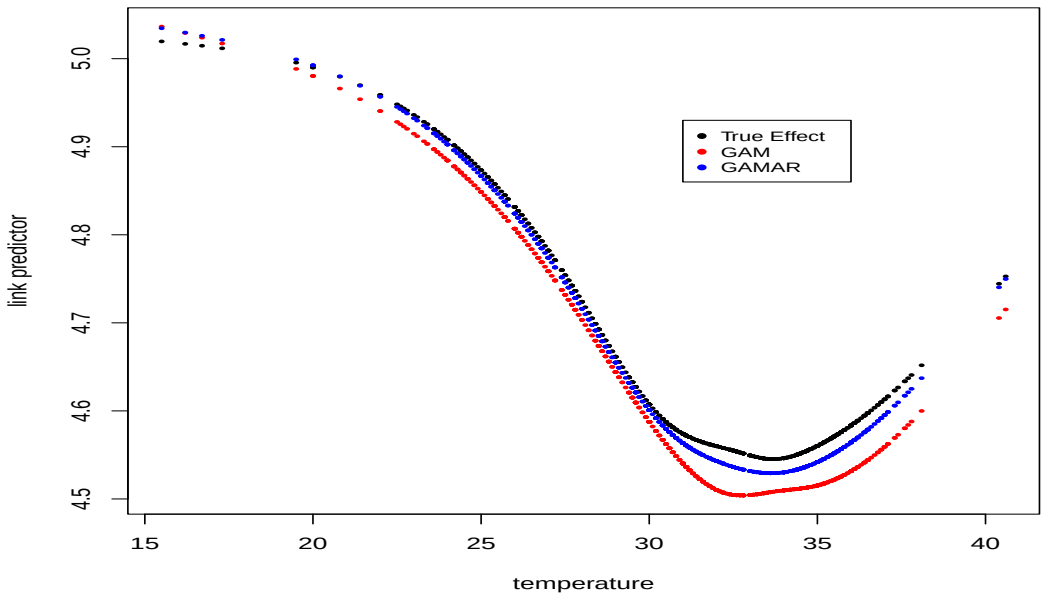


Figure 3.7: The temperature effects in link scale for case 3. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR (3)

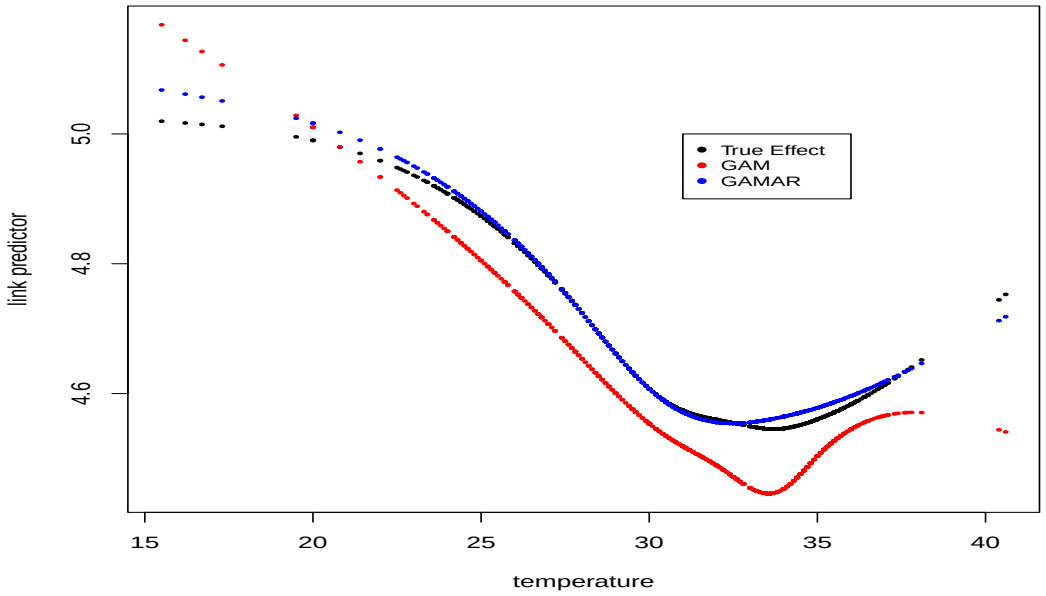


Figure 3.8: The temperature effects in link scale for case 4. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR (4)

From Tables 3.3 - 3.6 we have seen GAMAR models consistently showed lower biases and relative errors across all scenarios. This indicates that the addition of AR terms helps better capture the temporal dependencies, leading to more accurate parameter estimates. The coverage rates of the 95% confidence intervals generated by GAM were significantly lower than 95%, whereas those produced by GAMAR closely approximate 95%. Higher coverage rates in GAMAR models suggest that the confidence intervals were more reliable and likely to include the true parameter values. As the order of the AR terms increased from GAMAR (1) to GAMAR (4), there was a noticeable improvement in performance. GAMAR (4) demonstrated the best performance, suggesting that higher-order AR terms can enhance model accuracy and reliability.

For case 1, since the ACF cuts off and PACF tails off after lag 1 it is natural to use GAMAR (1) instead. The model is given below:

$$\ln(\mu_t) = ns(x_{1t}, 6) + c_1\Big(\ln\left(y^*_{t-1}\right) - ns(x_{t-1,1}, 6)\Big),$$

where $y^*_t = \max(y_t, \tau)$, $\tau = 0.5$.

For case 2, the model (GAMAR (2)) becomes:

$$\ln(\mu_t) = ns(x_{1t}, 6) + c_1\Big(\ln\left(y^*_{t-1}\right) - ns(x_{t-1,1}, 6)\Big) + c_2\Big(\ln\left(y^*_{t-2}\right) - ns\Big(x_{t-2,1}, 6\Big)\Big),$$

where $y^*_t = \max(y_t, \tau)$, $\tau = 0.5$.

For case 3, the model (GAMAR (3)) formulation adjusts as follows:

$$\begin{aligned}\ln(\mu_t) = {} & ns(x_{1t}, 6) + \ln\left(y^*_{t-1}\right) + c_1\left(\ln\left(y^*_{t-1}\right) - ns(x_{t-1,1}, 6)\right) \\ & + c_2\left(\ln\left(y^*_{t-2}\right) - ns(x_{t-2,1}, 6)\right) + c_3\left(\ln\left(y^*_{t-3}\right) - ns(x_{t-3,1}, 6)\right).\end{aligned}$$

where $y^*_t = \max(y_t, \tau)$, $\tau = 0.5$. For case 4, the model (GAMAR (4)) specification transforms to:

$$\begin{aligned}\ln(\mu_t) = {} & ns(x_{1t}, 6) + c_1\Big(\ln\left(y^*_{t-1}\right) - ns(x_{t-1,1}, 6)\Big) \\ & + c_2\Big(\ln\left(y^*_{t-2}\right) - ns(x_{t-2,1}, 6)\Big) + c_3\Big(\ln\left(y^*_{t-3}\right) - ns(x_{t-3,1}, 6)\Big) \\ & + c_4\Big(\ln\left(y^*_{t-4}\right) - ns(x_{t-4,1}, 6)\Big).\end{aligned}$$

where $y^*_t = \max(y_t, \tau)$, $\tau = 0.5$.

Table 3.3: Results from GAM and GAMAR (1) (case 1) in scenario 1

		GAM				GAMAR (1)			
	TruPar	MeaEst	Bias	RelErr	Coverage	MeaEst	Bias	RelErr	Coverage
β_0	5.02	5.0150	-.0050	0.0094	92.9	5.0219	0.0019	0.0076	95.4
β_1	-0.45	-0.4503	-0.0003	0.0937	94.1	-0.4531	-0.0031	0.0747	95.1
β_2	-0.46	-0.4590	0.0010	0.1233	92.1	-0.4632	-0.0032	0.0983	95.1
β_3	-0.48	-0.4792	0.0008	0.1028	93.2	-0.4825	-0.0025	0.0828	95.0
β_4	-0.43	-0.4250	-0.0050	0.0940	91.6	-0.4262	0.0038	0.0791	94.5
β_5	-0.38	-0.3766	0.0034	0.2948	92.4	-0.3855	-0.0055	0.2331	95.6
β_6	-0.25	-0.2482	0.0018	0.2076	88.2	-0.2483	0.0017	0.1634	95.2
	Mea_co		0.0247	0.1322	92.07		0.0031	0.0843	95.1
c_1	0.5					0.4968	-0.0032	0.0843	95.0
	Mea_ar						0.0032	0.0843	95.0

Bias: $\hat{\beta}_i - \beta_i$ or $\hat{c}_i - c_i$.

RelErr (Relative Error): $\left|\frac{\hat{\beta}_i - \beta_i}{\beta_i}\right|$ or $\left|\frac{\hat{c}_i - c_i}{c_i}\right|$.

Coverage: The percentage of estimated 95% CI which covers the true coefficient in all estimated 95% CI.

Mea_co: Mean absolute Bias, RelErr, Coverage for parameters of covariates.

Mea_ar: Mean absolute Bias, RelErr, Coverage for parameters of AR terms.

Table 3.4: Results from GAM and GAMAR (2) (case 2) in scenario 1

		GAM				GAMAR (2)			
	TruPar	MeaEst	Bias	RelErr	Coverage	MeaEst	Bias	RelErr	Coverage
β_0	5.02	5.0090	-.0110	0.0120	82.4	5.0209	0.0009	0.0083	93.7
β_1	-0.45	-0.4506	-0.0006	0.1264	83.0	-0.4532	-0.0032	0.0792	93.6
β_2	-0.46	-0.4607	-0.0007	0.1435	86.8	-0.4621	-0.0021	0.0952	93.8
β_3	-0.48	-0.4805	-0.0005	0.1315	84.3	-0.4826	-0.0026	0.0880	94.5
β_4	-0.43	-0.4232	0.0068	0.1075	86.4	-0.4246	0.0054	0.0773	94.3
β_5	-0.38	-0.3821	-0.0021	0.3465	86.3	-0.3857	-0.0057	0.2353	93.9
β_6	-0.25	-0.2490	0.0010	0.2461	81.6	-0.2480	0.0020	0.1604	94.3
	Mea_co		0.0032	0.1591	84.4		0.0031	0.1062	94.01
c_1	0.5					0.5008		0.0843	95.0
c_2	0.25				0.2375	-0.0125	0.1918	95.2	
	Mea_ar					0.0065	0.1437	95.1	

Bias: $\hat{\beta}_i - \beta_i$ or $\hat{c}_i - c_i$.

RelErr (Relative Error): $\left|\frac{\hat{\beta}_i - \beta_i}{\beta_i}\right|$ or $\left|\frac{\hat{c}_i - c_i}{c_i}\right|$.

Coverage: The percentage of estimated 95% CI which covers the true coefficient in all estimated 95% CI.

Mea_co: Mean absolute Bias, RelErr, Coverage for parameters of covariates.

Mea_ar: Mean absolute Bias, RelErr, Coverage for parameters of AR terms.

Table 3.5: Results from GAM and GAMAR (3) (case 3) in scenario 1

		GAM				GAMAR (3)			
	TruPar	MeaEst	Bias	RelErr	Coverage	MeaEst	Bias	RelErr	Coverage
β_0	5.02	4.9975	-.0225	0.0155	70.7	5.0199	-0.0001	0.0105	93.4
β_1	-0.45	-0.4538	-0.0038	0.1448	76.1	-0.4534	-0.0034	0.0776	94.3
β_2	-0.46	-0.4618	-0.0018	0.1617	82.5	-0.4623	-0.0023	0.0974	94.7
β_3	-0.48	-0.4818	-0.0018	0.1464	78.5	-0.4815	-0.0015	0.0851	94.1
β_4	-0.43	-0.4248	0.0052	0.1108	86.6	-0.4274	0.0026	0.0761	94.5
β_5	-0.38	-0.3844	-0.0044	0.3855	82.4	-0.3849	-0.0049	0.2409	94.3
β_6	-0.25	-0.2527	-0.0027	0.2783	77.3	-0.2496	0.0004	0.1653	93.6
	Mea_co		0.0061	0.1776	79.16		0.0022	0.0839	94.13
c_1	0.5					0.4994	-0.0006	0.0975	95.2
c_2	0.25					0.2375	-0.0125	0.2151	94.4
c_3	0.12					0.1127	-0.0073	0.4107	94.8
	Mea_ar						0.0068	0.2411	94.8

Bias: $\hat{\beta}_i - \beta_i$ or $\hat{c}_i - c_i$.

RelErr (Relative Error): $\left|\frac{\hat{\beta}_i - \beta_i}{\beta_i}\right|$ or $\left|\frac{\hat{c}_i - c_i}{c_i}\right|$.

Coverage: The percentage of estimated 95% CI which covers the true coefficient in all estimated 95% CI.

Mea_co: Mean absolute Bias, RelErr, Coverage for parameters of covariates.

Mea_ar: Mean absolute Bias, RelErr, Coverage for parameters of AR terms.

Table 3.6: Results from GAM and GAMAR (4) (case 4) in scenario 1

		GAM				GAMAR (4)			
	TruPar	MeaEst	Bias	RelErr	Coverage	MeaEst	Bias	RelErr	Coverage
β_0	5.02	4.9613	-0.0587	0.0237	53.4	5.0050	-0.0150	0.0161	90.7
β_1	-0.45	-0.4520	-0.0020	0.1824	66.9	-0.4527	-0.0027	0.0785	95.0
β_2	-0.46	-0.4562	-0.0038	0.2254	66.5	-0.4571	-0.0029	0.0976	95.9
β_3	-0.48	-0.4789	-0.0011	0.1878	68.8	-0.4811	-0.0011	0.0852	94.2
β_4	-0.43	-0.4231	0.0069	0.1185	83.6	-0.4250	0.0050	0.0744	95.0
β_5	-0.38	-0.3751	0.0049	0.5383	68.6	-0.3804	-0.0004	0.2417	95.4
β_6	-0.25	-0.2519	-0.0019	0.2963	75.6	-0.2495	0.0005	0.1640	94.5
	Mea_co		0.0113	0.2246	69.05		0.0039	0.1082	94.38
c_1	0.5					0.4965	-0.0035	0.0981	94.4
c_2	0.25					0.2416	-0.0084	0.2188	94.8
c_3	0.12					0.1140	-0.0060	0.4635	94.6
c_4	0.06					0.0536	-0.0064	0.8114	96.3
	Mea_ar						0.0061	0.3979	95.02

Bias: $\hat{\beta}_i - \beta_i$ or $\hat{c}_i - c_i$.

RelErr (Relative Error): $\left|\frac{\hat{\beta}_i - \beta_i}{\beta_i}\right|$ or $\left|\frac{\hat{c}_i - c_i}{c_i}\right|$.

Coverage: The percentage of estimated 95% CI which covers the true coefficient in all estimated 95% CI.

Mea_co: Mean absolute Bias, RelErr, Coverage for parameters of covariates.

Mea_ar: Mean absolute Bias, RelErr, Coverage for parameters of AR terms.

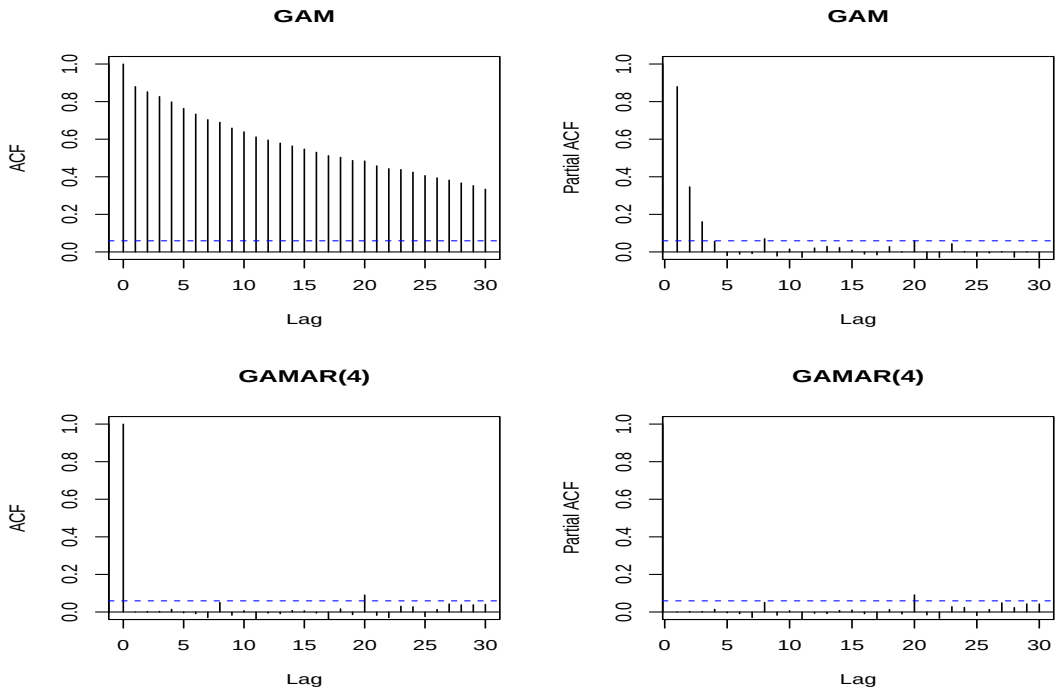


Figure 3.9: ACF and PACF of GAM and GAMAR (4) for case 16 from scenario 1

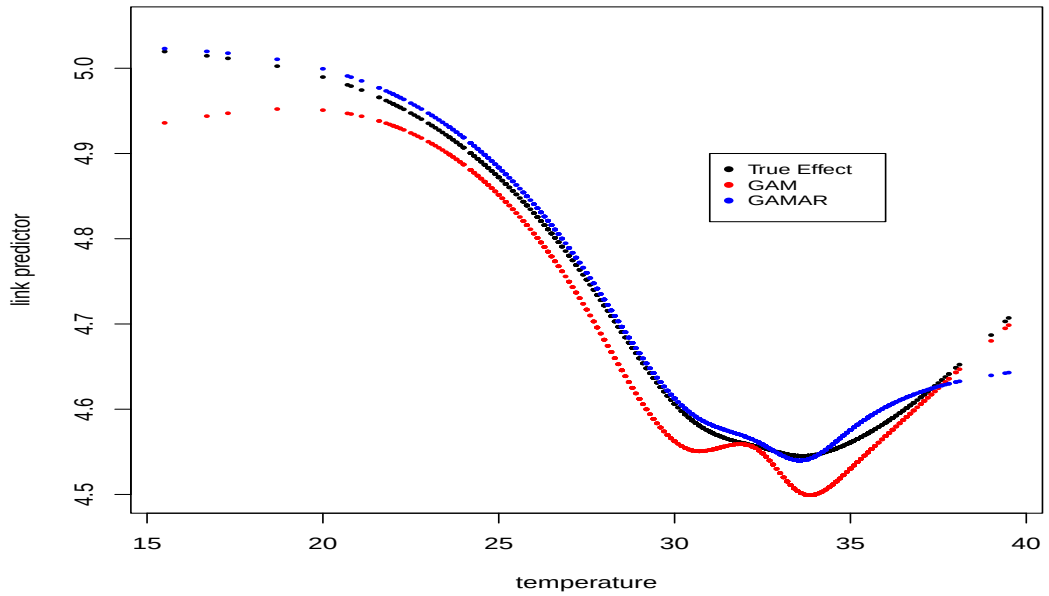


Figure 3.10: The temperature effects in link scale for case 16. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR (4)

Table 3.7: Results from GAM and GAMAR (4) (case 16) in scenario 1

		GAM				GAMAR (4)			
	TruPar	MeaEst	Bias	RelErr	Coverage	MeaEst	Bias	RelErr	Coverage
β_0	5.02	4.9667	-0.0533	0.0193	51.9	5.0156	-0.0044	0.0092	93.8
β_1	-0.45	-0.4458	0.0042	0.1685	59.4	-0.4454	0.0046	0.0639	95.3
β_2	-0.46	-0.4606	-0.0006	0.1859	60.0	-0.4600	-0.0000	-0.0053	94.9
β_3	-0.48	-0.4788	0.00121	0.1837	55.2	-0.4788	-0.0012	0.0657	95.3
β_4	-0.43	-0.4341	-0.0041	0.1202	62.4	-0.4353	-0.0053	0.0515	93.4
β_5	-0.38	-0.4054	-0.0254	0.4901	58.1	-0.4032	-0.0232	0.1872	94.4
β_6	-0.25	-0.2967	-0.0467	0.3888	52.5	-0.2948	-0.0448	0.2003	91.1
	Mea_co		0.0193	0.2223	57.07		0.0119	0.0930	94.02
c_1	0.5					0.5000	-0.0000	0.0475	95.7
c_2	0.25					0.2466	-0.0034	0.1081	94.5
c_3	0.12					0.1198	-0.0002	0.2154	95.9
c_4	0.06					0.0578	-0.0022	0.4034	95.8
	Mea_ar						0.0014	0.1936	95.47

Bias: $\hat{\beta}_i - \beta_i$ or $\hat{c}_i - c_i$.

RelErr (Relative Error): $\left|\frac{\hat{\beta}_i - \beta_i}{\beta_i}\right|$ or $\left|\frac{\hat{c}_i - c_i}{c_i}\right|$.

Coverage: The percentage of estimated 95% CI which covers the true coefficient in all estimated 95% CI.

Mea_co: Mean absolute Bias, RelErr, Coverage for parameters of covariates.

Mea_ar: Mean absolute Bias, RelErr, Coverage for parameters of AR terms.

3.2.2 Scenario 2

First Model

Just as in scenario 1, the ACF and PACF plots of the GAM Pearson residuals from the first sample for case 1-case 4 also showed clear autocorrelation (Figures 3.11-3.14). For GAM, we notice that the ACF gradually decreases and the PACF cuts off after lag p (p = 1, 2, 3, 4), indicating that an AR (p) model would be appropriate. Conversely, for the same data, the ACF and PACF of GAMAR (p) were both very close to 0 suggesting better fitting. The Pearson correlation coefficients between the estimated temperature effect and the actual effect also varied. The Table 3.8 presents the Pearson correlation coefficients between the estimated temperature effect and the true effect using two models: Generalized Additive Model (GAM) and Generalized Additive Model with Autoregressive terms (GAMAR). The coefficients indicate how well the estimated effects from these models correlate with the true effects. Higher values signify a stronger correlation and better performance in estimating the true effect.

Table 3.8: Pearson correlation coefficients

Case	GAM	GAMAR
First model		
1	0.9464	0.9510
2	0.8919	0.9340
3	0.9014	0.9910
4	0.7396	0.9232

Across all 4 cases, GAMAR consistently showed higher or comparable Pearson correlation coefficients compared to GAM, demonstrating that the inclusion of autoregressive terms generally improves the model's performance in estimating the true temperature effect. The GAMAR model appears to be particularly beneficial in cases where the simple GAM model has lower correlation coefficients, as seen in cases 1, 2, 3, and 4.

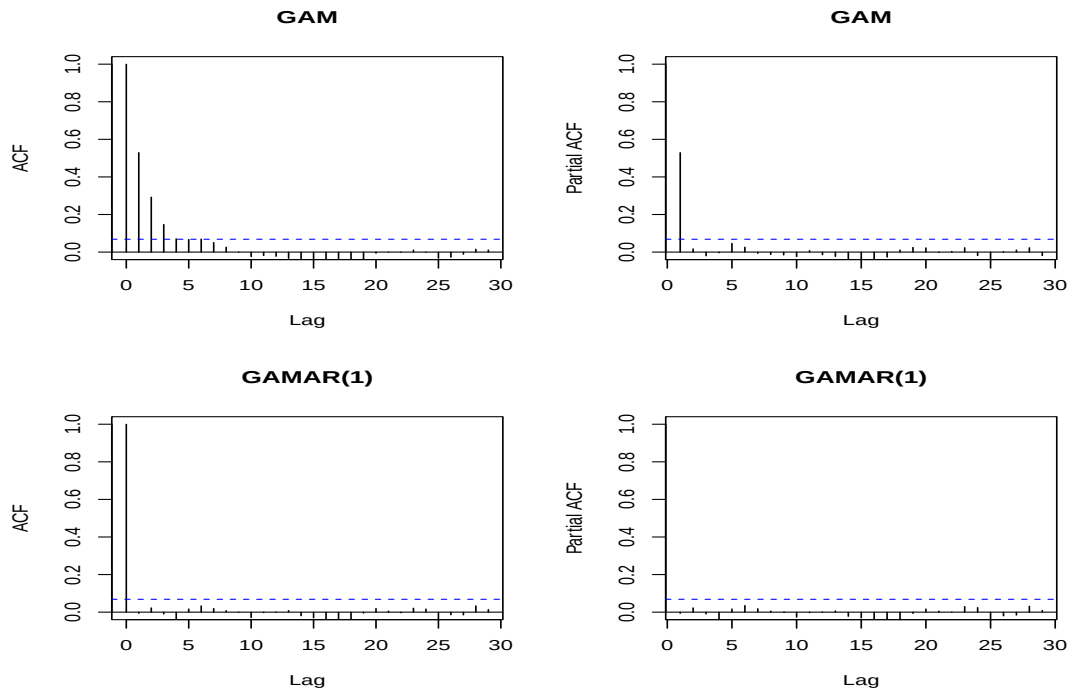


Figure 3.11: ACF and PACF of GAM and GAMAR (1) for case 1 using first model

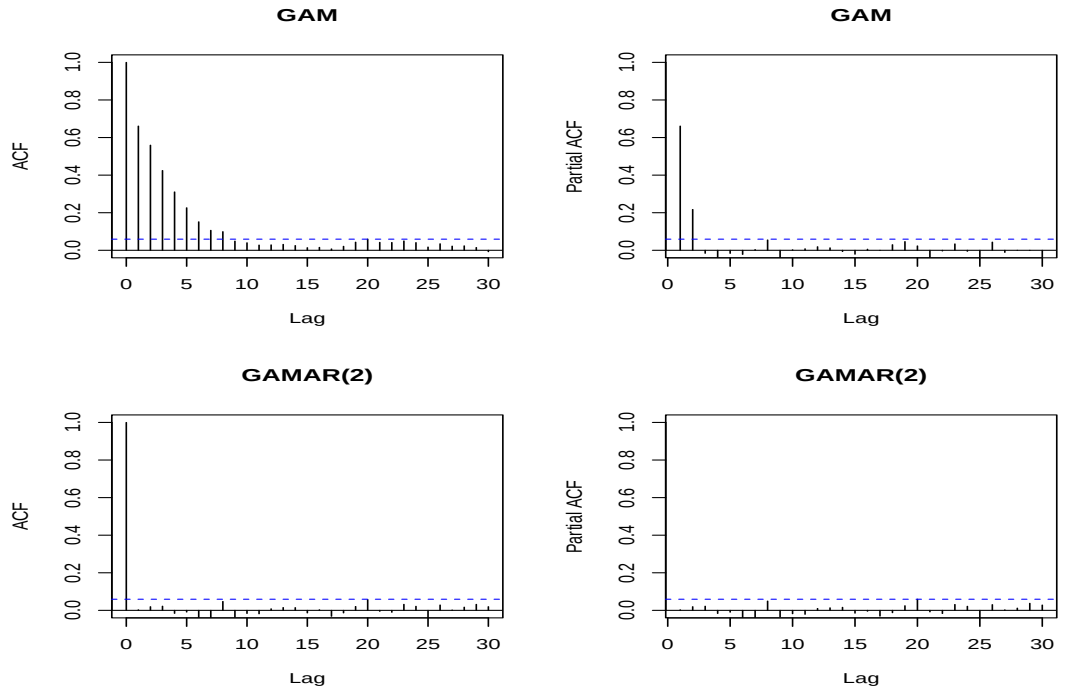


Figure 3.12: ACF and PACF of GAM and GAMAR (2) for case 2 using first model

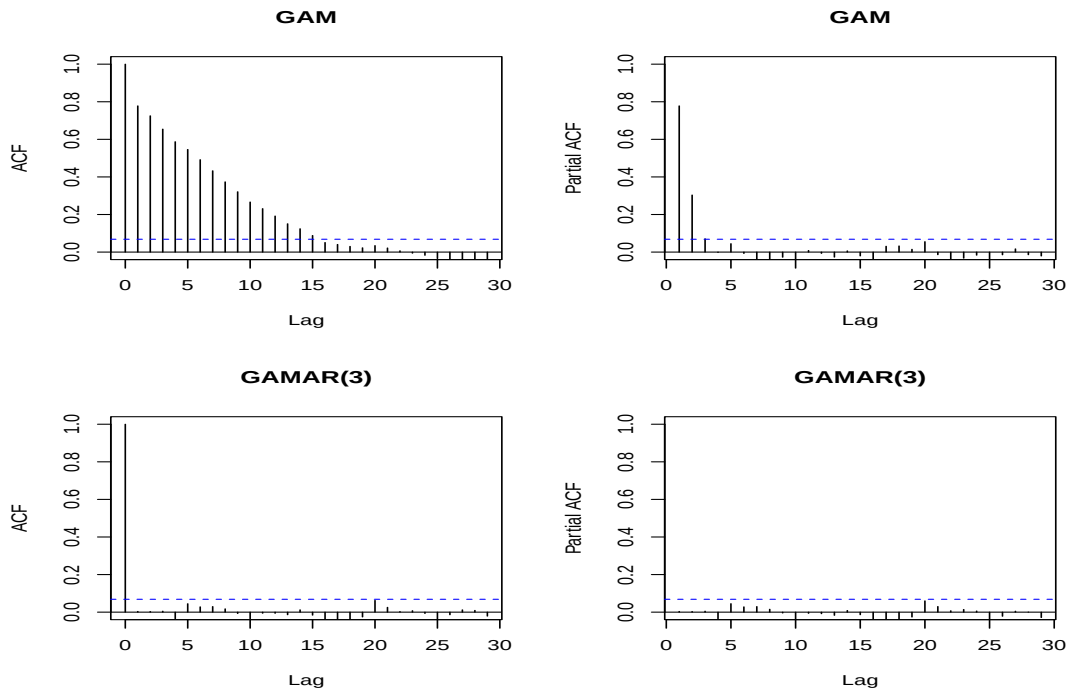


Figure 3.13: ACF and PACF of GAM and GAMAR (3) for case 3 using first model

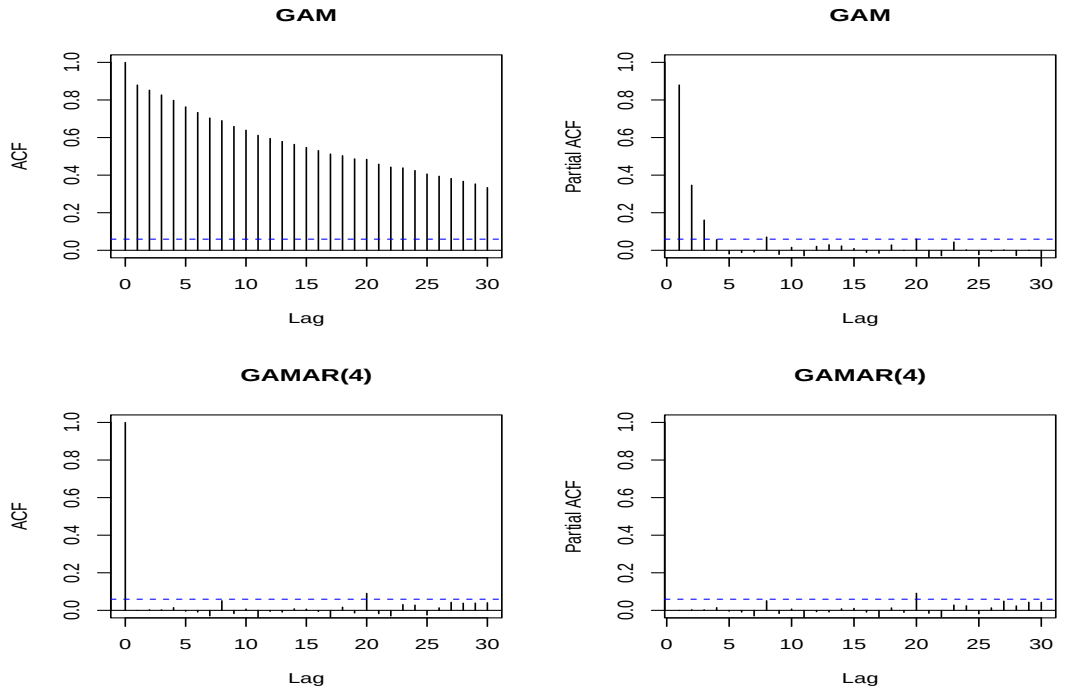


Figure 3.14: ACF and PACF of GAM and GAMAR (4) for case 4 using first model

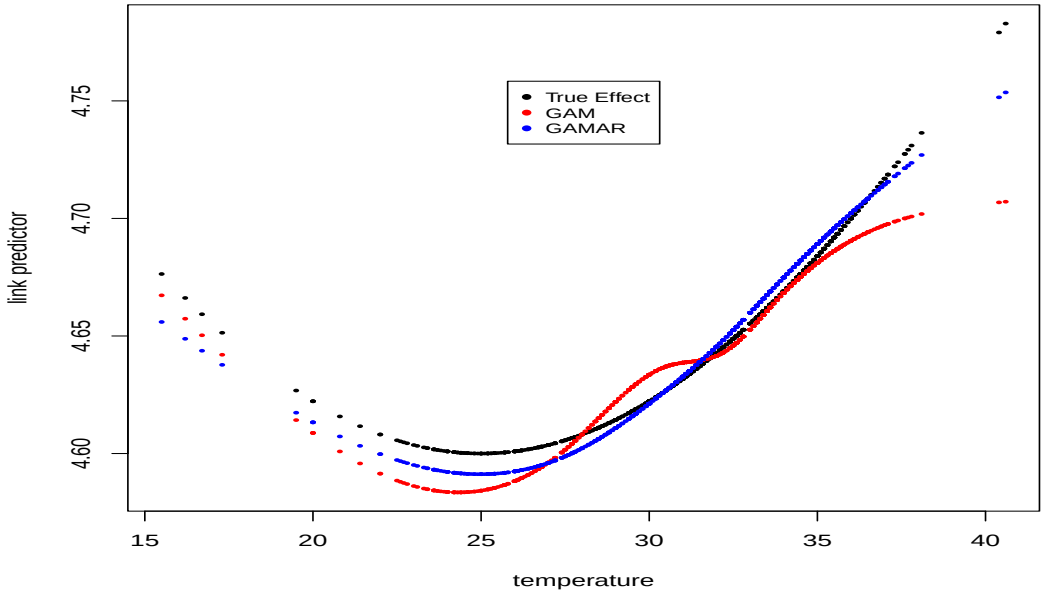


Figure 3.15: The temperature effects in link scale for case 1 using first model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(1)

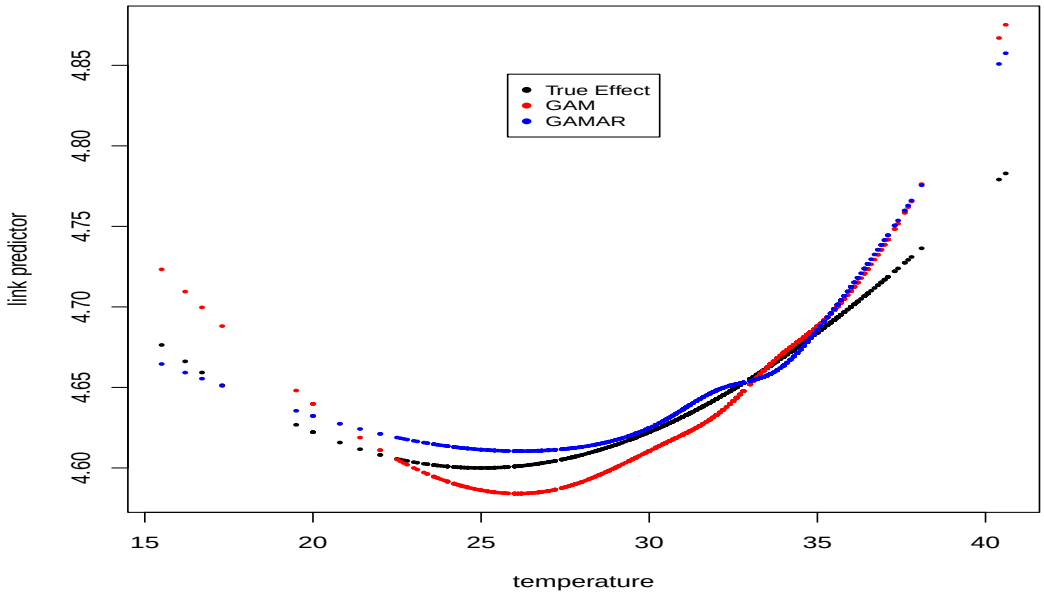


Figure 3.16: The temperature effects in link scale for case 2 using first model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(2)

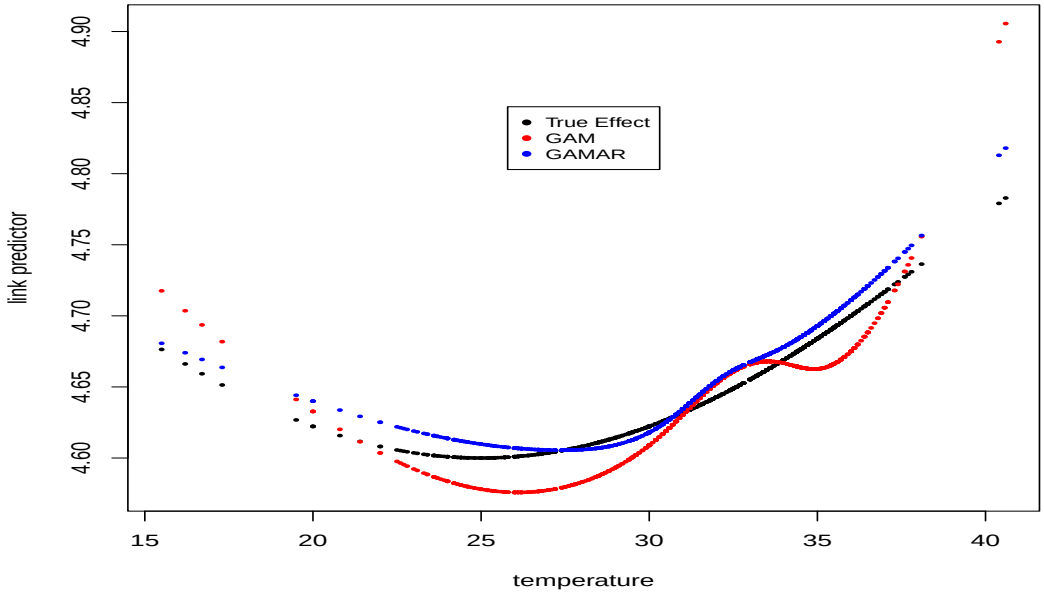


Figure 3.17: The temperature effects in link scale for case 3 using first model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(3)

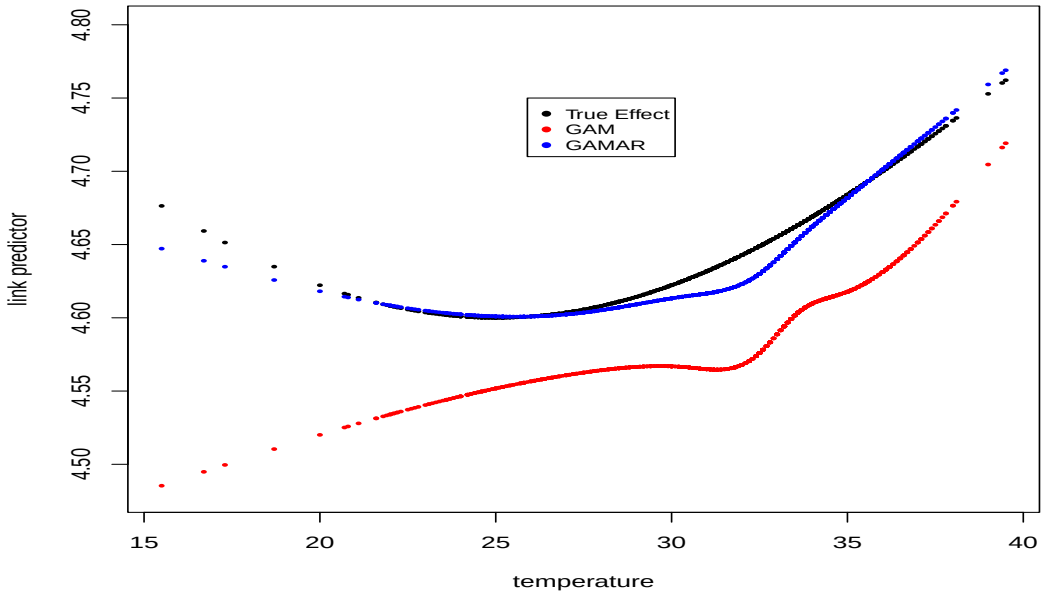
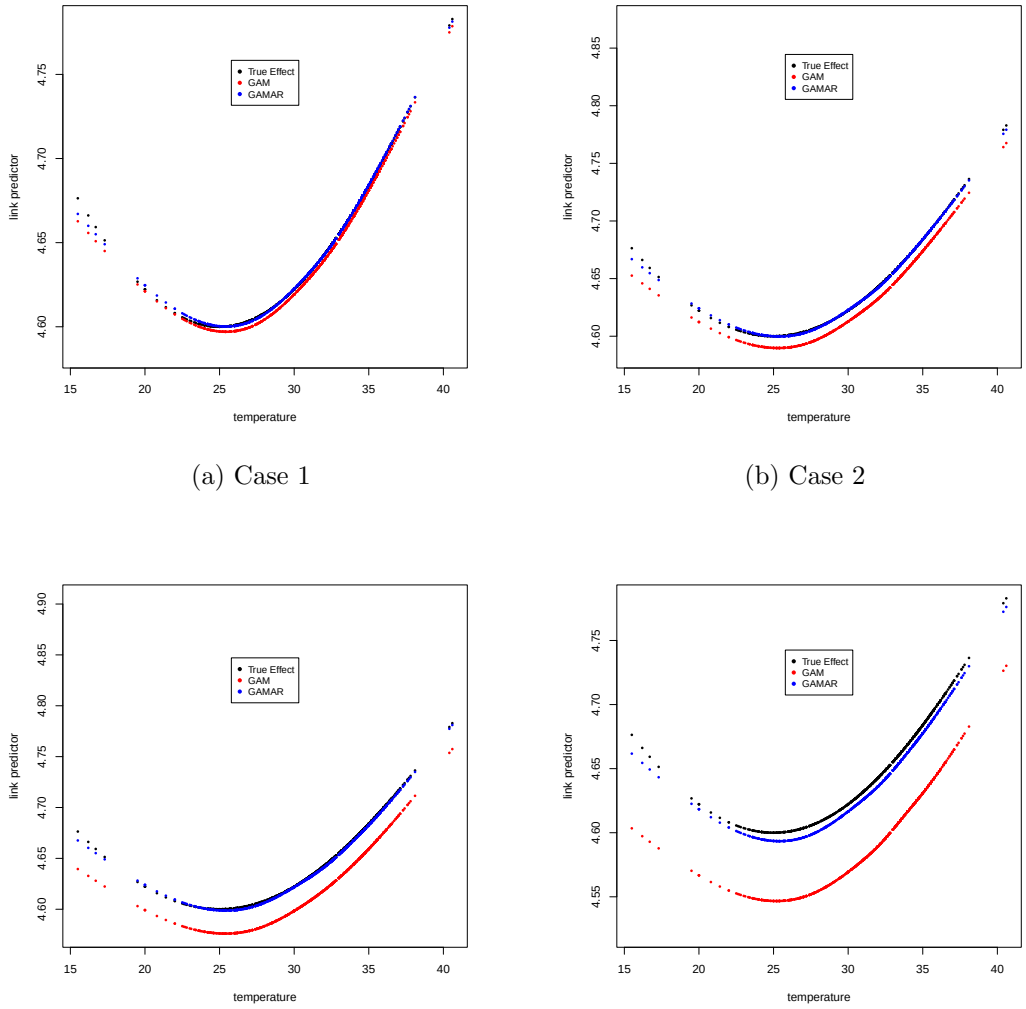


Figure 3.18: The temperature effects in link scale for case 4 using first model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(4)



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 3.19: The averaged temperature effects in link scale from scenario 2

Figures 3.15 - 3.18 demonstrated that the spline functions predicted by the GAMAR model aligned much more closely with the actual model compared to those predicted by the GAM model. We also examined the average estimated temperature effects derived from both the GAM and GAMAR models. Our findings indicate that the estimates from the GAMAR model were closer to the true effects (Figure 3.19), demonstrating the enhanced accuracy and reliability of incorporating autoregressive terms in the model.

Table 3.9: Results from GAM and GAMAR using first model

		GAM		GAMAR		
	TruPar	MeaEst	StdDev	MeaEst	StdDev2	DifStdDev
Case 1						
		4.6675	0.0667	4.6701	0.0543	0.0124
		-0.0440	0.0625	-0.0436	0.0489	0.0136
		-0.0269	0.0760	-0.0261	0.0631	0.0129
		-0.0086	0.0705	-0.0079	0.0559	0.0146
		0.0656	0.0517	-0.0652	0.0425	0.0032
		-0.0374	0.1523	-0.0360	0.1213	0.0310
		0.1218	0.0681	0.1224	0.0526	0.0155
c_1	0.5			0.4974	0.0537	
Case 2						
		4.6589	0.0800	4.6683	0.0563	0.0237
		-0.0418	0.0759	-0.0423	0.0494	0.0265
		-0.0254	0.0848	-0.0257	0.0595	0.0253
		-0.0057	0.0832	-0.0051	0.0554	0.0278
		0.0612	0.0579	0.0610	0.0405	0.0174
		-0.0378	0.1740	-0.0336	0.1234	0.0506
		0.1222	0.0772	0.1245	0.0519	0.0253
c_1	0.5			0.4999	0.0607	
c_2	0.25			0.2370	0.0607	
Case 3						
		4.6418	0.0950	4.6615	0.0716	0.0234
		-0.0382	0.0839	-0.0375	0.0514	0.0325
		-0.0204	0.0966	-0.0177	0.0624	0.0342
		-0.0022	0.0898	-0.0006	0.0578	0.0320
		0.0661	0.0602	0.0656	0.0408	0.0194
		-0.0228	0.1903	-0.0171	0.1276	0.0627
		0.1248	0.0874	0.1266	0.0526	0.0348
c_1	0.5			0.4980	0.0638	
c_2	0.25			0.2371	0.0708	
c_3	0.12			0.1145	0.0629	
Case 4						
		4.6197	0.1423	4.6580	0.1114	0.0309
		-0.0440	0.1017	-0.0412	0.0529	0.0488
		-0.0309	0.1269	-0.0257	0.0650	0.0619
		-0.0075	0.1093	-0.0036	0.0601	0.0492
		0.0602	0.0672	0.0608	0.0419	0.0253

Continued on next page

Table 3.9 – continued from previous page

		GAM		GAMAR		
	TruPar	MeaEst	StdDev	MeaEst	StdDev2	DifStdDev
		-0.0449	0.2510	-0.0314	0.1325	0.1185
		0.1218	0.0936	0.1274	0.0519	0.0417
c_1	0.5			0.4997	0.0634	
c_2	0.25			0.2404	0.0718	
c_3	0.12			0.1180	0.0705	
c_4	0.06			0.0469	0.0626	

The results from the GAM and GAMAR using the first model are presented in Table 3.9. The table displays the mean estimates (MeaEst), standard deviations (StdDev and StdDev2), and the differences in standard deviations (DifStdDev) for various parameters across four cases.

The mean estimates for TruPar showed minimal differences between the GAM and GAMAR models across all 4 cases (case 1-case 4). Similarly, other parameters mean estimates were very close between the two models, suggesting both models provide comparable mean estimates. The standard deviations for the parameter estimates were consistently lower in the GAMAR model compared to the GAM model. This indicates that the GAMAR model provides more precise estimates. The DifStdDev values, representing the difference in standard deviations between the GAM and GAMAR models, were generally positive. This indicates that the GAMAR model consistently reduces the variability in parameter estimates compared to the GAM model. The GAMAR model demonstrated high accuracy in estimating parameters close to their true values. Parameters such as c_1,c_2, c_3, and c_4, were also accurately estimated with the GAMAR model across different cases.

Second Model

The ACF and PACF plots of the GAM Pearson residuals from the first sample for case 1-case 4 showed clear autocorrelation (Figures 3.20 - 3.23). For GAM, we notice that the ACF gradually decreases and the PACF cuts off after lag p (p = 1, 2, 3, 4), indicating that an AR (p) model would be appropriate. Conversely, for the same data, the ACF and PACF of GAMAR (p) were both very close to 0 suggesting better fitting. The Pearson correlation coefficients between the estimated temperature effect and the actual effect also varied. The table 3.10 presents the Pearson correlation coefficients between the estimated temperature effect and the true effect using two models: Generalized Additive Model (GAM) and Generalized Additive Model with Autoregressive terms (GAMAR). The coefficients indicate how well the estimated effects from these models correlate with the true effects. Higher values signify a stronger correlation and better performance in estimating the true effect. Across all 4 cases, GAMAR consistently showed

Table 3.10: Pearson correlation coefficients

Case	GAM	GAMAR
Second model		
1	0.9641	0.9718
2	0.8928	0.9549
3	0.9691	0.9701
4	0.8783	0.9353

higher or comparable Pearson correlation coefficients compared to GAM, demonstrating that the inclusion of autoregressive terms generally improves the model's performance in estimating the true temperature effect. The GAMAR model appears to be particularly beneficial in cases where the simple GAM model has lower correlation coefficients, as seen in cases 1, 2, 3, and 4.

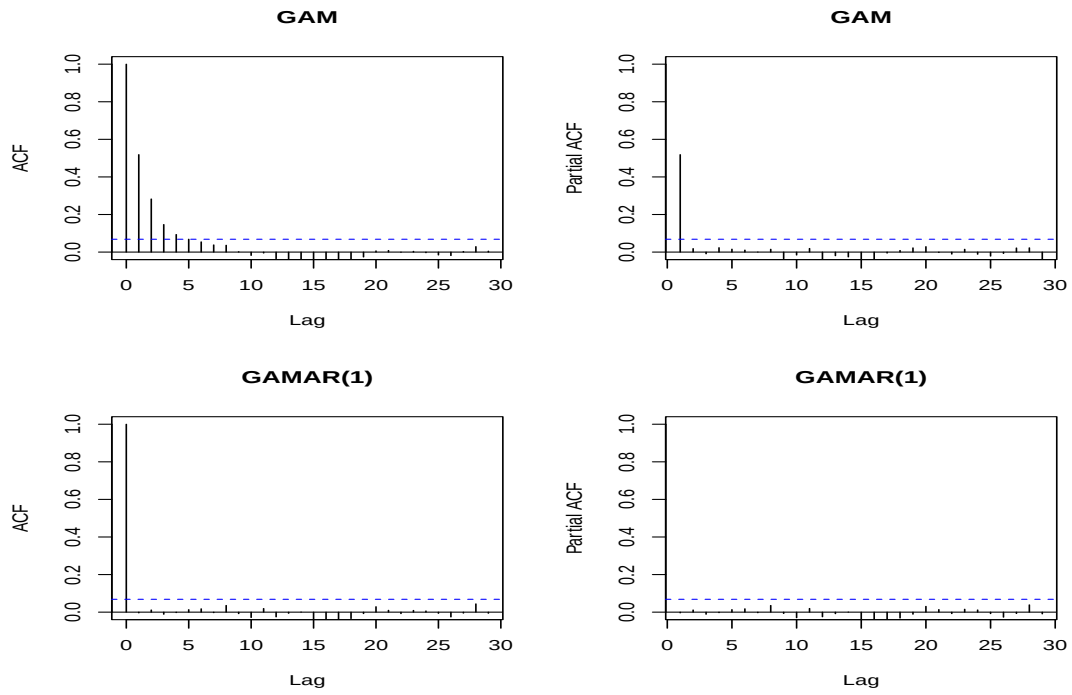


Figure 3.20: ACF and PACF of GAM and GAMAR (1) for case 1 using 2nd model

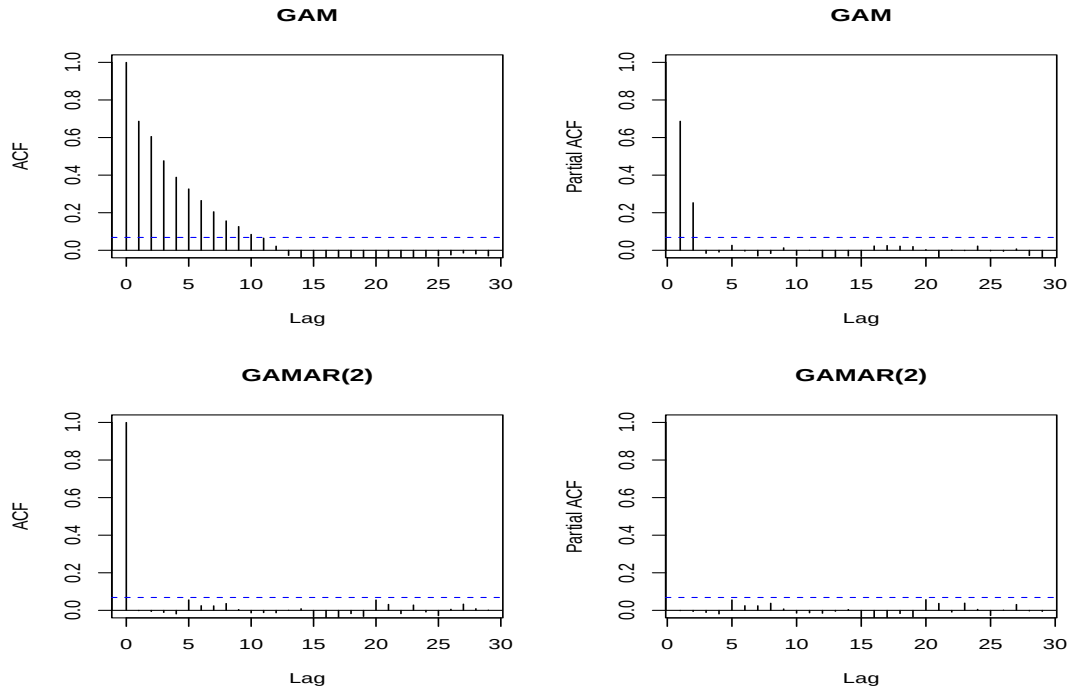


Figure 3.21: ACF and PACF of GAM and GAMAR (2) for case 2 using 2nd model

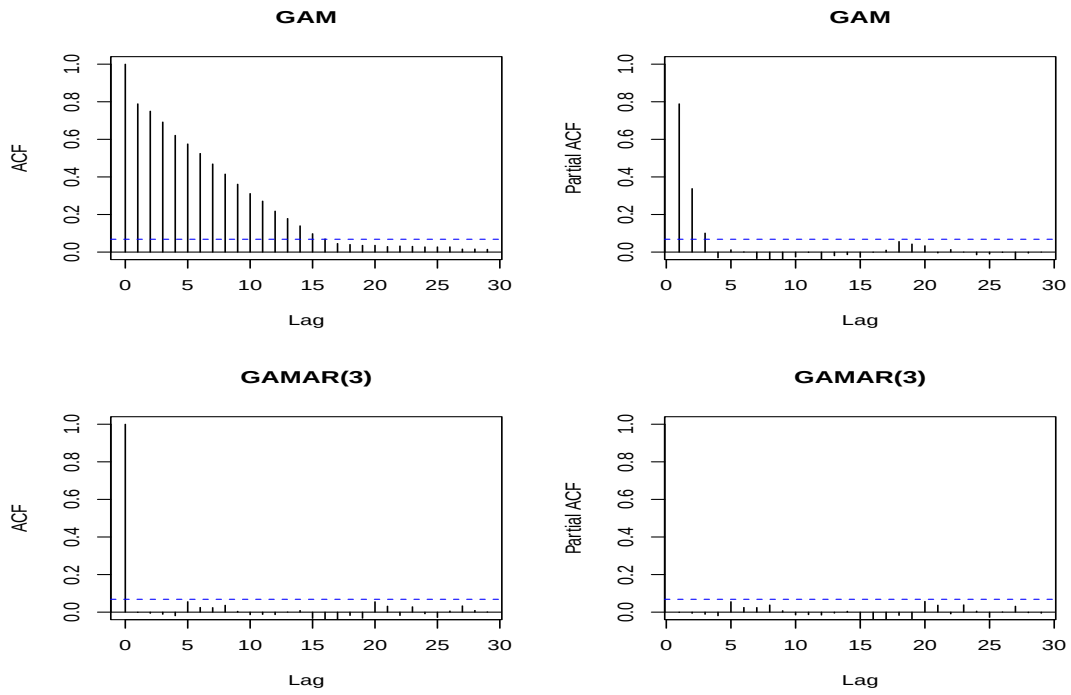


Figure 3.22: ACF and PACF of GAM and GAMAR (3) for case 3 using 2nd model

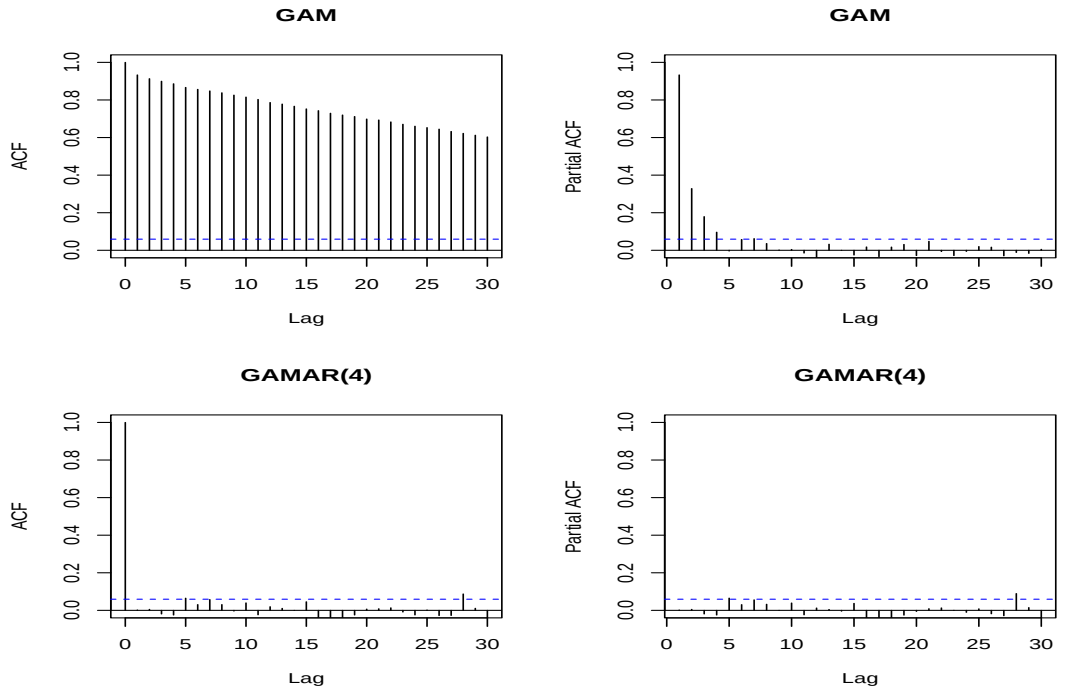


Figure 3.23: ACF and PACF of GAM and GAMAR (4) for case 4 using 2nd model

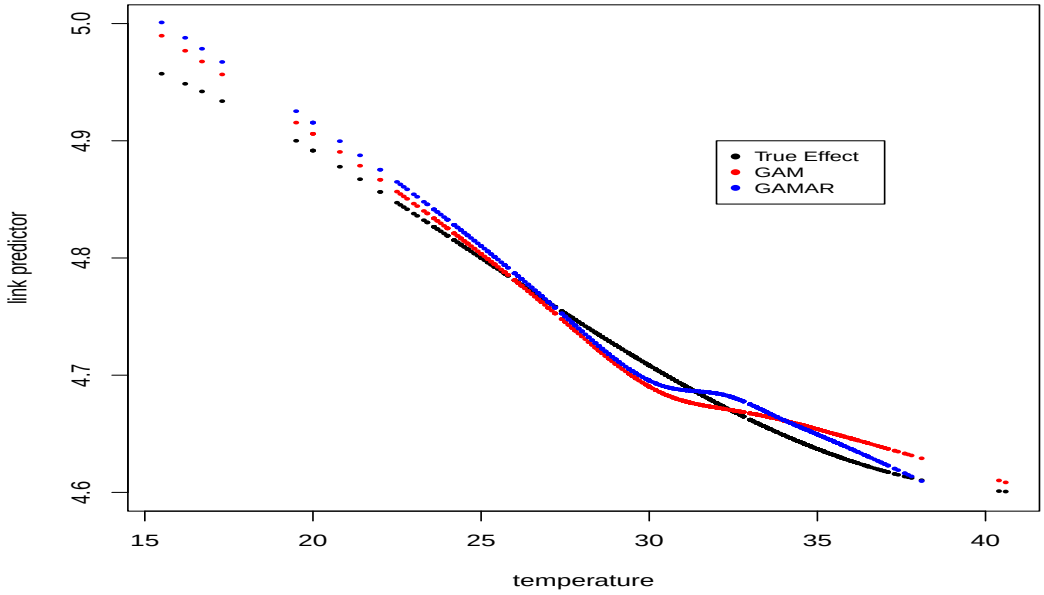


Figure 3.24: The temperature effects in link scale for case 1 using 2nd model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(1)

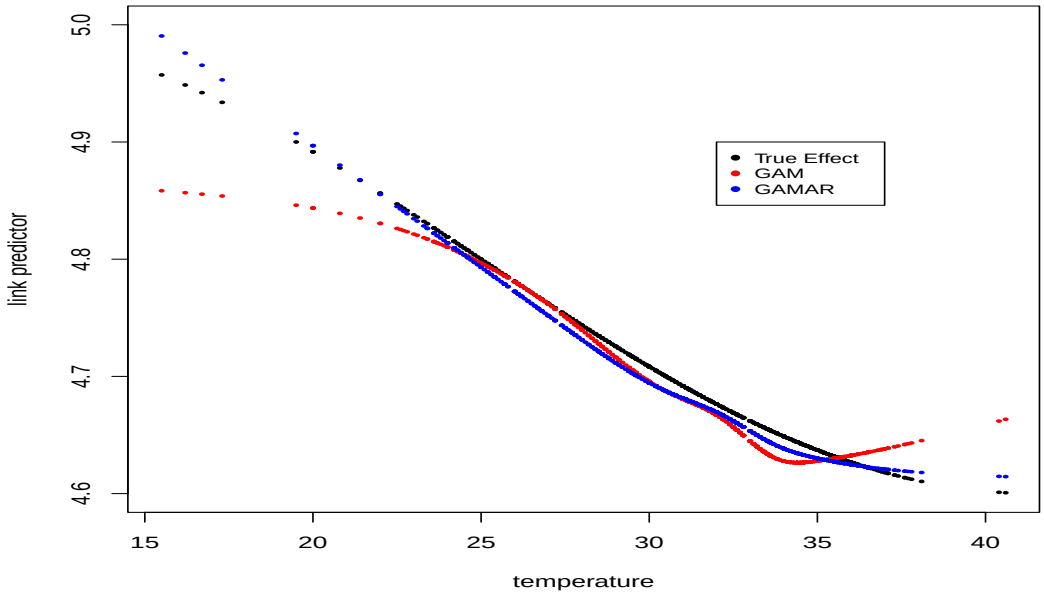


Figure 3.25: The temperature effects in link scale for case 2 using 2nd model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(2)

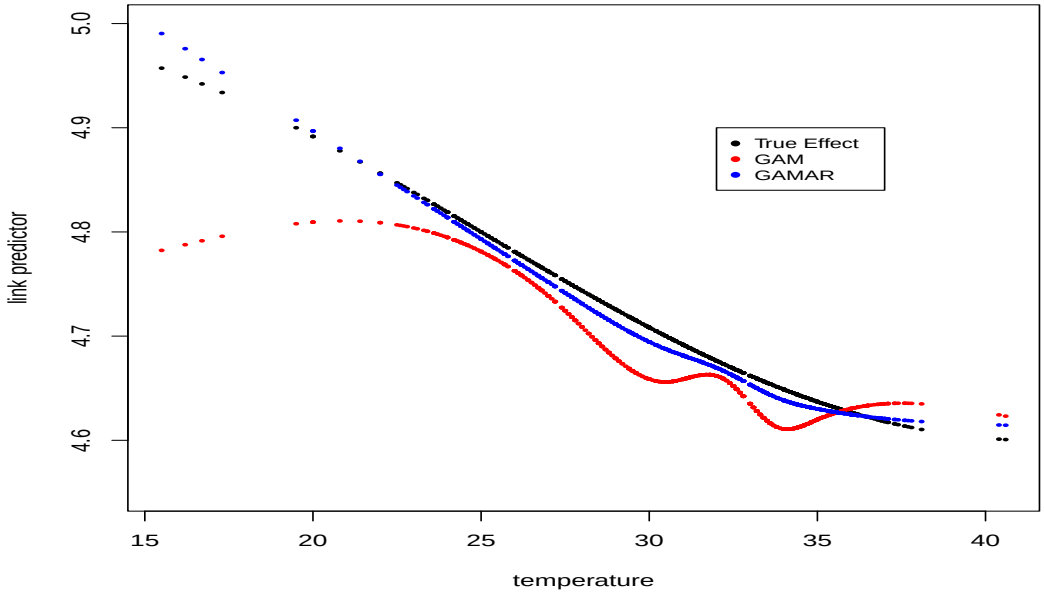


Figure 3.26: The temperature effects in link scale for case 3 using 2nd model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(3)

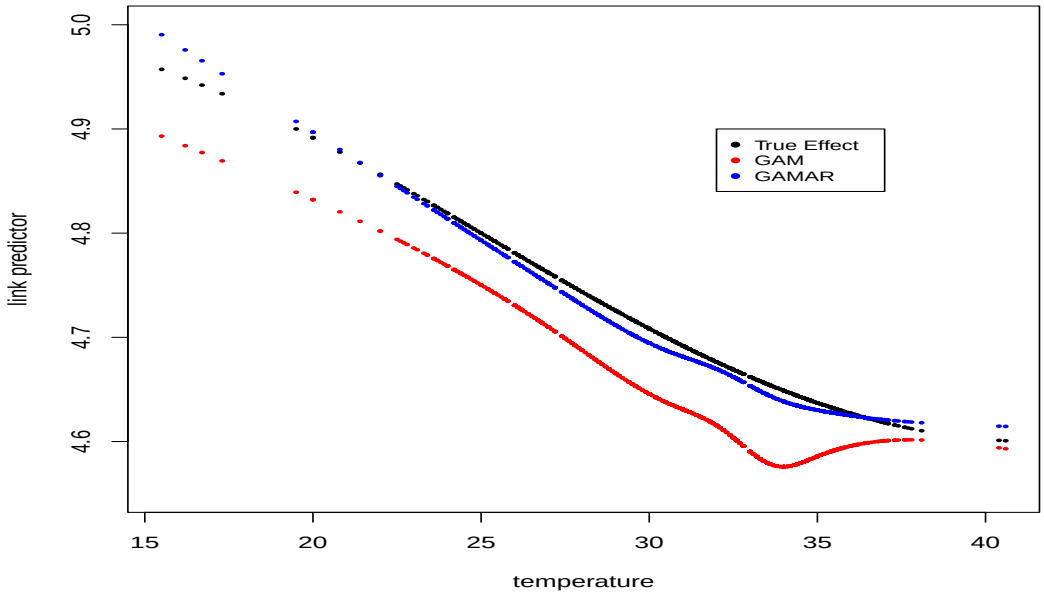
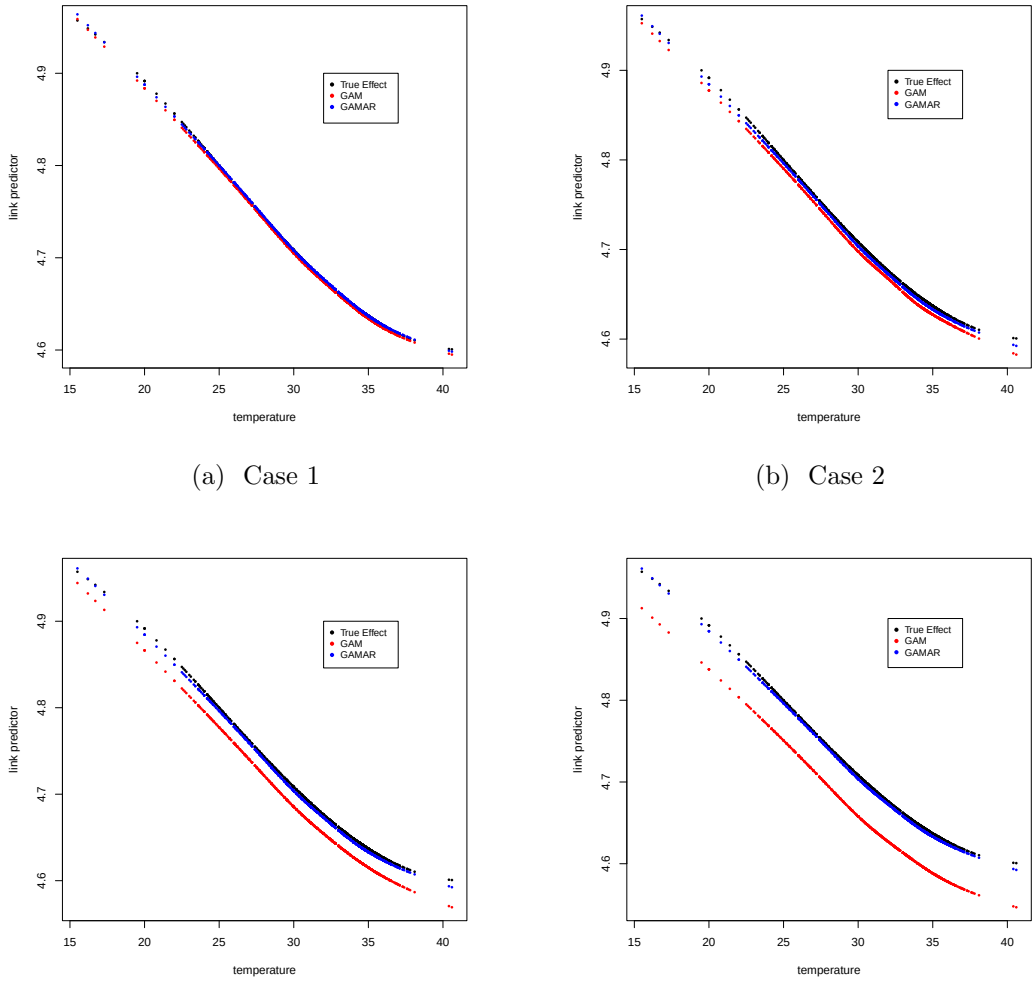


Figure 3.27: The temperature effects in link scale for case 4 using 2nd model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(4)



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 3.28: The averaged temperature effects in link scale using second model

Figures 3.24 through 3.27 showed that the spline functions predicted by the GAMAR model were much more closely aligned with the true model compared to those predicted by the GAM model. Additionally, we assessed the mean estimated temperature effects from both the GAM and GAMAR models (Figure 3.28). Our results indicate that the GAMAR model's estimates were nearer to the actual effects. Similar findings were observed in the first model as well, reinforcing the superior performance of the GAMAR model over the GAM model.

Table 3.11: Results from GAM and GAMAR using second model

		GAM		GAMAR		
	TruPar	MeaEst	StdDev	MeaEst	StdDev2	DifStdDev
Case 1						
		4.9620	0.0626	4.9673	0.0495	0.0131
		-0.2671	0.0591	-0.2691	0.0455	0.0136
		-0.2910	0.0721	-0.2938	0.0582	0.0139
		-0.3122	0.0672	-0.3137	0.0522	0.0150
		-0.2738	0.0515	-0.2759	0.0426	0.0089
		-0.4889	0.1444	-0.4940	0.1125	0.0319
		-0.2823	0.0696	-0.2813	0.0544	0.0152
c_1	0.5			0.4989	0.0518	
Case 2						
		4.9530	0.0730	4.9648	0.0512	0.0218
		-0.2631	0.0679	-0.2661	0.0428	0.0251
		-0.2886	0.0802	-0.2909	0.0541	0.0261
		-0.3100	0.0752	-0.3123	0.0512	0.0240
		-0.2749	0.0568	-0.2755	0.0398	0.0170
		-0.4856	0.1618	-0.4892	0.1118	0.0500
		-0.2841	0.0804	-0.2821	0.0519	0.0285
c_1	0.5			0.5002	0.0618	
c_2	0.25			0.2372	0.0610	
Case 3						
		4.9385	0.0931	4.9858	0.0675	0.0473
		-0.2639	0.0821	-0.2695	0.0444	0.0377
		-0.2890	0.0991	-0.2939	0.0564	0.0427
		-0.3097	0.0900	-0.3148	0.0506	0.0394
		-0.2756	0.0634	-0.2784	0.0414	0.0220
		-0.4822	0.1930	-0.4949	0.1129	0.0801
		-0.2810	0.0895	-0.2833	0.0539	0.0356
c_1	0.5			0.4994	0.0639	
c_2	0.25			0.2381	0.0702	
c_3	0.12			0.1125	0.0629	
Case 4						
		4.9157	0.1321	4.9568	0.0581	0.0411
		-0.2672	0.0951	-0.2687	0.0357	0.0594
		-0.2934	0.1157	-0.2919	0.0416	0.0595
		-0.3105	0.1035	-0.3128	0.0383	0.0509
		-0.2753	0.0648	-0.2759	0.0266	0.0382

Continued on next page

Table 3.11 – continued from previous page

		GAM		GAMAR		
	TruPar	MeaEst	StdDev	MeaEst	StdDev2	DifStdDev
		-0.4950	0.2356	-0.4919	0.0848	0.1188
		-0.2855	0.0925	-0.2829	0.0423	0.0502
c_1	0.5			0.4997	0.0634	
c_2	0.25			0.2383	0.0337	
c_3	0.12			0.1182	0.0327	
c_4	0.06			0.0476	0.0292	

The results of the GAM and GAMAR using the second model are summarized in Table 3.11. The table presented the mean estimates (MeaEst), standard deviations (StdDev and StdDev2), and the difference in standard deviations (DifStdDev) for various cases and parameters. The overall findings across the four cases are summarized below:

The mean estimates for TruPar were consistently similar between the GAM and GAMAR models across all cases. The mean estimates for other parameters also showed minimal differences between the two models, indicating that both models provide comparable mean estimates. Across all cases, the standard deviations for the parameter estimates were consistently lower in the GAMAR model compared to the GAM model. This indicates that the GAMAR model provides more precise estimates. The DifStdDev values, which represent the difference in standard deviations between the GAM and GAMAR models, were generally positive. This indicates that the GAMAR model consistently reduces the variability in parameter estimates compared to the GAM model.

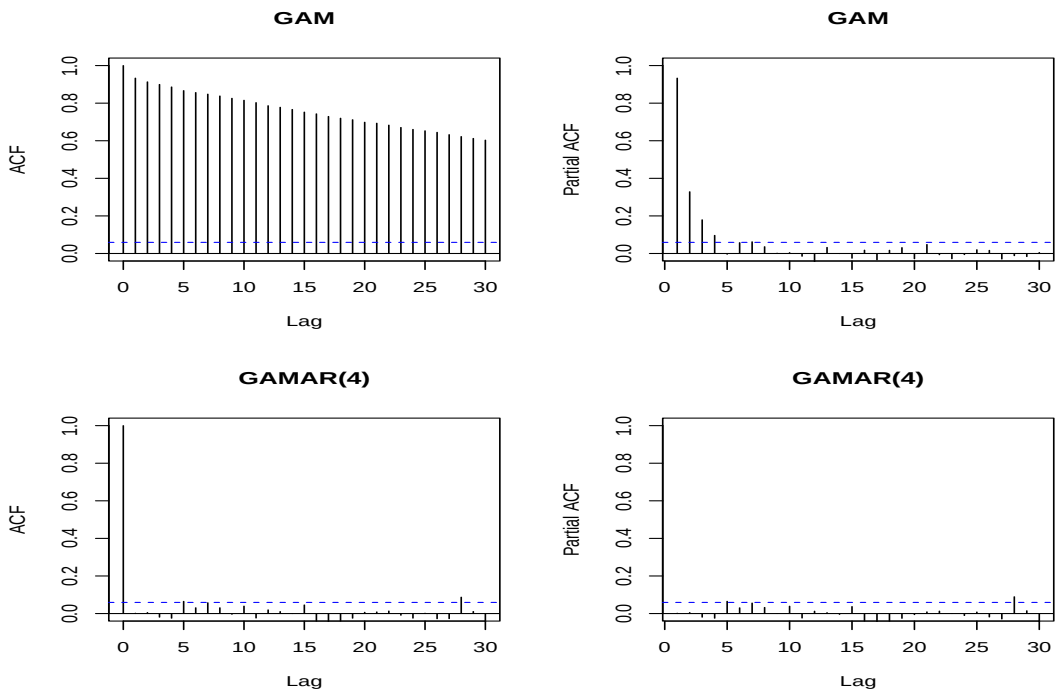


Figure 3.29: ACF and PACF of GAM and GAMAR (4) for case 16 using first model

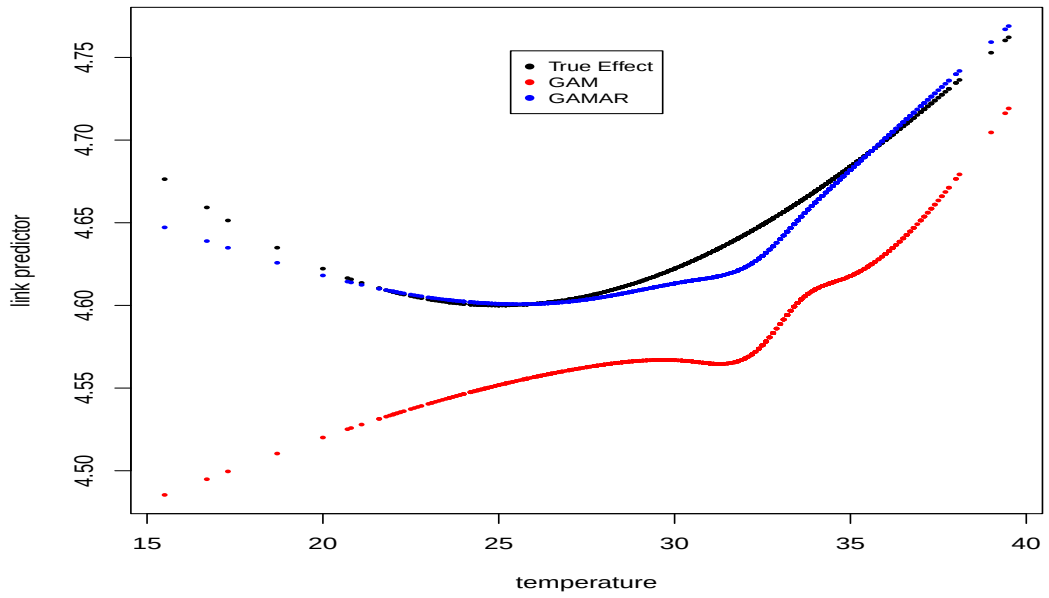


Figure 3.30: The temperature effects in link scale for case 16 using first model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(4)

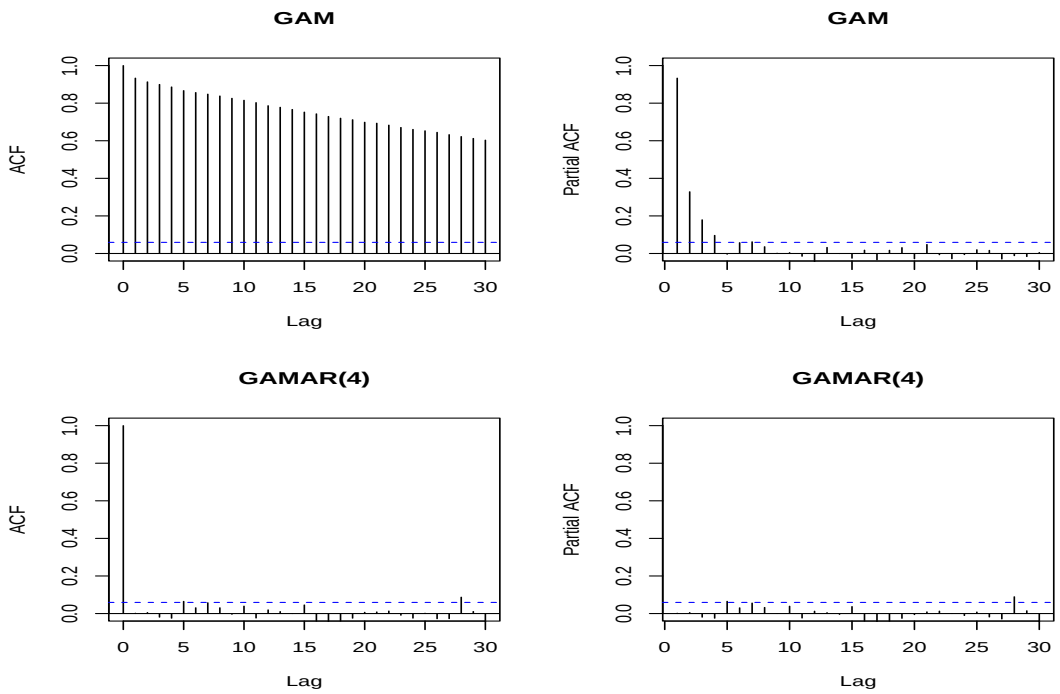


Figure 3.31: ACF and PACF of GAM and GAMAR (4) for case 16 using 2nd model

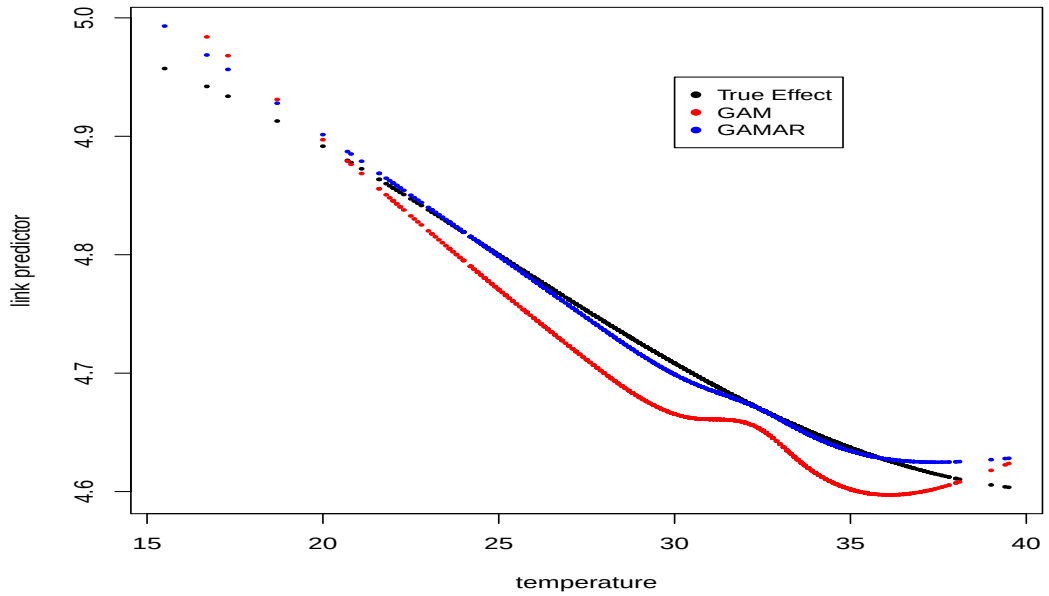


Figure 3.32: The temperature effects in link scale for case 16 using 2nd model. Black: The true effect, Red: $\widehat{ns}(x, 6)$ from GAM, Blue: $\widehat{ns}(x, 6)$ from GAMAR(4)

Table 3.12: Results from GAM and GAMAR(4) for case 16

		GAM		GAMAR(4)		
	TruPar	MeaEst	StdDev	MeaEst	StdDev2	DifStdDev
First Model						
		4.6102	0.1118	4.6628	0.0632	0.0486
		-0.0378	0.0946	-0.0423	0.0417	0.0529
		-0.0168	0.1097	-0.0224	0.0472	0.0625
		0.0027	0.1104	-0.0023	0.0444	0.0660
		0.0694	0.0631	0.0673	0.0287	0.0344
		-0.0038	0.2351	-0.0136	0.0969	0.1382
		0.1404	0.1108	0.1414	0.0406	0.0702
c_1	0.5			0.5000	0.0299	
c_2	0.25			0.2463	0.0324	
c_3	0.12			0.1197	0.0325	
c_4	0.06			0.0581	0.0298	
Second Model						
		4.9151	0.1104	4.9632	0.0581	0.0523
		-0.2603	0.0918	-0.2634	0.0357	0.0561
		-0.2890	0.1058	-0.2923	0.0416	0.0642
		-0.3105	0.1067	-0.3140	0.0383	0.0684
		-0.2807	0.0600	-0.2818	0.0266	0.0334
		-0.4875	0.2287	-0.4934	0.0848	0.1439
		-0.2866	0.1066	-0.2892	0.0423	0.0643
c_1	0.5			0.5002	0.0294	
c_2	0.25			0.2465	0.0337	
c_3	0.12			0.1195	0.0327	
c_4	0.06			0.0576	0.0292	

For all 16 cases, the simulation results were nearly identical for both models, so only the results for Case 16 were presented here. The findings indicate that the GAMAR model outperforms the GAM model. Furthermore, as the sample sizes AR order increases, the GAM model begins to show improved performance.

Chapter 4

Application of GAMAR in Real Life Data

4.1 Data Overview

This section provides a detailed overview of the data utilized in the application of the Generalized Additive Model with Autoregressive Terms (GAMAR) to study the relationship between dengue infected cases and environmental factors in Dhaka, Bangladesh. The primary data sources include the Directorate General of Health Services (DGHS), the Bangladesh Meteorological Department (BMD), and Time and Date, a reliable online source for weather and climate information (http://www.timeanddate.com/weather/bangladesh/dhaka/ext). The datasets encompass daily records of dengue case counts and various environmental variables, including temperature, rainfall, visibility, wind speed, and humidity, for the years 2022 and 2023.

Dependent Variable

The data on dengue infected cases were obtained from the Directorate General of Health Services (DGHS) in Bangladesh. The DGHS maintains comprehensive records of dengue incidence, providing crucial information for epidemiological studies. This dataset covers the years 2022 and 2023, offering a detailed temporal perspective on dengue incidence in Dhaka city. The primary variable is the daily count of dengue infected cases.

Independent Variables

Environmental factors, which play a significant role in the transmission dynamics of dengue [18], were obtained from the Bangladesh Meteorological Department (BMD). The dataset includes:

- **Temperature:** Daily average temperatures recorded at meteorological stations in Dhaka.
- **Rainfall:** Daily rainfall amounts, providing insight into precipitation patterns and potential breeding sites for mosquitoes in Dhaka.
- **Visibility:** Daily visibility records, indicating atmospheric clarity which can influence mosquito activity and human outdoor exposure.
- **Wind Speed:** Daily wind speed measurements, which can affect mosquito dispersal and breeding site conditions.

Data on daily average humidity levels, crucial for understanding the environmental conditions in Dhaka were collected from Time and Date, a reliable online source for weather and climate information.

4.2 Exploratory Data Analysis

Dengue Status in Dhaka

The number of total dengue cases in 2023 (108,841) was substantially higher than in 2022 (39,920), reflecting a significant increase in dengue incidence in Dhaka city, with 2023 experiencing nearly three times as many cases as 2022. In 2022, the highest number of dengue cases occurred in October, totaling 14,135 cases. In contrast, in 2023, the peak month for dengue cases shifted to August, recording 28,821 cases. On November 3, 2022, the highest daily tally of dengue cases reached 898. Conversely, on July 26, 2023, the peak daily count surged to 1,327 cases. Figure 4.1 below provides a visual comparison of the monthly distribution of dengue cases in Dhaka city for the years 2022 and 2023.

Table 4.1: Monthly dengue infected cases

Month	2022	2023
January	79	272
February	10	76
March	11	72
April	19	92
May	161	864
June	701	4701
July	1283	21956
August	2887	28821
September	7090	25201
October	14135	15947
November	10791	8667
December	2753	2172

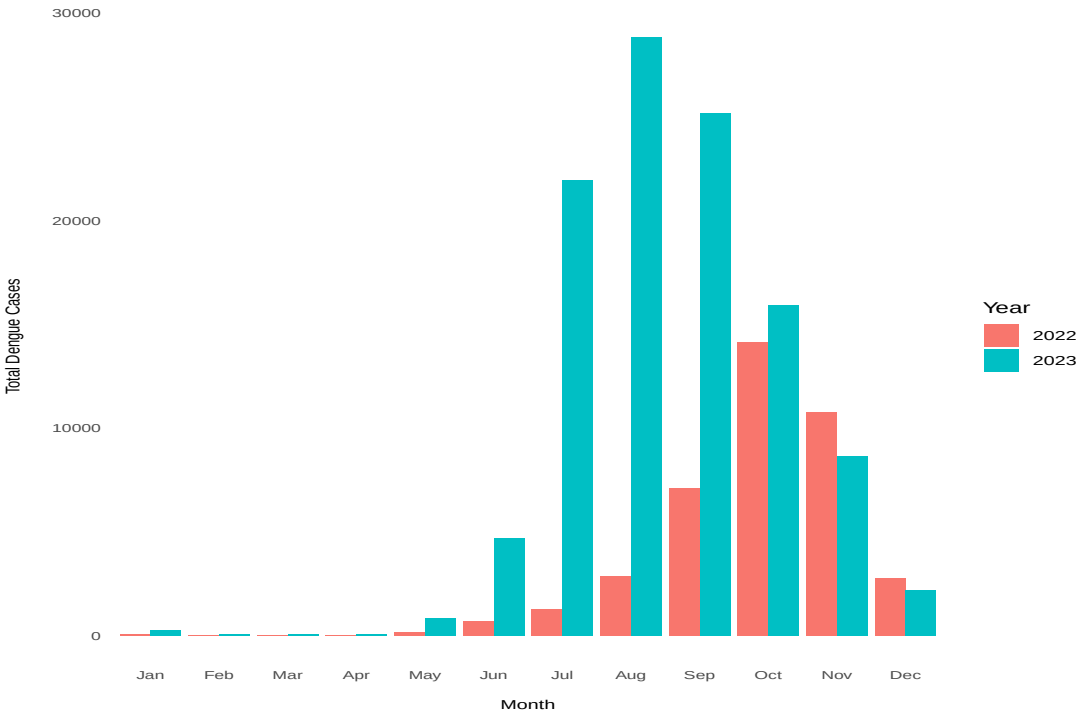


Figure 4.1: Month-wise dengue fever surveillance in Dhaka for 2022 and 2023

Table 4.2: Summary statistics of meteorological variables

Variable	Mean±SD	Maximum	Minimum
Summary statistics for 2022			
Temperature (°C)	31.7049 ± 3.5889	38.3	20.7
Humidity (%)	73.5945 ± 16.2590	100	40
Rainfall (mm)	3.5643 ± 15.9797	255	0
Visibility (km)	4.3084 ± 0.6331	5.4	0.4
Wind Speed (m/s)	2.6139 ± 1.2244	15	0
Summary statistics for 2023			
Temperature (°C)	32.0019 ± 4.1135	40.6	15.4
Humidity (%)	78.4575 ± 12.3031	100	40
Rainfall (mm)	5.9287 ± 16.7210	125	0
Visibility (km)	4.2635 ± 0.7261	6	1
Wind Speed (m/s)	2.3526 ± 1.0379	8	0

In 2022, temperatures ranged from a minimum of 20.7°C to a maximum of 38.3°C whereas in 2023, temperatures ranged from 15.4°C to 40.6°C. The temperatures in both years indicate a typical tropical climate, with occasional spikes in heat observed. Humidity levels in 2022 and 2023 varied between 40% and 100%. Higher average humidity suggests potentially more conducive conditions for mosquito activity, which is relevant for dengue transmission. Rainfall showed maximum variation in 2022, with amounts ranging from 0 mm to 255 mm. Conversely, 2023 experienced less varied rainfall, ranging from 0 mm to 125 mm. Visibility was relatively consistent across both years. Wind speeds also remained stable. These factors contribute to mosquito flight patterns and dispersal, influencing dengue transmission dynamics.

To identify potential non-linear relationships between the response variable and the predictors, we conducted an exploratory data analysis using scatter plots. Here the response was plotted against each predictor variable to visually inspect the relationship.

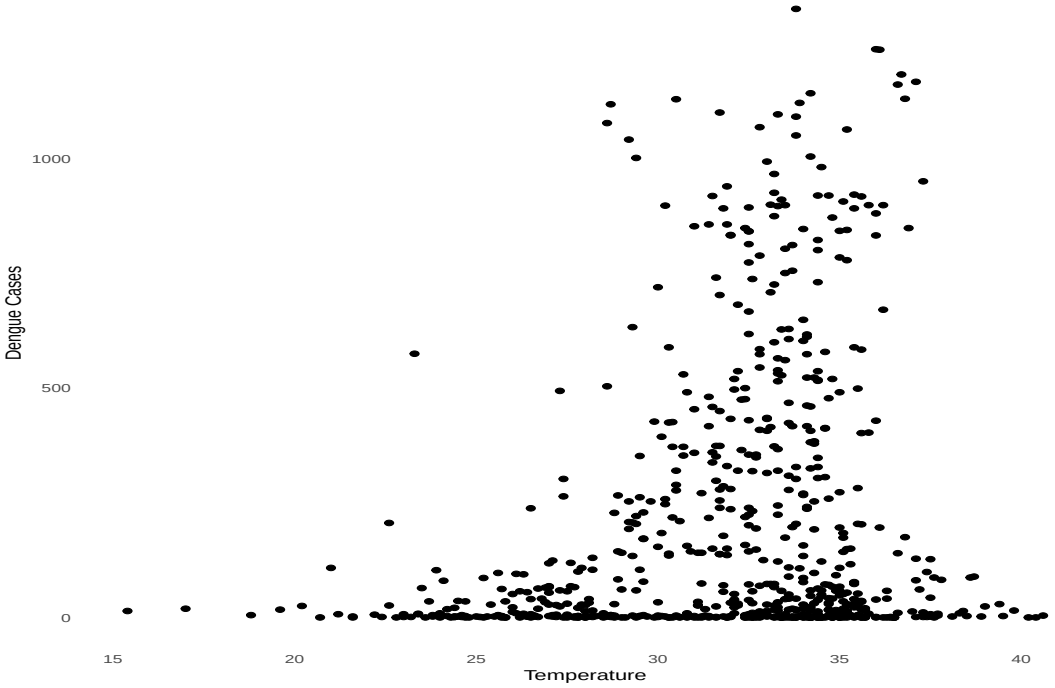


Figure 4.2: Scatter plot of temperature vs. dengue cases

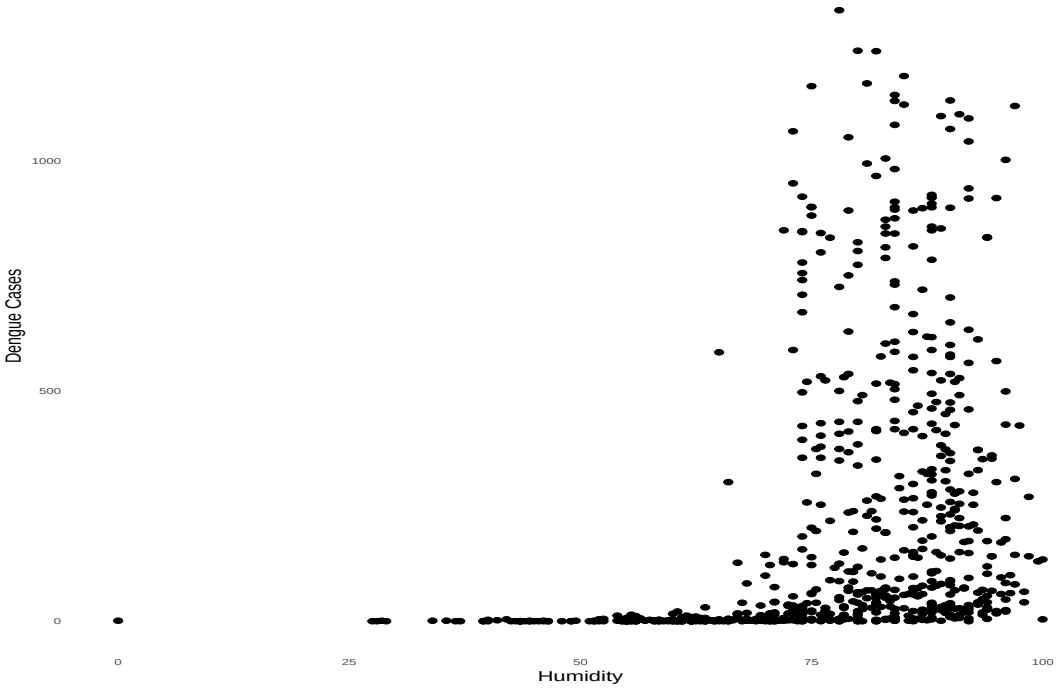


Figure 4.3: Scatter plot of humidity vs. dengue cases

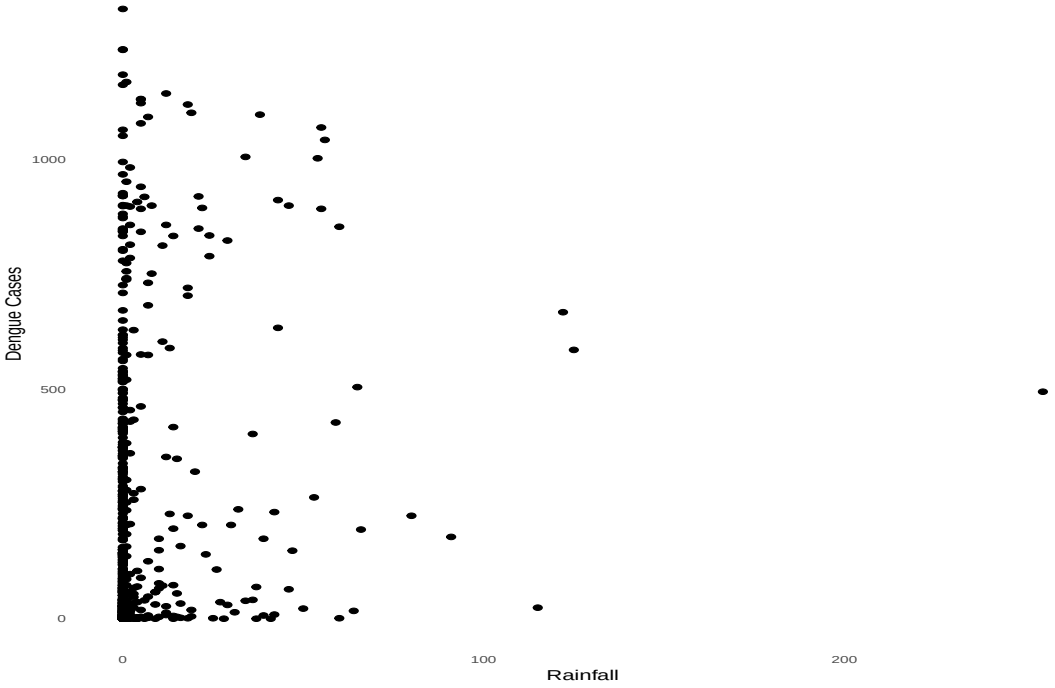


Figure 4.4: Scatter plot of rainfall vs. dengue cases

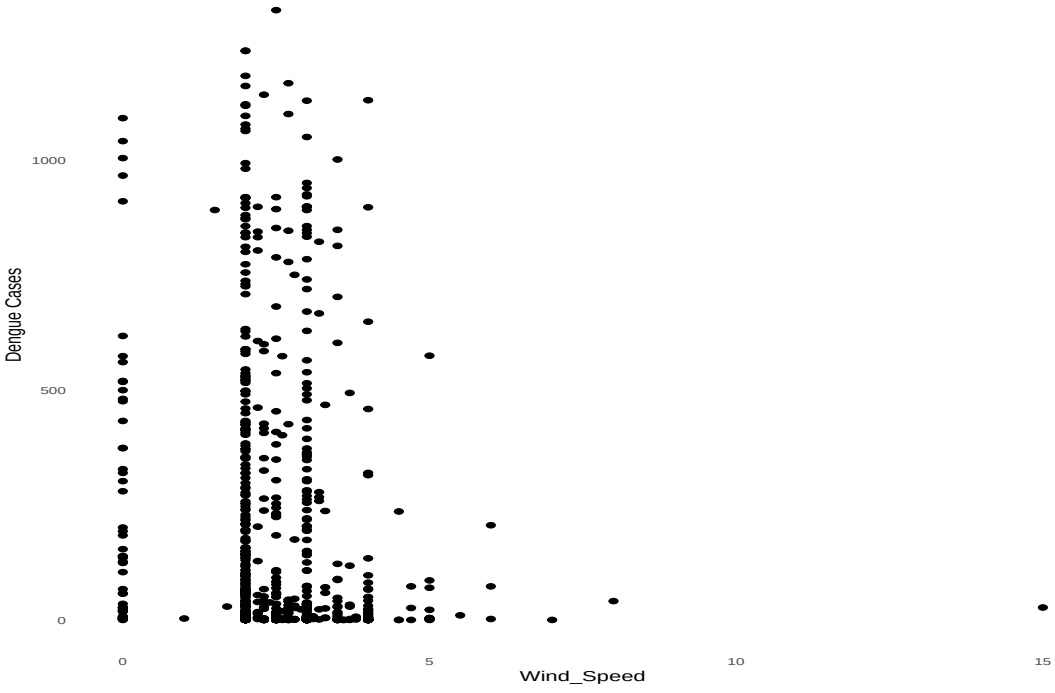


Figure 4.5: Scatter plot of wind speed vs. dengue cases

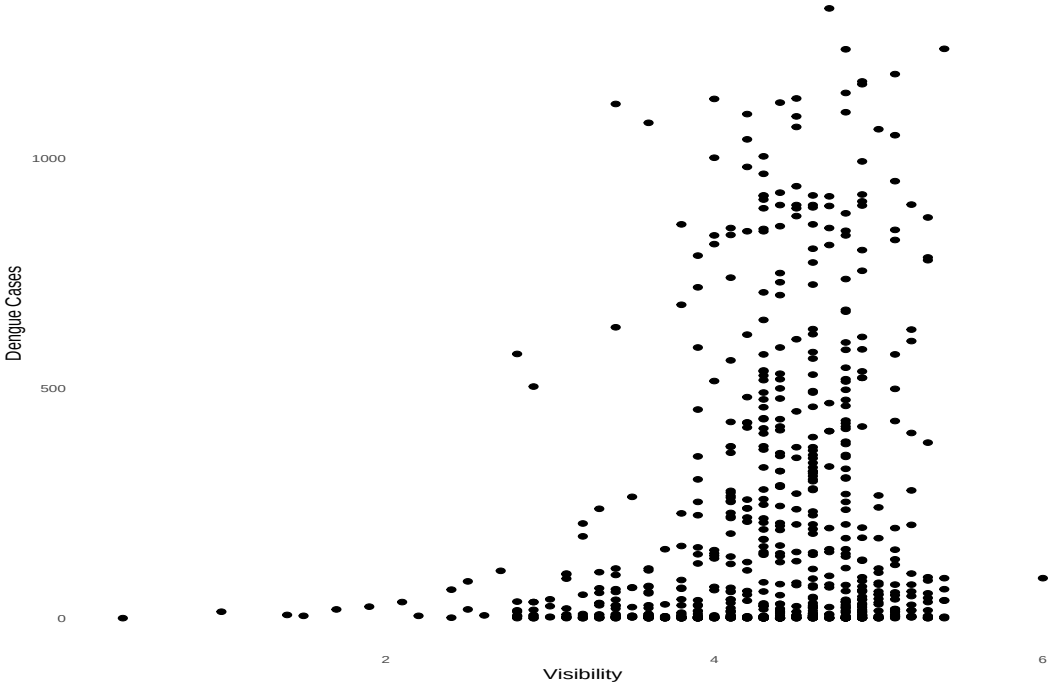


Figure 4.6: Scatter plot of visibility vs. dengue cases

The scatter plot of temperature versus dengue count (Figure 4.2) indicated a nonlinear relationship, where the number of dengue cases fluctuates at different temperature levels, suggesting that a simple linear model may not adequately capture this relationship. Similarly, the scatter plots of humidity, visibility, rainfall and wind speed (Figure 4.3 - 4.6) revealed nonlinear pattern, indicating that these variables have a complex effect on dengue incidence. These findings justify the use of a Generalized Additive Model (GAM) to better capture these complex relationships. GAMs allow each predictor to have its own smooth function, providing the flexibility needed to model the nonlinear effects observed in the data [29].

4.3 Generalized Additive Model with Autoregressive Terms: Analyzing Dengue Data

The Mann-Kendall test was performed to check if there exists any trend in the data. The value of the test was observed to be $\tau = 0.411$ with a corresponding p-value of < 0.001. A positive τ indicates an increasing trend, while a negative τ would indicate a decreasing trend. The test results indicate a significant up-

ward time trend. Incorporating the time trend directly into the model is often the most straightforward and effective approach, as it allows the model to account for both the trend and the relationships between response and explanatory variables simultaneously. We will start with a higher $df = 5$ to allow for more flexibility in modeling the upward trend.

In our study, we included lagged variables for temperature and humidity to account for their delayed impact on the response variable. We initially applied GAM, as described by the following equation:

For lagged temperature,

$$\ln(\mu_t) = \beta_0 + ns(\text{time}) + ns(\text{temperature}_{t-\text{lag1}}) + ns(\text{visibility}_t) + ns(\text{wind}_t) + ns(\text{rain}_t) + ns(\text{humidity}_t) + w_t(\text{week}_t)$$

For lagged humidity,

$$\ln(\mu_t) = \beta_0 + ns(\text{time}) + ns(\text{humidity}_{t-\text{lag1}}) + ns(\text{visibility}_t) + ns(\text{wind}_t) + ns(\text{rain}_t) + ns(\text{temperature}_t) + w_t(\text{week}_t)$$

First, lagged days for temperature terms and degrees of freedom (df) for all remaining natural spline functions were sequentially determined to minimize the Akaike Information Criterion (AIC). Subsequently, this initial set of parameters served as the starting point for a local optimization process aimed at further minimizing AIC [9]. The final selected parameters were as follows: lag1 = 6, dftime = 3, dftemperature lag1 = 10, dfvisibility = 9, dfwind = 2, dfrain = 1, dfhumidity = 5. For humidity, we employed a similar procedure to optimize the model parameters, resulting in the final selection of: lag1 = 5, dftime = 2, dfhumidity lag1 = 6, dfvisibility = 7, dfwind = 9, dftemperature = 15, dfrain = 1. The term w_t refers to the weekly effect, and $week_t$ denotes the day of the week that corresponds to date t. The indicator functions $I_i(week_t)$, for $i = 1, 2, 3, 4, 5, 6$, are defined as:

$$I_i(week_t) = \begin{cases} 1 & \text{if } week_t = i, \\ 0 & \text{if } week_t \neq i. \end{cases}$$

These functions indicate the position of the day within the week, ensuring w_t captures the weekly variation in the model.

For lagged temperature the ACF and PACF plots of the GAM Pearson residuals showed clear autocorrelation (Figure 4.7). Based on the autocorrelation plot, it was observed that the PACF values exceed the 95% confidence interval bounds (represented by the blue dashed line) [31] for lags less than 7. For lags larger than 6, the PACF values were contained within the bounds of the 95% CI. Therefore, based on this observation, the GAMAR with lag order 6 (GAMAR (6)) was selected as the appropriate model to fit the data. The model is given below:

$$\begin{aligned}
\ln(\mu_t) &= f(x_t) + \sum_{j=1}^{6} c_j \left(\ln\left(y^*_{t-j}\right) - f(x_{t-j})\right), \\
f(x_t) &= \beta_0 + ns(\text{time}) + ns(\text{temperature}_{t-6}) \\
&\quad + ns(\text{visibility}_t) + ns(\text{wind}_t) + ns(\text{rain}_t) \\
&\quad + ns(\text{humidity}_t) + w_t(\text{week}_t).
\end{aligned}$$

For lagged humidity, the ACF and PACF plots of the GAM Pearson residuals indicated clear autocorrelation (Figure 4.8). The PACF values exceeded the 95% confidence interval bounds (represented by the blue dashed line) [31] for lags less than 6, while for lags greater than 5, the PACF values remained within the 95% CI bounds. Consequently, a GAMAR model with a lag order of 5 (GAMAR (5)) was selected as the appropriate model to fit the data. The model is described below:

$$\begin{aligned}
\ln(\mu_t) &= f(x_t) + \sum_{j=1}^{5} c_j \left(\ln\left(y^*_{t-j}\right) - f(x_{t-j})\right), \\
f(x_t) &= \beta_0 + ns(\text{time}) + ns(\text{humidity}_{t-5}) \\
&\quad + ns(\text{visibility}_t) + ns(\text{wind}_t) + ns(\text{rain}_t) \\
&\quad + ns(\text{temperature}_t) + w_t(\text{week}_t).
\end{aligned}$$

All ACF and PACF values of Pearson residuals from the GAMAR (6) (Figure 4.7) for lagged temperature and GAMAR (5) (Figure 4.8) for lagged humidity are now below 0.1, indicating that the Pearson residuals from GAMAR approximated white noise.

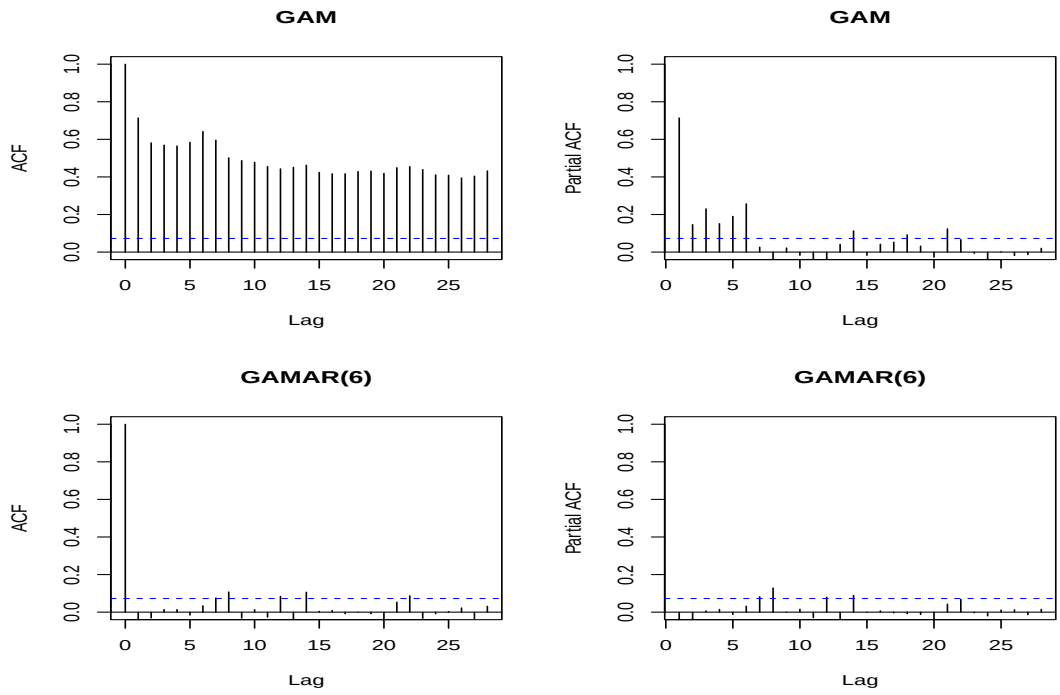


Figure 4.7: ACF and PACF for GAM and GAMAR (6) for temperature

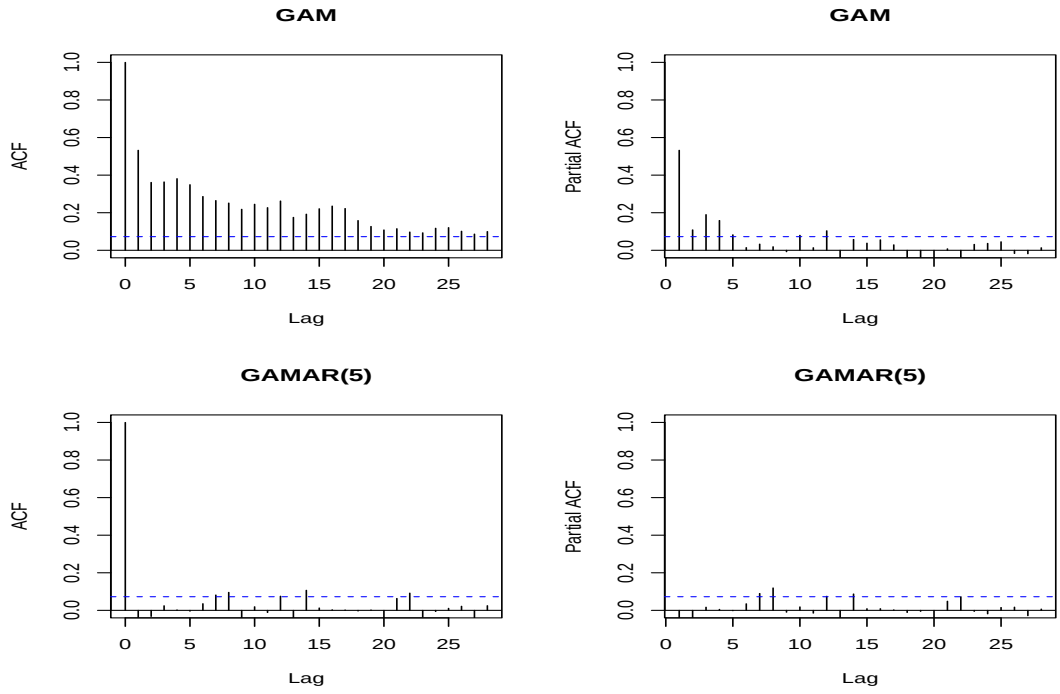


Figure 4.8: ACF and PACF for GAM and GAMAR (5) for humidity

Table 4.3: Estimates, standard error (SE) and p-value from GAM and GAMAR

	GAM		GAMAR	
Variable	Estimate (SE)	p-value	Estimate (SE)	p-value
For lagged temperature				
Constant	-1.6784 (0.2445)	< 0.001	-5.5713 (0.4647)	< 0.001
ns(time, 3)1	5.8093 (0.0386)	< 0.001	2.2210 (0.8193)	< 0.001
ns(time, 3)2	2.2237 (0.0099)	< 0.001	3.1937 (0.1385)	< 0.001
ns(time, 3)3	3.1152 (0.0125)	< 0.001	4.1268 (0.1458)	< 0.001
ns(temperature, 10)1	0.4187 (0.1074)	0.0277	0.0904 (0.1431)	< 0.001
ns(temperature, 10)2	0.6194 (0.1154)	< 0.001	0.2342 (0.1531)	< 0.001
ns(temperature, 10)3	1.2705 (0.1105)	< 0.001	0.2762 (0.1479)	< 0.001
ns(temperature, 10)4	0.8041 (0.1120)	< 0.001	0.2430 (0.1498)	< 0.001
ns(temperature, 10)5	1.4364 (0.1112)	< 0.001	0.2121 (0.1488)	< 0.001
ns(temperature, 10)6	1.1551 (0.1116)	< 0.001	0.1822 (0.1493)	< 0.001
ns(temperature, 10)7	0.8516 (0.1114)	< 0.001	0.3581 (0.1490)	< 0.001
ns(temperature, 10)8	3.1611 (0.0691)	< 0.001	0.3324 (0.0825)	< 0.001
ns(temperature, 10)9	-3.7859 (0.2425)	< 0.001	0.0152 (0.3192)	< 0.001
ns(temperature, 10)10	-6.2027 (0.1396)	< 0.001	-0.2534(0.1336)	< 0.001
ns(visibility, 9)1	2.3351 (0.2319)	< 0.001	0.0943 (0.1926)	< 0.001
ns(visibility, 9)2	2.9942(0.2428)	< 0.001	0.2149 (0.2013)	< 0.001
ns(visibility, 9)3	3.1494(0.2382)	< 0.001	0.3558 (0.1977)	< 0.001
ns(visibility, 9)4	2.9153 (0.2404)	< 0.001	0.0937 (0.1995)	< 0.001
ns(visibility, 9)5	3.1089 (0.2393)	< 0.001	0.2533 (0.1983)	< 0.001
ns(visibility, 9)6	3.2265 (0.2395)	< 0.001	0.2097 (0.1986)	< 0.001

Table continued from previous page

	GAM		GAMAR	
Variable	Estimate (SE)	p-value	Estimate (SE)	p-value
ns(visibility, 9)7	1.1579 (0.1269)	< 0.001	0.1893 (0.1080)	< 0.001
ns(visibility, 9)8	5.507 (0.4964)	< 0.001	0.6539 (0.4096)	< 0.001
ns(visibility, 9)9	2.5617 (1212)	< 0.001	0.8455 (0.1180)	< 0.001
ns(wind, 2)1	-3.3961 (0.0897)	< 0.001	0.0202 (0.0641)	< 0.001
ns(wind, 2)2	-6.3226 (0.2049)	< 0.001	0.1889 (0.1393)	< 0.001
ns(rain, 1)1	1.2992 (0.0348)	< 0.001	0.1889 (0.1393)	< 0.001
ns(humidity, 5)1	-2.7635 (0.6512)	0.0148	0.4977 (0.8998)	< 0.001
ns(humidity, 5)2	-2.8940 (0.6547)	0.0244	0.5755 (0.9047)	< 0.001
ns(humidity, 5)3	-1.5091 (0.4076)	0.0002	0.6781 (0.5590)	< 0.001
ns(humidity, 5)4	-5.8464 (1.2727)	0.0867	0.3774 (1.75680	< 0.001
ns(humidity, 5)5	-1.2688 (0.3045)	0.0208	0.3600 (0.4101)	< 0.001
as.factor(weekday)1	0.4917 (0.0108)	< 0.001	0.4215 (0.0100)	< 0.001
as.factor(weekday)2	0.4634 (0.0108)	< 0.001	0.4056 (0.0105)	< 0.001
as.factor(weekday)3	0.4383 (0.0108)	< 0.001	0.4306 (0.0105)	< 0.001
as.factor(weekday)4	0.4435 (0.0109)	< 0.001	0.3810 (0.0110)	< 0.001
as.factor(weekday)5	0.3897 (0.0107)	< 0.001	0.4182 (0.0106)	< 0.001
as.factor(weekday)6	0.4955 (0.0107)	< 0.001	4.8380 (0.0101)	< 0.001
AR1			0.2370 (0.0074)	< 0.001
AR2			0.1678 (0.0072)	< 0.001
AR3			0.1433 (0.0064)	< 0.001
AR4			0.0810 (0.0064)	< 0.001

Table continued from previous page

	GAM		GAMAR	
Variable	Estimate (SE)	p-value	Estimate (SE)	p-value
AR5			0.0780 (0.0060)	< 0.001
AR6			0.0894 (0.0067)	< 0.001
Log-likelihood	**-65420.28**		**-8238.81**	
AIC	**130900.6**		**16553.62**	
Deviance	**127137.5**		**12799.96**	
For lagged humidity				
Constant	-6.1038 (0.4189)	< 0.001	-4.1880 (0.5658)	< 0.001
ns(time, 2)1	5.6113 (0.0493)	< 0.001	20.0693 (0.7214)	< 0.001
ns(time, 2)2	1.0811 (0.0103)	< 0.001	2.7328 (0.1140)	< 0.001
ns(humidity, 6)1	4.2544 (0.3489)	< 0.001	-0.2546 (0.3999)	< 0.001
ns(humidity, 6)2	3.7822 (0.3505)	< 0.001	-0.2978 (0.4012)	< 0.001
ns(humidity, 6)3	4.0355 (0.3497)	< 0.001	-0.3891 (0.4006)	< 0.001
ns(humidity, 6)4	6.1574 (0.2220)	< 0.001	-0.0498 (0.2563)	< 0.001
ns(humidity, 6)5	-1.0073 (0.6869)	< 0.001	-1.9275 (0.7808)	< 0.001
ns(humidity, 6)6	6.6315 (0.1660)	< 0.001	-0.1398 (0.1924)	< 0.001
ns(visibility, 7)1	2.3394 (0.2269)	< 0.001	-0.0092 (0.2121)	< 0.001
ns(visibility, 7)2	2.6171 (0.2342)	< 0.001	0.1891 (0.2186)	< 0.001
ns(visibility, 7)3	2.2527 (0.2308)	< 0.001	-0.0023 (0.2158)	< 0.001
ns(visibility, 7)4	2.7067 (0.2323)	< 0.001	0.1012 (0.2171)	< 0.001
ns(visibility, 7)5	0.9847 (0.1249)	< 0.001	0.0062 (0.1175)	< 0.001
ns(visibility, 7)6	4.8664 (0.4739)	< 0.001	0.3796 (0.4450)	< 0.001

Table continued from previous page

	GAM		GAMAR	
Variable	Estimate (SE)	p-value	Estimate (SE)	p-value
ns(visibility, 7)7	2.8607 (0.1210)	< 0.001	0.5229 (0.1213)	< 0.001
ns(wind, 9)1	-2.9130 (0.4351)	< 0.001	1.4153 (0.5182)	< 0.001
ns(wind, 9)2	0.4167 (0.0102)	< 0.001	0.0237 (0.0099)	< 0.001
ns(wind, 9)3	0.3997 (0.0267)	< 0.001	0.0865 (0.02611)	< 0.001
ns(wind, 9)4	-0.1859 (0.0290)	< 0.001	0.0123 (0.0289)	< 0.001
ns(wind, 9)5	0.1260 (0.0175)	< 0.001	0.1598 (0.0166)	< 0.001
ns(wind, 9)6	-0.1608 (0.0184)	< 0.001	-0.1724 (0.0171)	< 0.001
ns(wind, 9)7	-2.0977 (0.1965)	< 0.001	1.6920 (0.2198)	< 0.001
ns(wind, 9)8	1.5365 (0.4062)	< 0.001	-0.6998 (0.4294)	< 0.001
ns(wind, 9)9	0.1530 (0.4932)	< 0.001	0.3323 (0.2852)	< 0.001
ns(temperature, 15)1	2.0938 (0.1962)	< 0.001	-0.0005 (0.1755)	< 0.001
ns(temperature, 15)2	2.0938 (0.2106)	< 0.001	0.3141 (.1858)	< 0.001
ns(temperature, 15)3	1.9185 (0.2024)	< 0.001	0.4050 (0.1778)	< 0.001
ns(temperature, 15)4	2.2522 (0.2064)	< 0.001	0.2871 (0.1812)	< 0.001
ns(temperature, 15)5	2.3998 (0.2039)	< 0.001	0.3433 (0.1791)	< 0.001
ns(temperature, 15)6	2.0266 (0.2044)	< 0.001	0.2426 (0.1797)	< 0.001
ns(temperature, 15)7	2.5007 (0.2045)	< 0.001	0.3611 (0.1996)	< 0.001
ns(temperature, 15)8	2.4805 (0.2043)	< 0.001	0.3594 (0.1997)	< 0.001
ns(temperature, 15)9	2.3934 (0.2048)	< 0.001	0.4229 (0.1804)	< 0.001
ns(temperature, 15)10	2.3999 (0.2044)	< 0.001	0.2449 (0.1798)	< 0.001
ns(temperature, 15)11	1.8615 (0.2047)	< 0.001	0.3451 (0.1799)	< 0.001

Table continued from previous page

	GAM		GAMAR	
Variable	Estimate (SE)	p-value	Estimate (SE)	p-value
ns(temperature, 15)12	1.9540 (0.2045)	< 0.001	0.4346 (0.1798)	< 0.001
ns(temperature, 15)13	2.9754 (0.1198)	< 0.001	0.1734 (0.1068)	< 0.001
ns(temperature, 15)14	-1.2416 (0.4438)	0.0051	0.4919 (0.3766)	< 0.001
ns(temperature, 15)15	-896739 (0.2525)	< 0.001	0.5085 (0.1647)	< 0.001
ns(rain, 1)1	1.9220 (0.04108)	< 0.001	0.2802 (0.0419)	< 0.001
as.factor(weekday)1	0.3325 (0.0110)	< 0.001	0.4340 (0.0099)	< 0.001
as.factor(weekday)2	0.3988 (0.0110)	< 0.001	0.4437 (0.0098)	< 0.001
as.factor(weekday)3	0.4112 (0.0109)	< 0.001	0.4407 (0.0096)	< 0.001
as.factor(weekday)4	0.4954 (0.0112)	< 0.001	0.4547 (0.0102)	< 0.001
as.factor(weekday)5	0.4279 (0.0110)	< 0.001	0.4566 (0.0098)	< 0.001
as.factor(weekday)6	0.4174 (0.0111)	< 0.001	0.4335 (0.0103)	< 0.001
AR1			0.2507 (0.0077)	< 0.001
AR2			0.2115 (0.0072)	< 0.001
AR3			0.1878 (0.0068)	< 0.001
AR4			00.1264 (0.0066)	< 0.001
AR5			0.1009 (0.0061)	< 0.001
Log-likelihood	**-47542.52**		**-8375.66**	
AIC	**95179.03**		**16855.34**	
Deviance	**91381.97**		**13063.18**	

To further evaluate the models, we compared the log-likelihood, AIC, and deviance values. The log-likelihood measures the goodness of fit of a model. In this case, the GAMAR model had a much higher log-likelihood (-8238.81) for lagged temperature and (-8375.66) for lagged humidity compared to the GAM model (-65420.28) and (-47542.52) respectively, suggesting that GAMAR fits the data better than GAM in both cases. The AIC is a measure used for model comparison, where lower values indicate a better model, balancing goodness of fitting and model complexity [32]. Here, the GAMAR model had a much lower AIC (16553.62 for lagged temperature and 16855 for lagged humidity) compared to the GAM model (130900.6 for lagged temperature and 95179 for lagged humidity), indicating that GAMAR is preferred over GAM when considering both fit and complexity. Additionally, the GAMAR model had a significantly lower deviance (12799.96 for lagged temperature and 13063 for lagged humidity) compared to the GAM model (127137.5 for lagged temperature and 91381 for lagged humidity), further suggesting that the GAMAR model provides a better fit to the data. We have calculated the dispersion parameter ϕ for both lagged temperature and lagged humidity, and the value being close to 1 suggests that the problem of overdispersion has been addressed.

The coefficients linked with natural splines describe the impact of the smooth function on dengue incidence. Significance of these splines indicates that their nonlinear relationships effectively explain the variability in dengue cases. Here, the coefficients cannot be interpreted in the same manner as in linear regression.

Table 4.3 showed that the coefficients of all the variable's natural spline associated with lagged temperature for both GAM and GAMAR models were significant, indicating their effective explanation of the variability in dengue cases. Similarly, the coefficients of all the variable's natural spline linked to lagged humidity were also significant, further suggesting their crucial role in explaining the variability in dengue cases.

To ensure the accuracy of the estimated relationship between temperature and

dengue cases, as well as humidity and dengue cases, we compared the results from our model with direct observational plots. Specifically, we plotted the mean dengue cases against temperature (Figure 4.9) and compared this with the effect of temperature derived from our model. Similarly, we plotted the mean dengue cases against humidity (Figure 4.11) and compared this with the effect of humidity derived from our model. These comparisons help to verify that the model correctly captures the underlying trends observed in the raw data. The effects of the predictor variables, as plotted, matched the patterns seen in the observational plot. Based on the simulation study from the previous chapter, we found that GAMAR outperforms GAM. Therefore, we will focus on explaining the GAMAR here.

Temperature

From the ACF and PACF plot we have concluded that GAMAR (6) was appropriate for modeling the data, indicating that the average temperature and dengue cases over the six days preceding a given dengue case can better predict it than the temperature on the day itself. On reflection such a model might be more sensible on biological and medical grounds: the temperature recorded in the data are not high enough to cause immediate acute disease and it seems more plausible that any effects would take some time to manifest themselves via for example the aggravation of existing medical conditions [29].

We have plotted the effects of temperature, considering the effects of other variables (visibility, rainfall, wind speed and humidity) as potential confounders. The partial effect plot for dengue cases vs lagged temperature (Figure 4.10) using GAMAR revealed a complex, non-linear relationship. We had a log link function so the partial effects are not the same as what we would expect on the actual scale of uptake. In Figure 4.10, temperature1 refers to the lagged temperature.

Moderate temperatures (28°C to 36°C) were associated with higher dengue case counts, whereas both lower and higher temperatures correlated with reduced transmission rates.

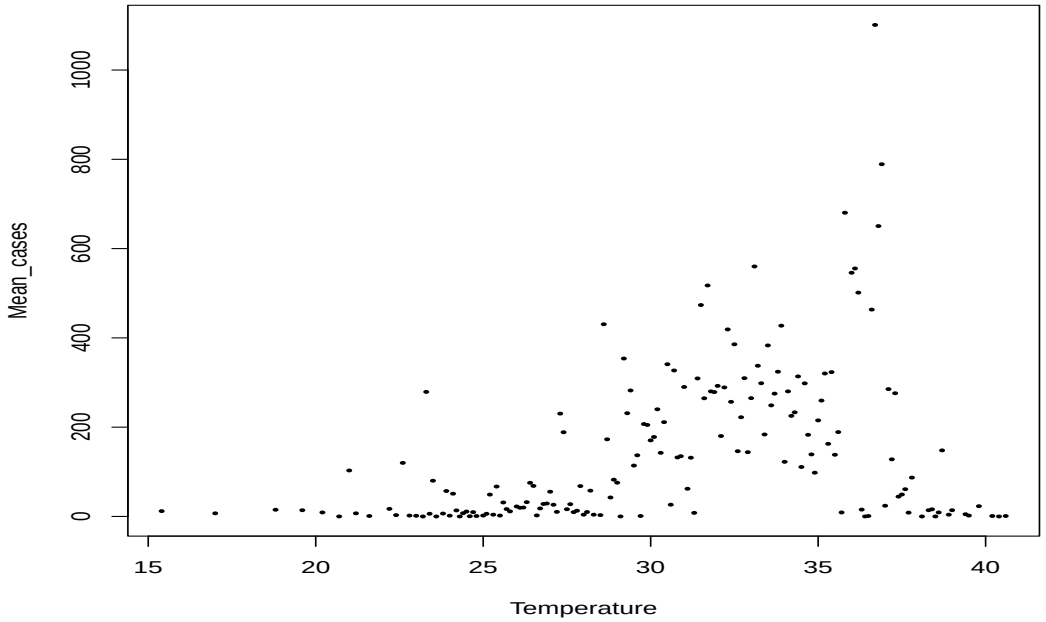
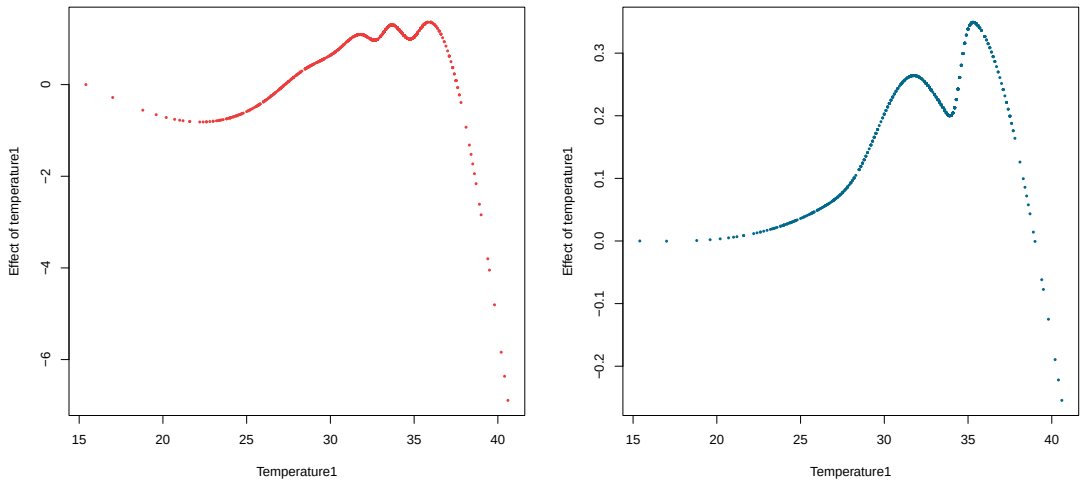


Figure 4.9: Scatter plot of temperature vs. mean dengue cases



(a) Effect of temperature from GAM (b) Effect of $temperature_{t-6}$ from GAMAR (6)

Figure 4.10: Effect of temperature from GAM and GAMAR (6)

More specifically, up to 25°C the effect of temperature on dengue cases remained relatively stable, showing an almost horizontal line. This indicates that within this temperature range, there was minimal change in the number of dengue cases. The stability might be due to temperatures being within an optimal range for mosquito survival but not significantly enhancing virus transmission or mosquito activity. As temperature increased from 25°C to 32°C, there was a gradual rise in the number of dengue cases. This suggests that slightly warmer temperatures in this range may enhance mosquito activity and virus replication rates, thereby increasing the risk of dengue transmission. After peaking at around 32°C, the number of dengue cases began to decrease slightly. This could indicate that while temperatures are still conducive for mosquito activity, other factors such as increased mortality rates at higher temperatures might begin to offset the benefits. The relationship showed another peak in dengue cases at around 36°C. This secondary peak suggests that extremely high temperatures can again promote conditions that favor dengue transmission, potentially through accelerated virus replication rates within mosquitoes or increased mosquito biting rates. After reaching the highest point at 36°C, there was a rapid decline in the number of dengue cases. This rapid decrease indicates that temperatures above this threshold likely exceed the optimal range for mosquito survival and virus replication. High temperatures may lead to increased mosquito mortality, thus reducing the population and subsequent dengue transmission. Both lower and higher extremes of temperature outside this range appear less favorable for dengue transmission, likely due to the impacts on mosquito physiology and virus replication dynamics.

Humidity

From the ACF and PACF plot we have concluded that GAMAR (5) was appropriate for modeling the data, indicating that the humidity and dengue cases in five days preceding a given dengue infected case can better predict it than the humidity on the day itself. On reflection such a model might be more sensible on medical grounds: the humidity recorded in the data are not high enough to cause immediate acute disease and it seems more plausible that any effects would take some time to manifest themselves via existing medical conditions [29].

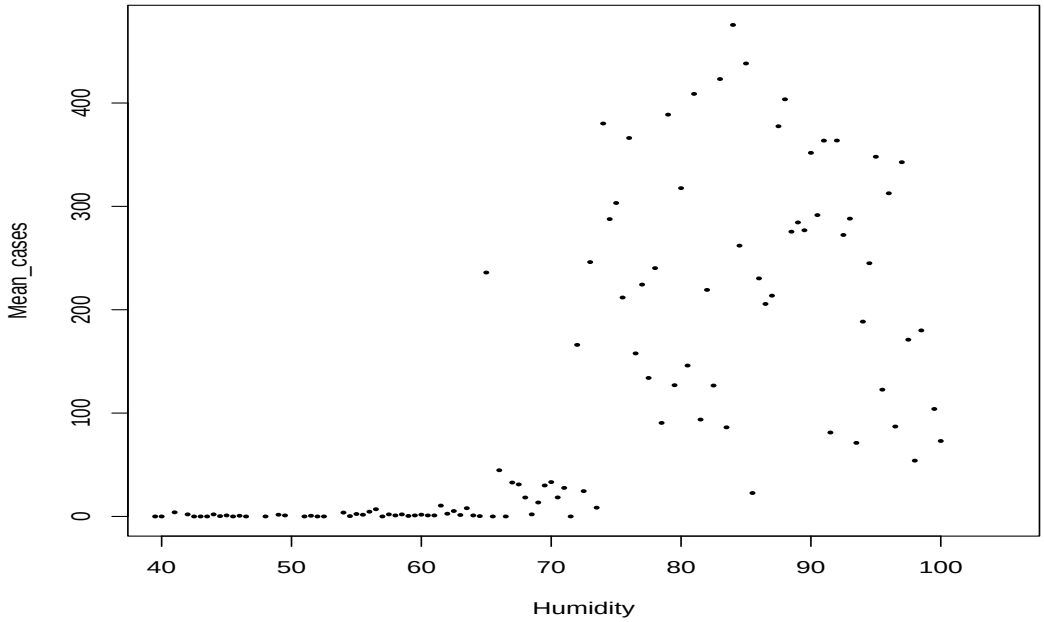
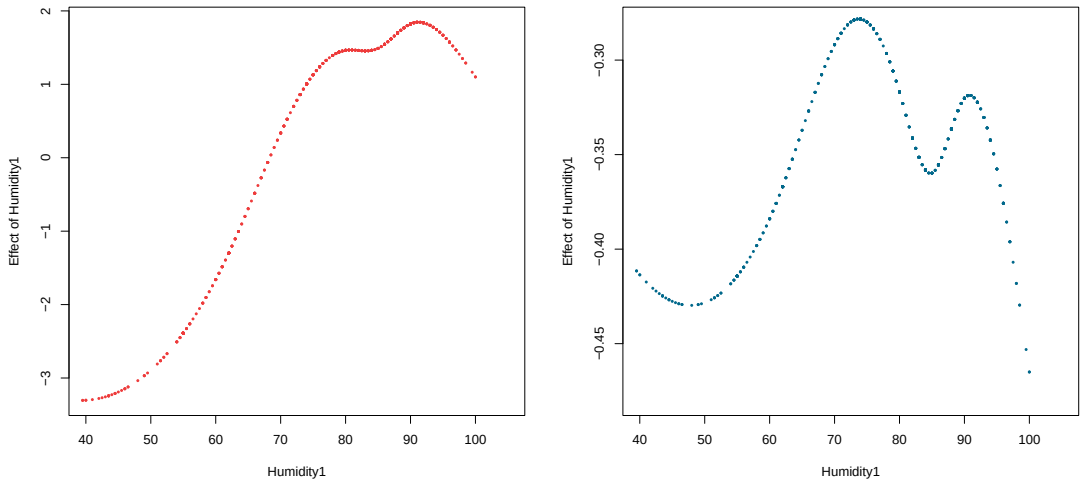


Figure 4.11: Scatter plot of humidity vs. mean dengue cases



(a) Effect of humidity from GAM (b) Effect of $humidity_{t-5}$ from GAMAR (5)

Figure 4.12: Effect of humidity from GAM and GAMAR (5)

We have plotted the effects of humidity, considering the effects of other variables (visibility, rainfall, wind speed and temperature) as potential confounders. The partial effect plot for dengue cases vs lagged humidity (Figure 4.12) using GAMAR revealed a complex, non-linear relationship. We have a log link function so the partial effects are not the same as what we would expect on the actual scale of uptake. In Figure 4.12, humidity1 refers specifically to the lagged humidity.

The results indicate that dengue transmission is highly sensitive to changes in humidity. Moderate humidity levels (50% to 75%) were associated with higher dengue case counts, whereas both lower (<50%) and higher (>90%) humidity levels correlated with reduced transmission rates. This complex pattern suggests that there are optimal humidity conditions for the survival and breeding of the mosquito vectors responsible for spreading dengue.

More specifically as humidity increases from 40% to 50%, there was a noticeable decrease in the number of dengue cases. This suggests that lower humidity levels might not be conducive to the transmission of dengue. From 50% to 75% humidity, there was a marked increase in the number of dengue cases. This indicates that moderate humidity levels create favorable conditions for the proliferation of dengue. The number of dengue cases peaked at around 75% humidity, reaching the highest levels observed in the study. However, as humidity continued to increase towards 85%, there was a subsequent decline in dengue cases. Between 85% and 92% humidity, dengue cases initially decreased but show a slight increase around 82% humidity. This fluctuation suggests that other factors may be influencing dengue transmission in this range. After 92% humidity, there was a rapid decrease in the number of dengue cases. Extremely high humidity levels appear to significantly inhibit the conditions necessary for dengue transmission.

Chapter 5

Conclusion

Lei et al. proposed the GAM model with autoregressive terms (GAMAR) by incorporating the autoregressive correlation structure of both the response and explanatory variables. They conducted simulation studies to compare the performance of GAM and GAMAR under two different setups. In the first setup, responses were generated by considering a predefined set of coefficients alongside autoregressive terms, while the second setup investigated whether their suggested approach could approximate a nonlinear curve. They considered one specific lag when simulating data from the AR process in the first setup and one specific functional form in the second setup to demonstrate the efficiency of their proposed model. However, different lag values and different functional forms were not explored in their simulation study.

Motivated by these gaps, this thesis conducted an extensive simulation study covering a range of lag values and functional forms. The efficiency of the GAMAR model was verified by comparing bias, relative error, and coverage. Our findings indicate that GAMAR consistently outperforms GAM, regardless of the lag values and functional forms considered. As sample size and AR order increase, the advantages of GAMAR become even more pronounced.

Our scatter plot analysis of dengue cases against various weather variables has revealed a non-linear relationship, as shown in Chapter 4. To model these non-linear relationships, GAMs are widely used due to their flexibility. The ACF and

PACF plots of the GAM for both temperature and humidity have indicated dependency on their past values, suggesting that GAMAR is appropriate for modeling the data. These findings are also consistent with the results obtained from the AIC value which showed that GAMAR has lower AIC value compared to GAM in both lagged temperature and lagged humidity case. We identified that dengue infected for a specific day depend on the past 6 days temperatures and dengue incidence. Similarly, dengue infected for a specific day depend on the past 5 days humidity and dengue incidence. The partial effect plot showed that both temperature and humidity exhibit complex non-linear relationships with dengue cases.

Rocommendations

Based on the findings from the analysis, the most vulnerable periods for dengue transmission in relation to temperature and humidity can be identified and used to guide public health interventions. Here are the specific recommendations:

- 28°C to 36°C: This temperature range is associated with higher dengue case counts, with a peak around 32°C and another peak at 36°C. When temperatures often fall within this range, public health efforts should be strengthened.

- 50% to 75% Humidity: This range is associated with a marked increase in dengue cases, peaking around 75%. Interventions should be intensified when humidity levels are within this range.

For future work, one may consider using the GAMAR as an alternative to classical count models for other datasets. Additionally, simulations involving more than one covariates can be explored.

Bibliography

[1] Scott L Zeger. "A regression model for time series of counts". In: *Biometrika* 75.4 (1988), pp. 621–629.

[2] Richard A Davis, WTM Dunsmuir, and Ying Wang. "Modelling time series of counts". In: *Asymptotics, Nonparametrics and Time Series, A Tribute to Madan-Puri* (1999), pp. 63–113.

[3] Roman Liesenfeld, Ingmar Nolte, and Winfried Pohlmeier. "Modelling financial transaction price movements: a dynamic integer count data model". In: *Empirical Economics* 30 (2006), pp. 795–825.

[4] Tina Hviid Rydberg and Neil Shephard. "Dynamics of trade-by-trade price movements: decomposition and models". In: *Journal of Financial Econometrics* 1.1 (2003), pp. 2–25.

[5] Peter McCullagh. *Generalized linear models*. Routledge, 2019.

[6] John R Leathwick, Jane Elith, and Trevor Hastie. "Comparative performance of generalized additive models and multivariate adaptive regression splines for statistical modelling of species distributions". In: *Ecological modelling* 199.2 (2006), pp. 188–196.

[7] David N Barron. "The analysis of count data: Overdispersion and autocorrelation". In: *Sociological methodology* (1992), pp. 179–220.

[8] Michael H Kutner et al. *Applied linear statistical models*. McGraw-hill, 2005.

[9] Lei Yang et al. "Using a generalized additive model with autoregressive terms to study the effects of daily temperature on mortality". In: *BMC medical research methodology* 12 (2012), pp. 1–13.

[10] Michael A Benjamin, Robert A Rigby, and D Mikis Stasinopoulos. "Generalized autoregressive moving average models". In: *Journal of the American Statistical association* 98.461 (2003), pp. 214–223.

[11] Maria G Guzman and Eva Harris. "Dengue". In: *The Lancet* 385.9966 (2015), pp. 453–465.

[12] Sorif Hossain et al. "Association of climate factors with dengue incidence in Bangladesh, Dhaka City: a count regression approach". In: *Heliyon* 9.5 (2023).

[13] Md Nazmul Karim et al. "Climatic factors influencing dengue cases in Dhaka city: a model for dengue prediction". In: *Indian journal of medical research* 136.1 (2012), pp. 32–39.

[14] Md Sahidul Islam, Masato Kimura, and Hisanori Miyata. "Generalized Moment Method for Smoluchowski Coagulation Equation and Mass Conservation Property". In: *Mathematics* 11.12 (2023), p. 2770.

[15] Ben Kirtman et al. "Near-term climate change: projections and predictability". In: (2013).

[16] Simon Hales et al. "Potential effect of population and climate changes on global distribution of dengue fever: an empirical model". In: *The Lancet* 360.9336 (2002), pp. 830–834.

[17] Duane J Gubler. "Resurgent vector-borne diseases as a global health problem." In: *Emerging infectious diseases* 4.3 (1998), p. 442.

[18] Md Aminul Islam et al. "Correlation of dengue and meteorological factors in Bangladesh: a public health concern". In: *International Journal of Environmental Research and Public Health* 20.6 (2023), p. 5152.

[19] Sabrina Islam et al. "Climate variability, dengue vector abundance and dengue fever cases in Dhaka, Bangladesh: a time-series study". In: *Atmosphere* 12.7 (2021), p. 905.

[20] J. A. Nelder and R. W. M. Wedderburn. "Generalized Linear Models". In: *Journal of the Royal Statistical Society. Series A (General)* 135.3 (1972), pp. 370–384. ISSN: 00359238. URL: http://www.jstor.org/stable/2344614 (visited on 06/29/2024).

[21] Trevor J Hastie and Daryl Pregibon. "Generalized linear models". In: *Statistical models in S*. Routledge, 2017, pp. 195–247.

[22] Kim Larsen. "GAM: the predictive modeling silver bullet". In: *Multithreaded. Stitch Fix* 30 (2015), pp. 1–27.

[23] Boor de. "A practical guide to splines". In: (1978).

[24] C Wan and W Zhong. "MultiKink: Estimation and Inference for Multi-Kink Quantile Regression". In: *R package version 0.1. 0* (2020).

[25] Armando Teixeira-Pinto. "Armando Teixeira-Pinto". In: (2023).

[26] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[27] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[28] Frank E Harrell et al. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Vol. 608. Springer, 2001.

[29] Simon N Wood. *Generalized additive models: an introduction with R*. chapman and hall/CRC, 2017.

[30] Joao Henrique F Flores, Paulo Martins Engel, and Rafael C Pinto. "Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting". In: *The 2012 International joint conference on neural networks (IJCNN)*. IEEE. 2012, pp. 1–8.

[31] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time Series Analysis: Forecasting and Control*. 3rd. Englewood Cliffs, NJ: Prentice Hall, 1994.

[32] Hirotugu Akaike. "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

Appendix

Selected R Codes Used in the Thesis

```
### Data Generation: Scenario 1 (Case 1) ###


simuns<-function(mx=1000,bb,pp,temp,seed=2024){
truevalue<-c(bb,pp)
b<-NROW(bb)-1
p<-NROW(pp)
x<-temp
X<-ns(x,b)
N<-NROW(x)
n<-u<-yy<-y<-a<-numeric(N)
cc<-0.5
X0<-cbind(rep(1,N),X)    # X0<-(aa,X) is the design matrix
no<-X0%*%bb
Y0<-matrix(0,mx,N)       # Y is stored in Y0
set.seed(seed)
for(ii in 1:mx)          # simulate y of number mx
{
## Since we are considering a lag of 1, the count of cases on a
given day will depend on the count of cases on the previous day,
so there will be no autocorrelation coefficient for the first day.
So we are generating a random number for the first day. ##
for(t in 1:p)
a[t]<-rnorm(1,0,0.2)
```

```
mm<-numeric(N)
for(t in (1+p):N)
{
for(i in 1:p)
mm[t]<-mm[t]+pp[i]*a[t-i];
n[t]<-X0[t,]%*%bb+mm[t]
u[t]<-exp(n[t])                 # Poisson distribution mean
y[t]<-rpois(1,u[t])
yy[t]<-max(y[t],cc)             # cc is threshold parameter
a[t]<-log(yy[t])-X0[t,]%*%bb    # autoregressive term
}
Y0[ii,]<-y
if((ii%%10)==0)
cat(ii,"\n")
}
simu<-list()
simu$Y0<-Y0
simu$no<-no
simu
}

## Required Package ##
library(foreign)
library(splines)
library(akima)
library(foreach)
library(mgcv)

simu<-simuns(mx=1000,bb=c(5.02,-0.45,-0.46,-0.48,-0.43,-0.38,-0.25),
pp=0.5,temp=temp,seed=2024)      # c2=0.25, c3=0.12, c4=0.06
```

```
### Data Generation: Scenario 2 (Case 1) ###

#### We have considered the first model ####

simucos<-function(mx=1000,pp,temp,seed=2024){
p<-NROW(pp)
x<-temp
no<-3.5+0.4*cos(20*pi*(x+5)/100)    # non-linear function considered
                                     in the first model
N<-NROW(x)
n<-u<-yy<-y<-a<-numeric(N)
cc<-0.5
Y0<-matrix(0,mx,N)                  # Y is stored in Y0
set.seed(seed)
for(ii in 1:mx)                     # simulate y of number mx
{for(t in 1:p)
a[t]<-rnorm(1,0,0.2)
mm<-numeric(N);
for(t in (1+p):N)
{for(i in 1:p)
mm[t]<-mm[t]+pp[i]*a[t-i];
#generate n,u,y,yy
n[t]<-3.5+0.4*cos(20*pi*(x+5)/100)+mm[t];
u[t]<-exp(n[t]);                    # Poisson distribution mean
y[t]<-rpois(1,u[t]);
yy[t]<-max(y[t],cc);
a[t]<-log(yy[t])-(3.5+0.4*cos(20*pi*(x+5)/100)); # autoregressive term
}
Y0[ii,]<-y
if((ii%%10)==0)
cat(ii,"\n")
}
```

```
simu<-list()
simu$Y0<-Y0
simu$no<-no
simu
}

# 4.8+0.2*sin(pi*(x[t]+8)/33) used as second model

simu<-simucos(mx=1000,pp=0.5,temp=temp,seed=2024)

######## Simulation Data Analysis ########

### Generalized Additive Model with Autoregressive Terms ###

 gamAR<-function (formula, data, p.ar = 1, starts = starts, w = rep(1,NN),
        family = "poisson", cc = 0.5, de = 0.01, control=list(...),...)
{
if(family!="poisson")
stop("sorry, only poisson family is currently implemented")
family=poisson()
control<-do.call("glm.control", control)
times<-control$maxit
epsilon<-control$epsilon
if (missing(data))
data<-environment(formula)

mf<-match.call(expand.dots = FALSE)
m<-match(c("formula", "data"), names(mf), 0L)
mf<-mf[c(1L, m)]
mf$drop.unused.levels<-TRUE
mf[[1L]]<-as.name("model.frame")
mf<-eval(mf, parent.frame())
```

```
term.labels<-attr(attributes(mf)$terms,"term.labels")
mt<-attr(mf, "terms")
y<-model.response(mf, "any")
X< model.matrix(mt, mf, contrasts)
xnames<-colnames(X)
ynames<-names(y)
b<-ncol(X)-1
p<-p.ar
NN<-NROW(y)
pp<-rep(0,p)
r<-matrix(0,times+1,1+b+p)
r[1,]<-c(starts,pp)

#aa is part of design matrix for AR resar pearson residuals
aa<-matrix(0,NN-p,p)
#Q is partial derivative of eta (linear predictor) on every parameter
Q<-matrix(0,NN-p,b+p+1);
QQ<-matrix(0,NN-p,b+1)
#transpose of X
tX<-t(X)
#all are variables used in the calculation
a<-mm<-n<-u<-numeric(NN);
#yy=max(cc,y) avoid that y<0
yy<-y
yy[y<cc]<-cc
pan <- 0

k=1
while(k<(times+1)&pan==0)
{
if(k==1)
#while in the first iteration
```

```
#give the initial parameters.
{
bb<-r[k,c(1:(b+1))];
pp<-r[k,c((b+2):(b+p+1))];
#calculate a, the intermediate variabel a
a<-log(yy)-X%*%bb;
for(i in 1:p)
aa[,p-i+1]<-a[i:(NN-p+i-1)]
n[(p+1):NN]<-(X%*%bb)[(p+1):NN]+aa%*%pp+log(w)[(p+1):NN]
u<-exp(n)
#log partial likelihood lpl=ln(L)
lple<-y[(1+p):NN]*n[(1+p):NN]-u[(1+p):NN]
lpl<-sum(lple)
lpl0<-lpl;
}

#Q are partial derivatives of eta on each coefficient
#Since the first p time points don't have the full AR terms
#Q matrix begin at time (p+1), Q[t,i] is the partial derivative
of eta on the
ith coefficient at time point (t+p).
rp<-rev(pp)
for(t in 1:(NN-p))
QQ[t,]<-as.matrix(tX[,t:(t+p-1)])%*%rp
Q[,1:(b+1)]<-X[(1+p):NN,]-QQ
Q[,(2+b):(1+b+p)]<-aa
#dpl is the partial derivative of partial likelihood on each coefficient#
edpl<-(y[(1+p):NN]-u[(1+p):NN])*Q;
dpl<-apply(edpl,2,sum)
#tt is fisher information matrix
tt<-matrix(0,b+p+1,b+p+1);
for(t in 1:(NN-p))
```

```
{tt[1:(1+b),1:(1+b+p)]<-tt[1:(1+b),1:(1+b+p)]+u[t+p]*Q[t,1:
(1+b)]%*%t(Q[t,1:(1+b+p)])
tt[(b+2):(b+p+1),(b+2):(b+p+1)]<-tt[(b+2):(b+p+1),(b+2):
(b+p+1)]+u[t+p]*Q[t,(b+2):(b+p+1)]%*%t(Q[t,(b+2):(b+p+1)])
for(j in (b+2):(b+p+1))
tt[1:(1+b),j]<-tt[1:(1+b),j]+(y[t+p]-u[t+p])*X[t+p+b+1-j,1:(1+b)]
}
tt[(2+b):(1+b+p),1:(1+b)]<-t(tt[1:(1+b),(2+b):(1+b+p)])
#eigen decomposition of tt
ev<-eigen(tt);
#val is the eigen values of tt
val<-ev$values;
#vec is the eigen vector matrix of tt
vec<-ev$vectors;
# if all eigen values are larger than de, tt1 is the inverse matrix of tt
# else
if(all(val>de))
tt1<-solve(tt)    else
{  tt[c(1:(b+1)),c((b+2):(b+p+1))]<-matrix(0,b+1,p);
tt[c((b+2):(b+p+1)),c(1:(b+1))]<-matrix(0,p,b+1);
tt1<-solve(tt);
}
#l indicates whether it is the first time to change the
coefficients in this iteration
l<-1;
#to get the new coefficients, add adr to the former coefficients
adr<-tt1%*%dpl;
#when the mode of adr is less than epsilon, make r[k+1]=r[k]
if(t(adr)%*%adr>epsilon)
{
#when the new lpl is larger than the old lpl, then r[k+1]= new coefficients
while(lpl<=lpl0)
```

```
{ if(l==1)
{
#new r=old r+adr
r[k+1,]<-r[k,]+adr
#change the indicator l
l=l+1;
} else      r[k+1,]<-r[k,]+runif(1,0,1.5)*adr;
#new r=old r+adr
#get lpl for comparison
bb<-r[k+1,c(1:(b+1))];
pp<-r[k+1,c((b+2):(b+p+1))];
a<-log(yy)-X%*%bb;
for(i in 1:p)
aa[,p-i+1]<-a[i:(NN-p+i-1)]
n[(p+1):NN]<-(X%*%bb)[(p+1):NN]+aa%*%pp+log(w)[(p+1):NN]
u<-exp(n)
lple<-y[(1+p):NN]*n[(1+p):NN]-u[(1+p):NN]
lpl<-sum(lple)
}
}  else
{r[k+1,]<-r[k,];
pan<-1;
}
lpl0<-lpl;k=k+1
}
fit<-list()
Estimate<-r[k,]
Std.Error<-sqrt(diag(tt1))
zvalue<- Estimate/Std.Error
Pr<-2 * pnorm(-abs(zvalue))
fit$coefficients<-cbind(Estimate, Std.Error, zvalue, Pr)
dimnames(fit$coefficients)<-list(c(xnames,paste("AR", 1:p.ar, sep = "")),
```

```
c("Estimate","Std.Error","z value","Pr(>|z|)"))
names(n)<-1:NN
names(u)<-1:NN
n<-n[-(1:p)]
u<-u[-(1:p)]
fit$linear.predictor<-n
fit$fitted.values<-u
y<-y[-(1:p)]
w<-w[-(1:p)]
dev.resids<-family$dev.resids
aic<-family$aic
Pearson.res<-(u-y)/sqrt(u)
phi<-sum(Pearson.res^2)/(NN-1-b-2*p)

fit$aic<-aic(y, mu=u, wt=w) + 2 * (1+b+p)
fit$dev<-sum(dev.resids(y, mu=u, wt=w))
fit$Pearson.residuals<-Pearson.res
fit$phi<-phi
fit$X<-X
fit$loglikelihood<--(1/2)*aic(y, mu=u, wt=w)
ll<-c("Intercept",term.labels,"AR")
fc<-c(attr(X, "assign"),rep(max(attr(X,"assign"))+1,p))
aaa<-factor(fc, labels = ll)
asgn<-split(order(fc), aaa)
nterms<-length(asgn)

predictor<-matrix(ncol = nterms, nrow = NN)
dimnames(predictor)<-list(rownames(X), names(asgn))
aab<-rbind(matrix(0,p,p),aa)
X1<-cbind(X,aab)

for (i in seq(1,nterms)) {
```

```
iipiv<-asgn[[i]]
predictor[, i]<-
X1[, iipiv, drop = FALSE] %*% Estimate[iipiv]
}
fit$term.predictors<-predictor[-(1:p),]
fit$pan<-pan
fit
}

#### Function defined for fitting GAM ####
grogam<-function(temp,Y0,b=6,NL,M){
temp<-temp[-(1:M)]
Y<-Y0[,-(1:M)]
par<-matrix(0,NL,1+b)            # beta coefficients are stored in par
cona<-array(0,dim=c(NL,1+b,2))   # upper limits and lower limits are stored
pan<-numeric(NL)
for (i in 1:NL)
{
y<-Y[i,]
gambb<-gam(y~ns(temp,b),family=poisson)
par[i,]<-gambb$coef
dia<-summary.glm(gambb)$coefficient[,2]*qnorm(0.975)
cona[i,,1]<-par[i,]-dia
cona[i,,2]<-par[i,]+dia
if(i==sample(1:1000,1,replace=FALSE))
res1<-residuals(gambb,type = "pearson")
if((i%%10)==0)
cat(i,"\n")
}
grofit0ar<-list()
grofit0ar$par<-par
grofit0ar$cona<-cona
```

```
grofit0ar$res1<-res1
grofit0ar
}

#### Function defined for fitting GAMAR ####
grogamAR<-function(temp,Y0,b=6,p=1,NL,M){
temp<-temp[-(1:M)]
Y<-Y0[,-(1:M)]

par<-matrix(0,NL,1+b+p)          # beta coefficients and autocorrelation
                                  coefficients are stored in par
cona<-array(0,dim=c(NL,1+b+p,2))
pan<-numeric(NL)
for (i in 1:NL)
{
y <-Y[i,]
glmbb<-gam(y~ns(temp,b),family=poisson)

## coefficients obtained GAM are used as initial value for GAMAR ##
fit<-gamAR(y~ns(temp,b),p.ar=p,starts=glmbb$coef)
par[i,]<-fit$coefficients[,1]
dia<-fit$coefficients[,2]*qnorm(0.975)
cona[i,,1]<-fit$coefficients[,1]-dia
cona[i,,2]<-fit$coefficients[,1]+dia
pan[i]<-fit$pan
if(i==1)
res1<-fit$Pearson.residuals
if((i%%10)==0)
cat(i,"\n")
}
grofit1ar<-list()
grofit1ar$pan<-pan
```

```
grofit1ar$par<-par
grofit1ar$cona<-cona
grofit1ar$res1<-res1
grofit1ar
}

N<- NROW(temp)
NL<- 1000
M=456                              # eliminated effect of initial value
grofit0ar<-grogam(temp=temp,Y0=Y0,b=6,NL=NL,M=M)
grofit1ar<-grogamAR(temp=temp,Y0=Y0,b=6,p=1,NL=NL,M=M)
resnar<-grofit0ar$res1
resar<-grofit1ar$res1

## residual plot for GAM and GAMAR ##
par(mfrow=c(2,2))
acf(resnar,main="GAM",ylim=c(0,1))
pacf(resnar,main="GAM",ylim=c(0,1))
acf(resar,main="GAMAR(1)",ylim=c(0,1))
pacf(resar,main="GAMAR(1)",ylim=c(0,1))

x<-temp[-(1:M)]
Xb<-ns(x,b)
X<-cbind(rep(1,NN),Xb)
no1<-no[(M+1):N]
ii<-1
yhat1<-X%*%parnar[ii,]
yhat2<-X%*%par[ii,1:(1+b)]
par(mfrow=c(1,1))
plot(x,no1,ylab="link predictor")
points(x,yhat1,col="red")
points(x,yhat2,col="blue")
```

```
#### Overdispersion ####
NN=N-M
phiar <-(sum(resar^2))/(NN-1-b-2*p)
phinar=(sum(resnar^2))/(NN-1-b)
pvalue=pchisq(phinar,df=NN-1-b,

#### Real Case Analysis (chapter 6) ####

### Choosing the degrees of Freedom ###

### We have used the data in year 2022-2023 ###

NN<-as.numeric(as.Date("2023-12-31")-as.Date("2021-12-31"))
time<-as.Date(1:NN,origin="2021-12-31")
weekday<-as.numeric(time-as.Date("2021-12-31"))%%7
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)

### initially we will consider all the variables have 10 df except time ###
parafind<-function(dftime=5,dftemp1=10,dfvis=10,dfwind=10,
dfhum=10,dfrain=10,lag1=10){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
for (i in 1:15) {
aa<-parafind(lag1=i)
```

```
if (MIN>aa) {
MIN<-aa
lag1<- i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best lag1:",lag1, "\n")

#### We have found lag1=4 ####
parafind<-function(dftime=5,dftemp1=10,dfvis=10,dfwind=10,
dfhum=10,dfrain=10,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
for (i in 1:15) {
aa<-parafind(dftemp1=i)
if (MIN>aa) {
MIN<-aa
dftemp1<-i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best dftemp1:",dftemp1, "\n")

#### We have found dftemp1=10 ####
```

```
parafind<-function(dftime=5,dftemp1=10,dfvis=10,dfwind=10,
dfhum=10,dfrain=10,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
for (i in 1:15) {
aa<-parafind(dfvis=i)
if (MIN>aa) {
MIN<-aa
dfvis<- i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best dfvis:",dfvis, "\n")

#### We have found dfvis=7 ####
parafind<-function(dftime=5,dftemp1=10,dfvis=7,dfwind=10,
dfhum=10,dfrain=10,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
```

```
for (i in 1:15) {
aa<-parafind(dfwind=i)
if (MIN>aa) {
MIN<-aa
dfwind<- i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best dfwind:",dfwind, "\n")

#### We have found dfwind=2 ####
parafind<-function(dftime=5,dftemp1=10,dfvis=7,dfwind=2,
dfhum=10,dfrain=10,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
for (i in 1:15) {
aa<-parafind(dfhum=i)
if (MIN>aa) {
MIN<-aa
dfhum<-i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best dfhum:",dfhum, "\n")
```

```
#### We have found dfhum=3 ####
parafind<-function(dftime=5,dftemp1=10,dfvis=7,dfwind=10,
dfhum=3,dfrain=10,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}

MIN<-Inf
for (i in 1:15) {
aa<-parafind(dfrain=i)
if (MIN>aa) {
MIN<-aa
dfrain<-i
}
if (aa>MIN) break
}
cat("Minimum AIC:", MIN, "\n")
cat("Best dfrain:",dfrain, "\n")

#### We have found dfrain=1 ####
parafind<-function(dftime=5,dftemp1=10,dfvis=7,dfwind=2,
dfhum=3,dfrain=,lag1=4){
temp1<-temp[(11-lag1):(740-lag1)]
formula0<-count~ns(time,dftime)+ns(temp1,dftemp1)+ns(vis,dfvis)+
ns(wind,dfwind)+ns(hum,dfhum)+ns(rain,dfrain)+as.factor(weekday)
gambb<-gam(formula0, family = poisson)
gambb$aic
}
```

```
MIN<-parafind()
best_params<-list()
for (i1 in 4:6) {
for (i2 in 9:11){
for (i3 in 8:9) {
for (i4 in 1:3) {
for (i5 in 2:4) {
for (i6 in 3:5) {
for (i7 in 3:5) {
aa <- parafind(dftime = i1, dftemp1 = i2,  dfvis = i3, dfwind = i4,
dfhum = i5, dfrain = i6, lag1 = i7)
if (MIN > aa) {
 MIN <- aa
 best_params <- list(dftime = i1, dftemp1 = i2,
 dfvis = i3, dfwind = i4, lag1 = i5)
} } } } } } } }
cat("Optimal parameters:\n")
cat("dftime:", best_params$dftime, "\n")
cat("dftemp1:", best_params$dftemp1, "\n")
cat("dfvis:", best_params$dfvis, "\n")
cat("dfwind:", best_params$dfwind, "\n")
cat("dfhum:", best_params$dfhum, "\n")
cat("dfrain:", best_params$dfrain, "\n")
cat("lag1:", best_params$lag1, "\n")
cat("Minimum AIC:", MIN, "\n")

#### We have continued the procedure until the smallest AIC remained
unchanged, so we have found the parameters which minimize AIC locally.
Finally we have found dftime=3, dftemp1=10, dfvis=9, dfwind=2, dfhum=3,
dfrain=1, lag1=6 ####
```

```
### W folllowed the same procedure for lagged humidity ###

### Fitting GAM in real data ###
gambb<-gam(formula0, family = poisson)
starts<-gambb$coef

### Fitting GAMAR in real data ###
### coefficients of GAM will be used as initial value for GAMAR ###
abc1<-gamAR(formula0, p.ar=6, starts=starts)

resnar<-residuals(gambb,type = "pearson")      # residuals of GAM
resar<- abc1$Pearson.residuals                 # residuals of GAMAR
## residual plot ##
par(mfrow=c(2,2))
acf(resnar,main="GAM",ylim=c(0,1))
pacf(resnar,main="GAM",ylim=c(0,1))
acf(resar,main="GAMAR(6)",ylim=c(0,1))
pacf(resar,main="GAMAR(6)",ylim=c(0,1))

## partial effect of temperature from GAM ##
temp1.pre<-ns(temp1,dftemp1)%*%gambb$coef[(2+dftime):(1+dftime+dftemp1)]

## partial effect of temperature from GAMAR ##
temp1.pre.ar <-abc1$term.predictors[,3]

## partial effect plot for GAM ##
plot(temp1,temp1.pre,ylab="Effect of temperature1",xlab="Temperature1",
pch=16,cex=.5,col="brown2")

## partial effect plot for GAMAR ##
plot(temp1[-(1:7)],temp1.pre.ar,col="deepskyblue4",ylab="Effect of
temperature1", xlab="Temperature1",,pch=16,cex=.5)
```