University of Tartu
Faculty of Science and Technology
Institute of Mathematics and Statistics

Dhruba R. Gnawali

GLARMA time series modeling of counts

Actuarial and Financial Engineering
Master's Thesis (30 ECTS)

Supervisor: Märt Möls, Ph.D.

Tartu 2023

Contents

GLARMA TIME SERIES MODELING OF COUNTS

Master thesis

Dhruba R. Gnawali

Abstract

This thesis investigates the potential of using GLARMA (Generalised Linear Autoregressive Moving Average) models in insurance. Traditional time series analysis assumes a Gaussian distribution for the dependent variable, which may not be suitable for discrete variables like the number of accidents. GLARMA models provide an alternative by incorporating autoregressive or moving average models for variables that follow Poisson or negative binomial distributions, making them an appealing choice for insurance applications. The objective of this thesis is to explain the GLARMA modeling framework, apply it to predict the number of accidents in Finland and assess the limitations and benefits of this approach. The study employs the Glama package to implement the GLARMA model on car accident datasets. Through a comparative study with two other ARMAX models, it is found that the GLARMA model provides a comparatively better framework for forecasting car accidents in Finland. The forecasted data reveals that accident incidence typically peaks during the summer months (June to August) and decreases during the winter months (December to February). The observed pattern is primarily attributed to the increase in traffic volume. This study introduces the promising possibilities of utilizing the GLARMA model in insurance, particularly in scenarios where count data is prevalent.

research specialization: P160 Statistics, operation research, programming, actuarial mathematics

Key Words: Count data, GLARMA, modeling, forecast

GLARMA MUDEL DISKREETSETE AEGRIDADE JAOKS

Magistritöö
Dhruba R. Gnawali

Lühikokkuvõte

Käesolev magistritöö uurib GLARMA (Generalised Linear Autoregressive Moving Average) mudelite kasutamise potentsiaali kindlustuses. Traditsiooniline aegridade analüüs eeldab, et uuritav tunnus on normaaljaotusega. Paljud kindlustusvaldkonnas esinevad tunnused - nagu näiteks aset leidnud õnnetuste arv - on aga diskreetsed. Normaaljaotuse eeldus selliste diskreetsete tunnuste korral ei kehti. GLARMA mudelid lisavad diskreetsete tunnuste modelleerimiseks mõeldud üldistatud lineaarsetele mudelitele aegridade analüüsist tuttavaid elemente, lubades sõltuvust eelnenud vaatlustest või prognoosivigadest. Kas saadud mudelid on aga ka praktikas kasutatavad ja kas keerulisemad GLARMA mudelid võimaldavad ka tegelikult uuritavat tunnust täpsemalt prognoosida? Neile küsimustele vastamiseks kasutatakse R tarkvara lisamoodulit glarma Soomes juhtuvate liiklusõnnetuste arvu prognoosimiseks. Saadud prognoose võrreldakse klassikaliste ARMAX mudelite abil saadud prognoosidega. Selgub, et GLARMA mudel võimaldab täpsemalt prognoosida tulevikus aset leidvate liiklusõnnetuste arvu

CERCS teaduseriala: P160 Statistika, operatsioonanalüüs, programmeerimine, finants- ja kindlustusmatemaatika
Märksõnad: Andmete loendamine, GLARMA, modelleerimine, prognoos

1 Introduction

Uncertainty is a ubiquitous feature of the world we live in, and time series analysis can provide a set of tools for understanding it[1]. In particular, when it comes to accidents, which involve numerous risk factors and probabilities[2]. Every year, road accidents claim the lives of more than 1.2 million people globally[3]. Apart from the devastating human toll, road accidents have significant financial implications. In 2020, the European road safety observatory reported a total of 223 fatalities resulting from traffic accidents in Finland. In contrast, Finish Insurance Report 2020 disclosed that 669 million euros were paid out in claims for motor vehicle insurance during the same period. Despite this, the country has one of the lowest numbers of fatalities per million inhabitants among the 27 EU member states, ranking 12th. However, compared to the EU average, Finland has a relatively high proportion of car occupant fatalities (Figure 1). The risk related to the number of accidents may be correlated by various factors including weather conditions, time of day, driver behavior, road condition, age and experience of the driver, vehicle type, location, and more. Hence, a scientific study and forecast of future car accidents are crucial in mitigating the risks associated with such accidents. Our objective is to introduce a novel time series modeling framework to forecast future accidents using historical data for counts of accidents. Additionally, we seek to comprehend the likelihood of insurance claims resulting from that car accidents.

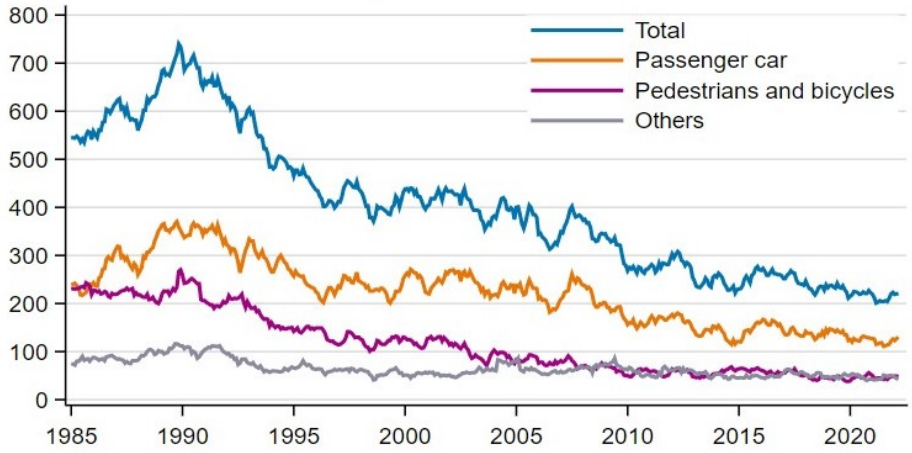

Figure 1: Trend of fatality in road traffic accident in Finland from 1985 to 2022, preliminary data by month[3]

A time series is an ordered sequence of stochastic observations $\{X(t) : t \in T\}$ that are recorded over time, and it has diverse applications across various fields. With the advancement of technology, there has been an increase in the availability of a multitude of time series data sets. This has facilitated the exploration of the underlying data-generating mechanisms of time series data[4]. The goal of time series analysis is to understand the patterns and relationships that exist within the time series data and to use this understanding to make predictions about future observations. Count data series refer to discrete event counts that are recorded over a specific time inter-

val. These measurements are usually taken at regular intervals. We have well-known models such as ARMA, ARIMA, SARIMA, and ARMAX that assume the response variable is normally distributed and continuous making them suitable for explaining the continuous time series but there is a significant gap in the development of flexible models that can better explain count data processes.

In regression models, where the focus is on making correct inferences about the impact of covariates on the response series, accounting for serial dependence is critical. However, modeling dependence when the outcomes are discrete random variables can be challenging since traditional methods for detecting serial dependence in regression residuals are often ineffective[5]. The need for modeling discrete response time series has become increasingly prevalent in a wide range of fields, such as econometrics, actuarial science, finance, public policy assessment, environmental science, and disease incidence modeling. Considerable progress has been made over the past two decades in developing models for discrete-valued time series, the GLARMA is one of them.

The GLARMA term encapsulates the fundamental nature of the models under consideration, which can be understood as a synthesis of generalized linear (GL) and autoregressive moving average (ARMA) models. The state process in GLARMA models depends linearly on covariates and non-linearly on past values of the observed process. Given the state process, the observations are independent and follow a distribution from the exponential family, which can accommodate three distributions such as Poisson, Negative binomial, and Binomial.[6] Therefore, GLARMA models offer a flexible framework for modeling complex, non-linear relationships between covariates and discrete response variables, which could offer valuable insights for our project.

This thesis examines the GLARMA (generalized linear autoregressive moving average) time series model utilizing a dataset from car accident count data that occurred in Finland and attempts to fit a GLARMA model and based on this model forecast future car accidents in Finland. To assess the performance of the model, a comparison is made with ARMAX models, and the most optimal model is selected as the final choice. This can provide valuable insights for stakeholders including insurance companies to develop effective strategies toward their goals.

2 Methods

2.1 Model Formulation

The generalized linear autoregressive moving average (GLARMA) models have been introduced to encompass a wide range of observation-driven models. While there isn't a single model class that includes all of them, GLARMA models serve as a relatively comprehensive category[7]. The following is a summary of the model.

Let Y_t be a time series of counts comprised of values of the discrete random variables $\{Y_t : t = 1, ..., n\}$ where, y_t represents a specific value of the response variable at a particular time point. x_t be an observed p-dimensional vector of regressors available at the time t and n is consecutive times at which the response and regressor series are observed. Let W_t be the historical information available on the response series and past and present information on the regressors. In GLARMA model The distribution of response variable Y_t conditional on W_t is assumed to be an exponential family of the form.

$$f(y_t|W_t) = \exp\{y_t W_t - a_t b(W_t) + c_t\} \tag{1}$$

While equation eq(1) is not the fully general form of the exponential family, it encompasses several popular and useful distributions. It can be easily adapted to include other response distributions or more general specifications of link functions[5]. An example of such an extension is the use of the negative binomial response distribution.

$$f(y_t|W_t,\alpha) = \frac{\Gamma(\alpha_t + y_t)}{\Gamma(\alpha)\Gamma((y_t+1)} \left[\frac{\alpha}{\alpha+\mu_t}\right]^{\alpha} \left[\frac{\mu_t}{\alpha+\mu_t}\right]^{y_t} \tag{2}$$

Where μ_t is the mean of the distribution. It includes an extra parameter α which controls the degree of overdispersion in the distribution. As alpha approaches infinity, the negative binomial density approaches and converges to the Poisson density

$$f(y_t|W_t) = \frac{e^{\mu_t}{\mu_t}^{y_t}}{y_t!}. \tag{3}$$

The state process in the GLARMA model depends linearly on covariates and nonlinearly on the past value of the observed process

$$W_t = x_t^T\beta + Z_t. \tag{4}$$

The general form of the GLARMA model can be described by specifying Z_t as an autoregressive moving average recursion of the form [6]

$$Z_t = \sum_{i=1}^{p} \phi_i Z_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i}. \tag{5}$$

It can be conceptualized as a combination of generalized linear and ARMA models.

The predictive residuals can be formulated as

$$e_t = \frac{Y_t - \mu_t}{\vartheta_t}. \tag{6}$$

Where ϑ_t is the scaling sequence to be selected

2.2 Model fit

we use maximum likelihood estimation to fit our model, utilizing iterative methods such as Newton's Raphson and Fisher scoring. For a response series consisting of n successive observations $y_t(t = 1, 2.....n)$, and fixed initial conditions for the recursions, we construct the likelihood as the product of the conditional density of Y_t given F_t.[6] This results in the log-likelihood corresponding to the specific distribution being used.

$$I(\delta) = \sum_{t=1}^{n} \log f(y_t|W_t(\delta)) \tag{7}$$

The Glarma package[**?**] is designed to handle only two types of response distributions: binomial and Poisson. For these specific distributions, the log-likelihood can be calculated[5]

$$I(\delta) = \sum_{t=1}^{n} y_t W_t(\delta) - a_t b(W_t(\delta)) + c_t. \tag{8}$$

where, $\delta = (\beta, \phi, \theta)$ for the negative binomial response distribution the log-likelihood is more complicated because shape parameter α also has to be estimated along with the β, ϕ and θ we maximize the likelihood by starting from a suitable initial value of the parameter δ and using a Fisher scoring iteration method.

2.3 Selecting appropriate lags for the ARMA components

Determining the appropriate lags for the autoregressive (AR) and moving average (MA) components in the GLARMA model is generally more challenging when dealing with discrete-valued response data compared to the Gaussian series. In the case of the Gaussian series, residuals obtained from the least squares model fitting process can offer valuable insights into the relationship between data points using autocorrelation and partial autocorrelation functions. However, when working with discrete-valued responses in the GLM framework, the residuals from the GLM fit are often less helpful in guiding the selection of the appropriate p and q values required to specify the GLARMA model, especially when the dependence between data points is weak. It would be better to utilize the GLM residuals to estimate the autocorrelation and partial autocorrelation functions only if there are significant patterns observed in the residuals[5]. It is recommended to start with low orders for both AR and MA. A once stable estimation has been achieved for a lower-order specification, higher values of AR or MA can be tried in order to achieve stability[6]. In our specific scenario, the GLARMA model with phiLags and thetaLags of the same order 7 and 11 successfully converged and resulted in the lowest AIC value.

2.4 Forecasting with GLARMA model

Forecasting future values of time series using discrete responses is not as advanced as traditional ARMA or ARMAX models for continuous responses. For one-step-ahead forecasts, the GLARMA model estimates the predictive distribution for Y_{n+1} by forecasting the state variable W_{n+1} [5].

$$W_{n+1} = \hat{x}_{n+1}\beta + \hat{Z}_{n+1} \tag{9}$$

$$\hat{Z}_{n+1} = \sum_{j=1}^{p} \hat{\phi}_j \hat{Z}_{n+1-j} + \sum_{j=1}^{q} \hat{\theta}_j \hat{e}_{n+1-j} \tag{10}$$

Where, Z_{n+1} is determined using the value Z_t and e_t. The predictive distribution for Y_{n+1} is estimated to be $f(y_t|\, W_{n+1})$ where f is the density function of Equation(2). Multi-step ahead forecasts are more complex, and complete enumeration of the sums and products is often impossible. Simulation is a feasible approach for short-range forecasting and is used in the glarma. package[8].

2.5 Glarma Package

The Glarma function is utilized to construct generalized linear autoregressive moving average models that can accommodate multiple distributions such as Poisson, binomial, and negative binomial. It can use Pearson residuals, score residuals, or identity residuals for the binomial distribution to fit the model. Additionally, it estimates the GLARMA model parameters by employing either Fisher scoring or Newton-Raphson iteration[8]. The log link is implemented for Poisson and negative binomial response distributions.

Glarma models are expressed symbolically with a typical structure of y (response) and X (terms). The y vector represents the count response, while X denotes a set of terms that define a linear predictor for the response. It is important to include a vector of 1s as the first column of X to represent the intercept in the model. The

initial parameters for the model consist of β, ϕ, θ, and an optional parameter α, which is used for negative binomial models [8].

The package[8] also offers likelihood ratio and Wald tests which enables testing of serial dependence in generalized linear model settings. Graphical diagnostics, such as model fits, autocorrelation functions, and probability integral transform residuals, are included in the package.

3 Data source and data processing

The data used in this study was Car accident data, retrieved from Statistics Finland and the data was obtained from records of road traffic accidents involving personal injury that were reported by the police to Statistics Finland. It has 239 observations from 2003 to 2022. The explanatory variables included seasonal effects. The seasonal effect was introduced as a categorical variable with four levels corresponding to the seasons: winter, spring, and autumn keeping summer as a reference season. In addition to this to increase the accuracy of the forecast number of days in the month, month numbers are included as a data component. The trend of car accidents (Figure2) shows that the number of car accidents is decreasing from 2003 to 2022.

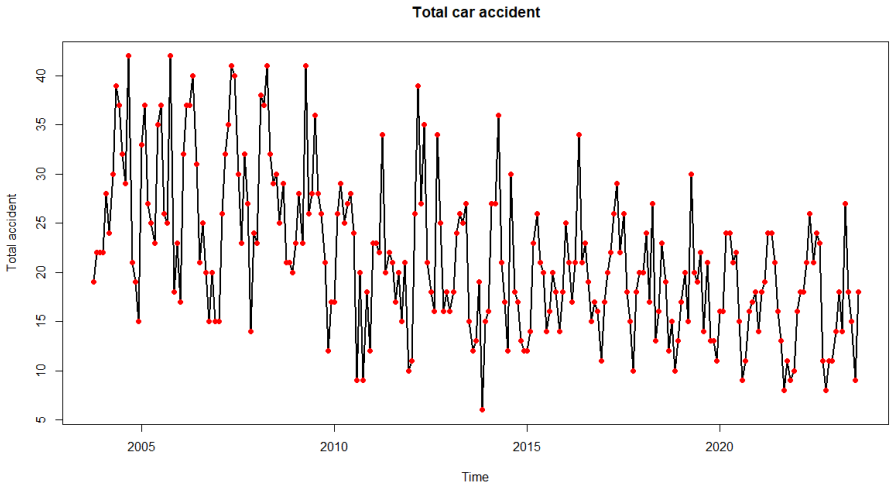

Figure 2: Trend of car Accidents in Finland During 2003 - 2022

3.1 Statistical Methods

We employed the GLARMA modeling framework using the Glarma package in the R statistical program, to analyze the historical time series of accident counts. The choice of response distribution for the time series was based on the estimated value of the shape parameter. If the shape parameter was found to be statistically significant, a negative binomial distribution was selected to account for over-dispersion, which

occurs when the data exhibits more variation than expected from a Poisson distribution. Conversely, if the shape parameter was not significant, a Poisson distribution was chosen, assuming equal variance and mean, which is suitable for modeling count data with low variability [9]. By selecting the appropriate distribution, the model can effectively capture the underlying patterns in the data and can generate more accurate forecasts. To assess whether the response demonstrated serial dependence, we conducted two tests: the likelihood ratio test and the Wald test, both available in the glarma package[5]. These tests help determine if there is a significant correlation between observations in the time series. The best model was determined using statistical measures such as the Root Mean Square Error (RMSE) and the Akaike Information Criterion (AIC). These measures provide valuable insights into the model's predictive performance and allow for the comparison of different models to identify the most suitable one.

4 Result

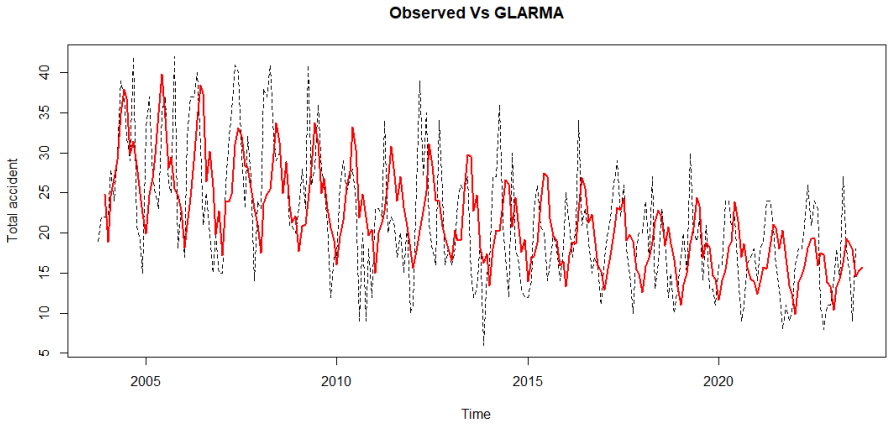

Figure 3: Graph count versus Time where the Black dotted line represents the observed data sets and the thick red line represents the estimated total number of car accidents based on historical trends and current data using GLARMA model

We utilized the Generalized Linear Auto-Regressive Moving Average (GLARMA) model to analyze car accident data. To ensure the accuracy of our analysis, we carefully determined the optimal order of auto-regressive and moving average parameters for the GLARMA model. Subsequently, we visualized the comparison between the observed data and the fitted GLARMA model in (Figure 3). Notably, we observed that the GLARMA model closely follows the trend of the observation data.

To see the performance of the GLARMA model, we conducted diagnostic checks. We plotted various diagnostic plots, including the Auto-Correlation Function (ACF) (Figure5), Partial Auto-Correlation Function(PACF) plots(Figure 6), QQ plot(Figure 7), score residual plots(Figure 4, bottom right), and histograms of probability integral

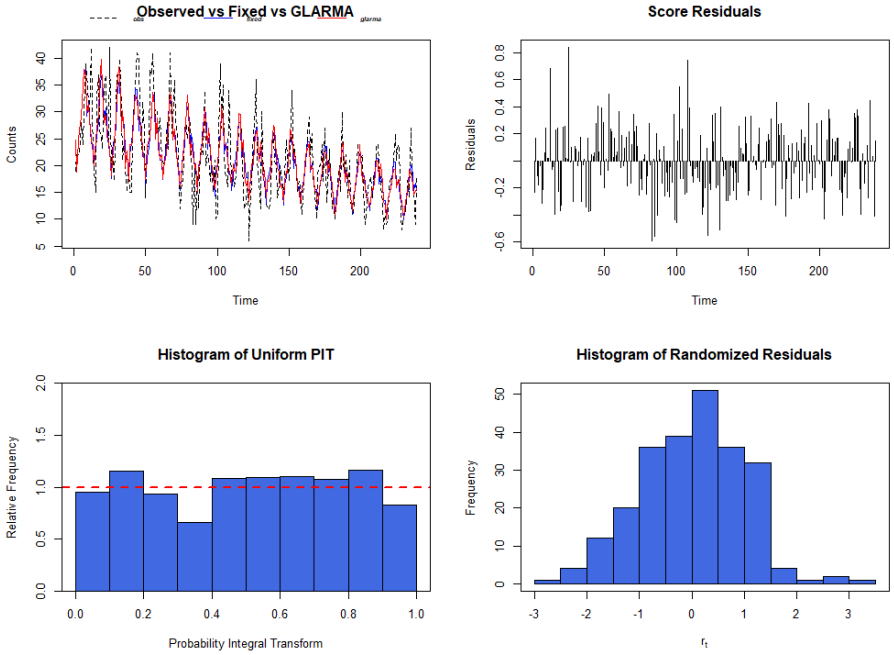

Figure 4: In the square, there are four distinct components. At the top left corner, there is a graph that depicts the observed data (represented by a black dotted line), the fitted graph (shown as a blue line), and the graph of the GLARMA model (displayed as a red line). Moving to the top right corner, there is a representation of the score residuals. At the bottom left corner, there is a histogram that displays the uniform probability integral transform. Finally, at the bottom right corner, there is another histogram that shows the randomized residuals, and upon closer examination of the fitted graphs (Figure 2, top left) noticed that the fitted curve also appeared to move in a similar pattern.

transform(Figure 4, bottom left)) and randomized residuals (Figure 4, bottom right). The score residuals were found to be distributed between -0.6 and 0.8 (top right)(Figure 4) indicating good model performance and the histogram of the probability integral transform displayed an almost uniform distribution. Furthermore, the histogram of randomized residuals exhibited a near-Gaussian distribution.

Of particular interest was the ACF (Figure 5)and PACF(Figure 6) plots, which showed that all the lags remained within the bounded region.

4.1 Fitted GLARMA model

GLARMA model was fitted with the Negative Binomial distribution using Phi and theta lags of equal order 7 and 11. The model also incorporated independent variables, including an intercept term, the logarithm of days, and three seasonal variables (summer, autumn, and spring) to account for seasonality. The summer month was

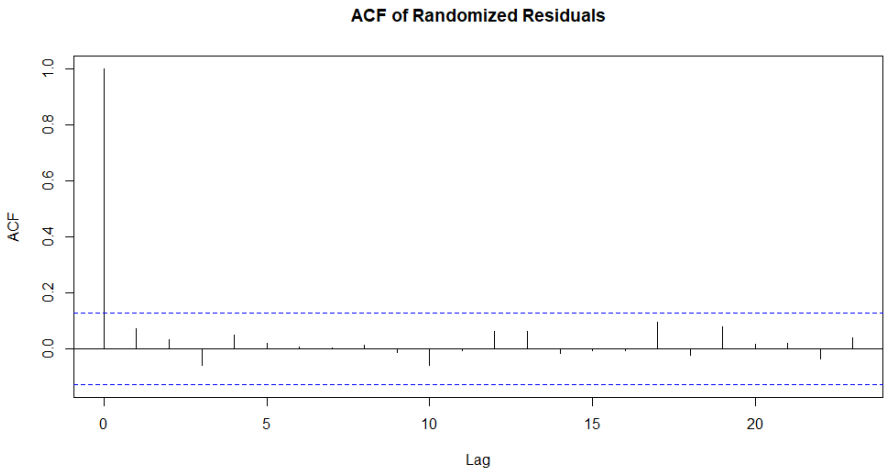

Figure 5: ACF of randomized residuals of GLARMA model

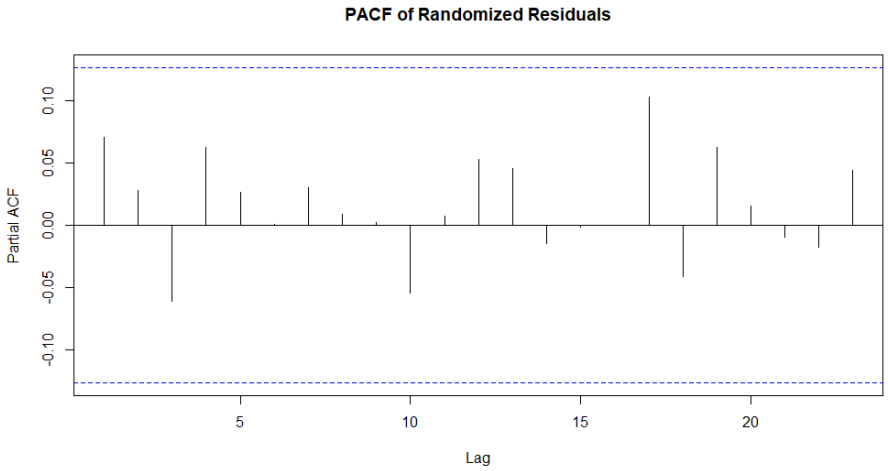

Figure 6: PACF of randomized residuals of GLARMA model

used as the reference. The model can be expressed statistically as follows

$$\text{E(Total accident} \mid W_t) = \exp\Big((5.47 + 2.65 \times logdays_t - 0.44_{\{season_t = winter\}} - 0.16 \times I_{\{season_t = autumn\}} - 0.27 \times I_{\{season_t = spring\}} - 0.02 \times Monthnumber_t) +$$

$$0.14 \times (Z_{\{t-7\}} + e_{\{t-7\}}) - 0.19 \times (Z_{\{t-11\}} + e_{\{t-11\}})\Big). \quad (11)$$

4.2 Model validation

In order to assess the predictive capabilities of the models the dataset is divided into a training set and a test set. The training set comprises 8 years of sample data (2003 to 2011) and is used to construct the models. The test set spans 9 years of sample data (2012 to 2021) and is employed to assess the predictive accuracy[10]. In each iteration, the training set's length increases by one year, while the test set's length decreases by one year, until reaching one year prior to the final data point. The forecasting process commences by estimating the model using the training data. Subsequently, the estimated model is utilized to generate one-step-ahead forecasts for the total number of car accidents. To evaluate the accuracy of the forecasts, the predicted mean of the total number of car accidents within a year is compared to the actual(True) total number of car accidents in that year.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Forecast_i - True_i)^2} \quad (12)$$

To calculate the Root Mean Square Error (RMSE), we use Equation (14), which involves computing the root mean of the difference between the forecasted number of car accidents (Forecast) generated by our model and the actual number of car accidents (True) available in the historical dataset. The model with the lowest Root Mean Square Error (RMSE) and Akaike Information Criterion (AIC) values would be considered the best model among the options.

5 Discussion

The primary aim of our study was to introduce a GLARMA modeling framework for forecasting car accidents and assess the limitation and benefits of the approach. GLARMA time series model that is specifically tailored for count data with discrete response variables and can effectively address the issue of serial dependence in the residuals. Traditional models such as ARMA, ARIMA, and SARIMA, which assume continuous and normally distributed response variables, may not be suitable for modeling the discrete nature of car accident data. Additionally, while GLM can capture the relationship between the response and predictor variables, it may not adequately capture the autocorrelation in the residuals, making it less ideal for modeling time series data with serial dependence. To overcome these limitations, we opted for the GLARMA (Generalized Linear Autoregressive Moving Average) model as a potential solution. The GLARMA model combines the strengths of both GLM and ARMA, allowing for the modeling of the conditional mean of the response variable through the GLM component, and the conditional variance of the residuals through the ARMA component. This makes it well-suited for capturing non-linear relationships between the discrete response and predictor variables, while also addressing the issue of serial dependence in the residuals[11]. In the first part of our study, we implemented the GLARMA model on car accident data and conducted an analysis of its diagnostics results. Initially, we fitted the model with the Poisson distribution, which resulted in

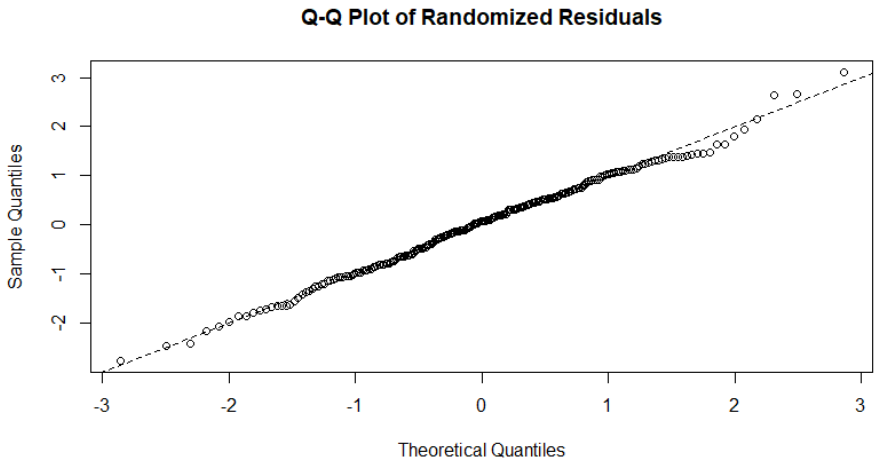

Figure 7: QQ plot of randomized residuals

a large overdispersion value of 2.78 indicating poor model fit. To address this issue, we switched to using the negative binomial distribution and found that the overdispersion value reduced to 1.71, which provided a better fit for the data and improved the model's performance. Our GLARMA model was fitted with autoregressive and moving averages of the same orders 7 and 11, using the negative binomial distribution and model fully conversed. Firstly, examining the output graphs of the GLARMA model (Figure4, top left) revealed that the fitted curve(blue dotted line) and red curve appeared to move in a similar fashion.

To understand this phenomenon, we try to clarify the interpretation of these graphs, the fixed graphs provide an estimation of the fixed effect of the predictor variables on the count of the response variables over time[8]. The fixed effect is calculated only from $X^T\beta$ in contrast, the Glarma effect is calculated from both $X^T\beta$ and Z_t.

$$Fit_{fixed} = \exp \eta = \exp(X^T\beta) \tag{13}$$

$$Fit_{Glarma} = \exp W_t = \exp(X^T\beta + Z_t) \tag{14}$$

In specific to our result; the fixed graph and the GLARMA model graph are moving in a similar fashion(blue dotted line)(Figure4) (top left)), this suggests that the model is consistent in capturing the underlying patterns in the data.

Score residuals represent the standardized residuals obtained by dividing the residuals by the estimated standard deviation and score residuals fall within the range of -0.6 to 0.8(4), which generally indicates the accuracy and consistency of the model's predictions. PIT is a statistical technique used to transform the random variable from any arbitrary distribution into a uniform distribution[12]. In our case, the PIT histogram which closely resembles a straight line at the top with up and downs in certain places, suggests how the model's predicted probabilities are closer to the observed data[5]. The nearly straight line pattern in the diagonal of the QQ plot(Figure 7) and

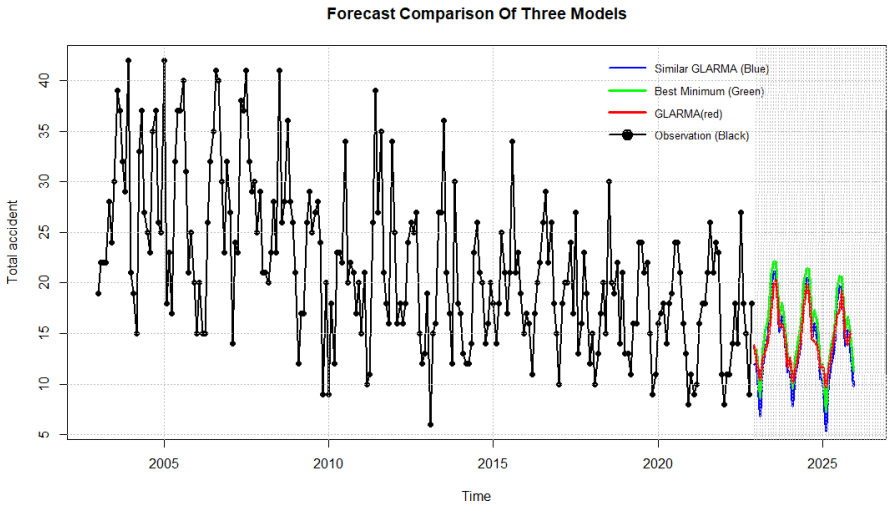

Figure 8: Combined graph of observed and forecasted total number of car accidents. The x-axis represents time, and the y-axis represents the total number of car accidents. The black line represents the observed data, obtained from historical records. The figure presented illustrates a comparison of three distinct models used for forecasting the total number of car accidents. The first model, "similar GLARMA,"(Blue line) is an ARMAX model that employs the same parameters as our GLARMA model. The second model, "best minimum,"(Green line) uses the minimum lags possible to fit the model and is also an ARMAX model. The third and final model, "our GLARMA,"(Red line) is the GLARMA model that we developed in our work.

a nearly Gaussian-shaped histogram of randomized residuals indicate that residues are nearly normally distributed[13].

The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are essential tools in analyzing autocorrelation in time series data. The GLARMA model output revealed that all the lag values in both the ACF and PACF fall within the bounded region, indicating the absence of significant autocorrelation in the residuals[14]. This suggests that the model has effectively captured the temporal dependence in the data, and the residuals exhibit characteristics of independent and identically distributed random variables. Moreover, it implies that the model accurately accounts for serial correlation, and there are no discernible patterns in the residuals. This valuable information enables us to assess the adequacy of the model and validate its assumptions. Additionally, the p-value (0.422) of the L-Jung box test further confirmed that the residuals are white noise, and there is no autocorrelation in the residuals of the fitted model[15].

To investigate whether serial dependence was present in our response, we utilized the likelihood ratio and Wald test which is available in glarma package. Specifically, it compared the likelihood of each model with that of a GLM model possessing the same

regression structure [5]. For both tests, the null hypothesis was the absence of serial dependence, with the GLM model being sufficient. On the other hand, the alternative hypothesis was the presence of serial dependence. We evaluated the statistics obtained from these tests against the chi-squared distribution, where the degrees of freedom were determined by the number of ARMA parameters[5]. The results of the Likelihood ratio test and Wald test with p values 2e-16 for both tests support that the model follows the GLARMA processes, as opposed to a Generalized Linear Model (GLM) with the same regression structure.

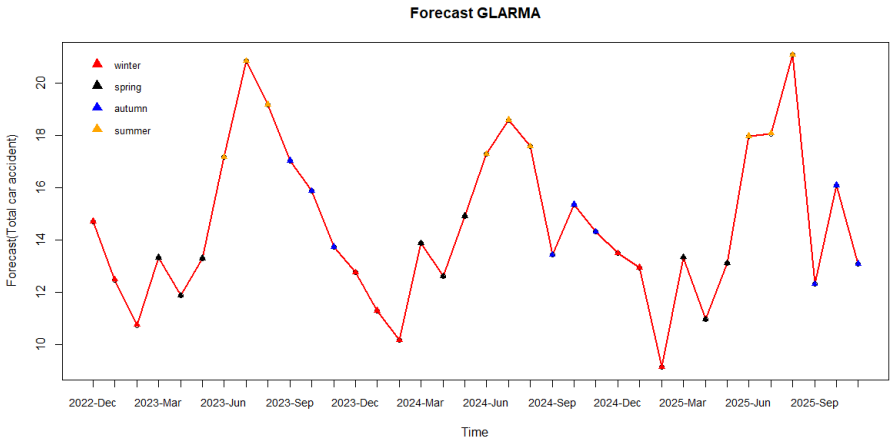

Figure 9: The figure presented above shows the seasonal forecast of the total number of car accidents from 2020 to 2026. The graph represents the four seasons of the year; winter is denoted by red spots, spring by black, autumn by blue, and summer by yellow. It is evident from the graph that the highest number of car accidents occurs in the summer months, whereas the lowest number of car accidents is reported during the winter months.

As part of our investigation into the potential of the GLARMA model, we initially examined its goodness of fit, which resulted in an AIC value of 1468.9. To further explore its potential, we conducted additional analyses by fitting two ARMAX models: one called "similar GLARMA," with the same parameters as the GLARMA model, and another called "best minimum," which used the minimum possible lags. All three models were employed to forecast the total number of car accidents from December 2021 to December 2025, with a one-month forecast horizon. The forecast comparison, presented in the single graph (Figure 8), reveals similar patterns among all three models in terms of the forecasted values (represented by the dark region of the graph). Additionally, the forecast results from both ARMAX models exhibit comparable seasonal patterns, such as an increase in accidents during the summer and a decrease during the winter, as depicted in (Figure 15) and (Figure 16).

To validate the model's performance, we decided to divide the data into training and test sets. The models were fitted using the training data, and their forecasting capabilities were evaluated on the test data. The total number of accidents was forecasted for a one-year horizon, and the results were presented in a single graph (Figure

Table 1: AIC and RMSE of three models

Metric	GLARMA	SimilarGLARMA	Best Minimum
AIC	1468.9	1497.9	1510.8
RMSE	18.8	25.7	36.0

10) alongside the true number of accidents. The GLARMA model (red line) achieved the lowest root mean squared error (RMSE) of 18.8, indicating comparatively better predictive accuracy. Its corresponding Akaike Information Criterion (AIC) value was 1468.9. The similar GLARMA model (blue line) yielded an intermediate RMSE of 25.7, with an AIC value of 1497.9. Another model with the minimum lags (best minimum) had a higher RMSE of 36.0 and an AIC value of 1510.8. These metrics provide insights into the accuracy and fit of the models for forecasting accidents. We analyze ACF(Figure 11) (Figure 13) and PACF (Figure 12) (Figure 14) of both ARAMX models and both reveal that there is no autocorrelation in residuals and all lags are within the bounded region. However, considering the RMSE and AIC values, we favored the GLARMA model over the other ARMAX models. Additionally, we argue that the GLARMA model is the most suitable among the three since it assumes a discrete response, which aligns well with our dataset.

The car accident data for Finland exhibits a fluctuating pattern in the number of accidents over the years and months. The forecast data (Figure 9), (Figure 15) and (Figure 16) shows that accidents tend to peak during the summer months, from June to August, and decrease during the winter months, from December to February. The pattern of car accidents in Finland can be influenced by various risk factors, such as weather conditions, time of day, driver behavior, road condition, driver age and experience, vehicle type, and location. Finland's severe winters, with heavy snow and icy roads, make driving more dangerous and increase the risk of accidents, but the number of cars on the roads tends to decrease during these extreme weather conditions. In contrast, the summer months typically have better weather conditions and longer daylight hours, leading to an increase in traffic volume and a higher risk of accidents. The holiday season, long daylight, and pleasant weather can encourage gatherings and activities, increasing the likelihood of alcohol consumption while driving and adding to the risk of accidents. A study in Finland showed that a tire's tread pattern depth of less than 1.6 mm was 3 times more likely to result in a fatal crash than a depth of 3.5 mm or above. A low albeit legal tread pattern depth (1.6 to 4 mm) increased the risk of an accident by 40 %[16].In addition to this, the traffic injury research foundation 2017 reported that winter tires outperform all-season tires and it is strictly implemented in Finland from December to February. Thus we can argue that winter tires can be a key possible measure to control the number of accidents during winter. The forecast for car accidents in Finland emphasizes the risk of accidents during the summer months. It can be recommended that adhere to tire tread pattern depth of more than 2 mm even in summer and mandatory implementation of winter tires during the winter and strict traffic safety measures in all seasons to reduce the number of car accidents in Finland.

Finally, the total number of car accidents and insurance claims are closely related since a car accident can result in insurance claims [17, 18, 19, 20]. The GLARMA modeling framework could be a promising possibility in insurance. From our study it is revealed that the total number of accidents will be higher in summer due to increased traffic volume, there are likely more insurance claims in summer.

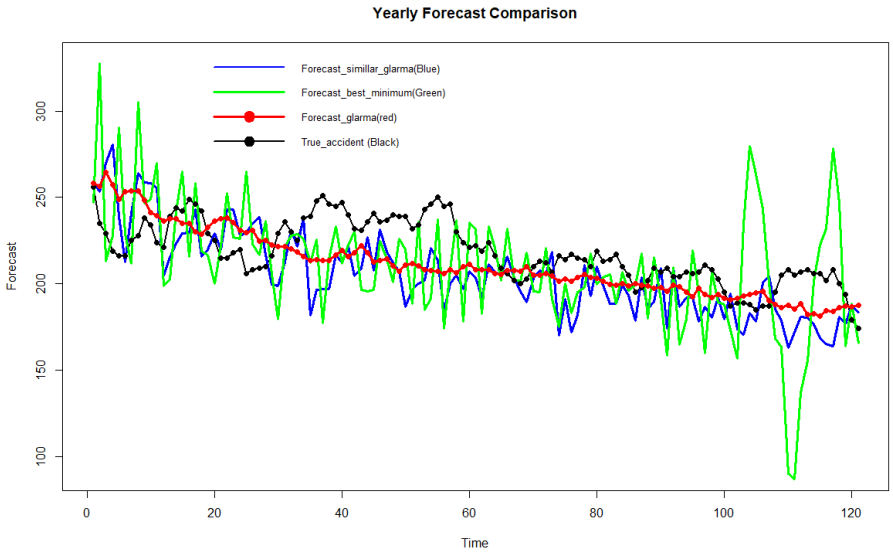

Figure 10: The figure presented above illustrates the yearly forecast results of three models "GLARMA", "similarGLARMA" and best minimum, which are utilized to predict the total number of car accidents. The x-axis represents the months, while the y-axis represents the forecasted total number of accidents. The black line represents the actual number of accidents, where each black point corresponds to the sum of true accidents within a year. The red line corresponds to the forecast generated by the GLARMA model, the blue line represents the forecast from the similarGLARMA model, and the green line indicates the forecast from the best minimum model. Moving along the x-axis, each step signifies one month, and the accumulation of the 12-month forecast results is represented by a single point on the graph.

6 Conclusion

In summary, this study utilized the Glarma package to explain the GLARMA modeling framework and apply it to forecast future car accidents in Finland. Additionally, compare the forecast result with the other two ARMAX models. The GLARMA model was chosen due to its ability to handle count data with discrete. The model was fitted with the same autoregressive and a moving average order of 7 and 11, using the negative binomial distribution. The model's diagnostics results, including score residuals, PIT histogram, ACF, PACF, AIC(1474.3), and RMSE (18.8) suggest that the GLARMA model outperforms the "similar GLARMA" and "best minimum" ARMAX models, as indicated by its lower AIC value of 1468.9 compared to 1497.9 and 1510.8, respectively. Moreover, the absence of significant autocorrelation in the residuals suggests that the model is able to account for serial correlation, and there are no discernible patterns in the residuals. The L-Jung box test further supports that the residuals are white noise. Forecast data reveal that accidents tend to peak during the summer months, from

June to August, and decrease during the winter months, from December to February. The summer months report a higher total number of accidents, attributed to increased traffic volume due to the holiday season. It is reasonable to argue that winter tires could be the key reason that could potentially lead to a reduction in the number of car accidents during the winter. Overall, the GLARMA model provides comparatively better performance for forecasting car accidents in Finland. The findings of our study can have implications for policymakers in formulating effective road accident policies, as well as for insurance companies in predicting accidents and evaluating the likelihood of future non-life insurance claims.

Acknowledgement

I am grateful to Prof. Märt Möls, my supervisor, for dedicating his time, providing valuable support, and giving insightful suggestions throughout my thesis. Additionally, I extend my thanks to our program director Prof. Meelis Käärik.

References

[1] Nabilah Filzah Mohd Radzuan, Zalinda Othman, and Azuraliza Abu Bakar. Analysis of uncertainty in time series data: Issues and challenges. In *Proc. the Asian Conference on Technology, Information & Society 2014*, pages 13–24, 2014.

[2] Mehmet Sari, A Sevtap Selcuk, Celal Karpuz, and H Sebnem B Duzgun. Stochastic modeling of accident risks associated with an underground coal mine in turkey. *Safety science*, 47(1):78–87, 2009.

[3] World Health Organization. Violence, Injury Prevention, and World Health Organization. *Global status report on road safety: time for action*. World Health Organization, 2009.

[4] Yi Zhang. *Count data time series models and their applications*. Missouri University of Science and Technology, 2021.

[5] William TM Dunsmuir and David J Scott. The glarma package for observation-driven time series regression of counts. *Journal of Statistical Software*, 67:1–36, 2015.

[6] Richard A Davis, Scott H Holan, Robert Lund, and Nalini Ravishanker. *Handbook of discrete-valued time series*. CRC Press, 2016.

[7] Michael A Benjamin, Robert A Rigby, and D Mikis Stasinopoulos. Generalized autoregressive moving average models. *Journal of the American Statistical association*, 98(461):214–223, 2003.

[8] William TM Dunsmuir, Cenanning Li, MASS Imports, and Suggests RUnit. Package 'glarma'. 2018.

[9] Tatiana Petukhova, Davor Ojkic, Beverly McEwen, Rob Deardon, and Zvonimir Poljak. Assessment of autoregressive integrated moving average (arima), generalized linear autoregressive moving average (glarma), and random forest (rf) time series regression models for predicting influenza a virus frequency in swine in ontario, canada. *PloS one*, 13(6):e0198313, 2018.

[10] Josephine Dufitinema. Forecasting the finnish house price returns and volatility: a comparison of time series models. *International Journal of Housing Markets and Analysis*, 15(1):165–187, 2021.

[11] Sittipan SITTIKARIYA, Venky N SHANKAR, Ming-Bang Shyu, and Songrit CHAYANAN. Accounting for serial correlation in count models of traffic safety. *Journal of the Eastern Asia Society for Transportation Studies*, 6:3645–3657, 2005.

[12] Francoise Seillier-Moiseiwitsch. Sequential probability forecasts and the probability integral transform. *International Statistical Review/Revue Internationale de Statistique*, pages 395–408, 1993.

[13] Keya Rani Das and AHMR Imon. A brief review of tests for normality. *American Journal of Theoretical and Applied Statistics*, 5(1):5–12, 2016.

[14] 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)*, pages 1–8. IEEE, 2012.

[15] Imoh Udo Moffat, Emmanuel Alphonsus Akpan, et al. White noise analysis: a measure of time series model adequacy. *Applied Mathematics*, 10(11):989, 2019.

[16] Riikka Rajamäki. Kesärenkaiden urasyvyys ja onnettomuusriski. *Helsinki: Edita Prima Oy. Saatavissa: http://www. vtt. fi/inf/pdf/tiedotteet/2010*, 2525, 2010.

[17] Yung-Ching Hsu, Pai-Lung Chou, and Yung-Ming Shiu. An examination of the relationship between vehicle insurance purchase and the frequency of accidents. *Asia Pacific management review*, 21(4):231–238, 2016.

[18] Dominique Estival and Françoise Gayral. An nlp approach to a specific type of texts: Car accident reports. *arXiv preprint cmp-lg/9502032*, 1995.

[19] Sabine Gebert-Persson, Mikael Gidhagen, James E Sallis, and Heléne Lundberg. Online insurance claims: when more than trust matters. *International Journal of Bank Marketing*, 37(2):579–594, 2019.

[20] Peng Shi, Xiaoping Feng, and Anastasia Ivantsova. Dependent frequency–severity modeling of insurance claims. *Insurance: Mathematics and Economics*, 64:417–428, 2015.

Appendix1

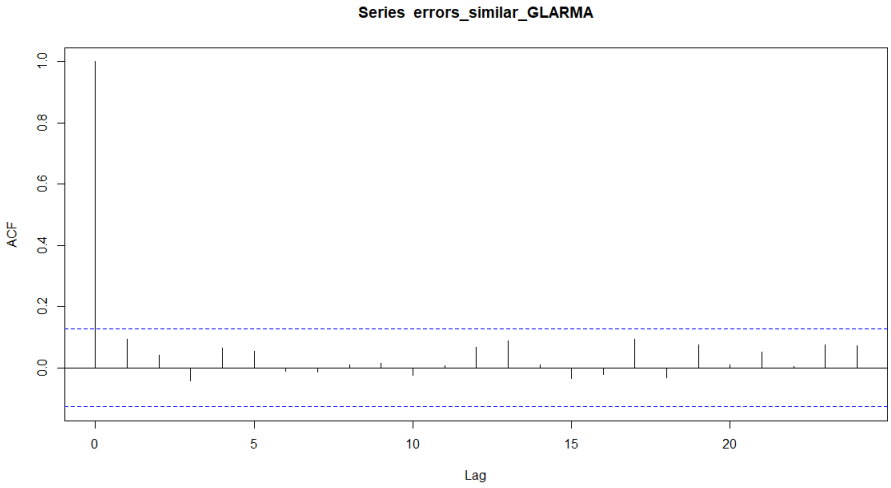

Figure 11: ACF of model exactly similar parameter as our GLARMA model

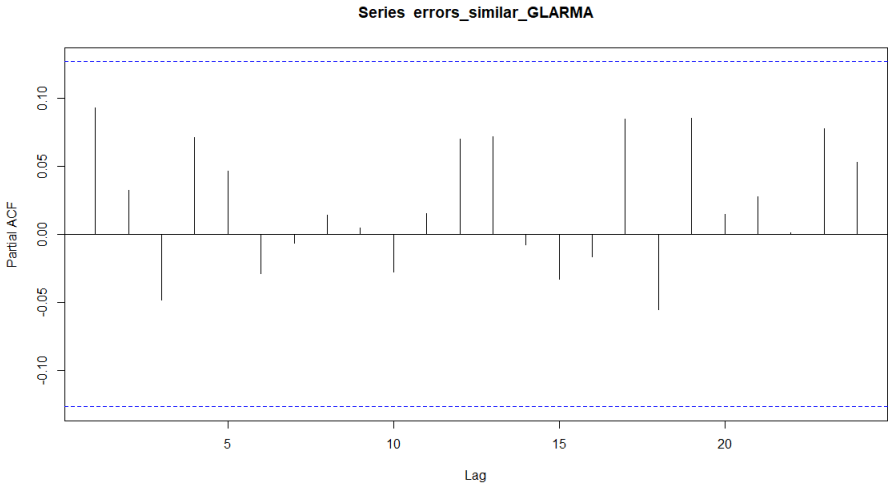

Figure 12: PACF of the model exactly similar parameter as our GLARMA model

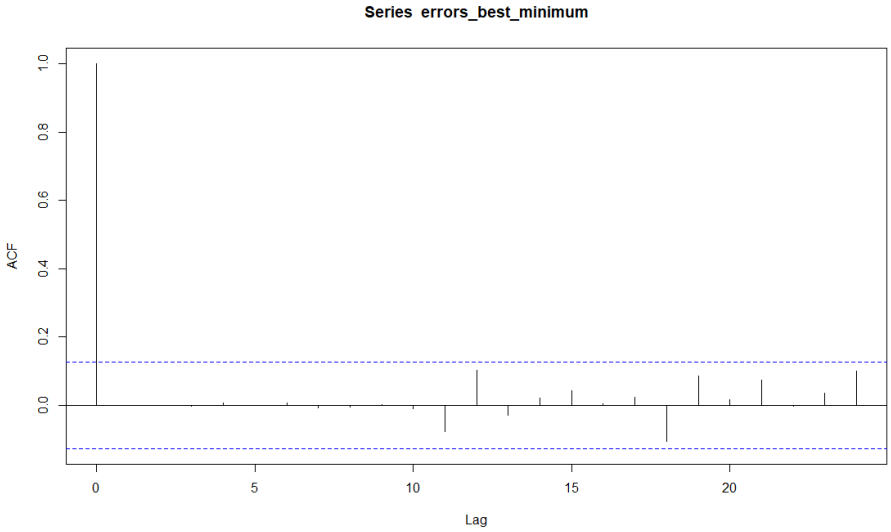

Figure 13: ACF of the best model with minimum lags possible

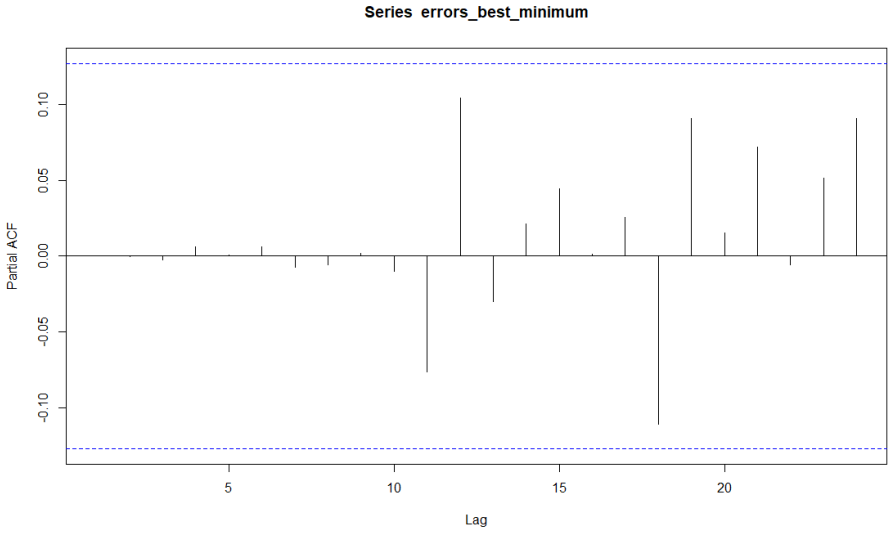

Figure 14: ACF of best model with minimum lags possible

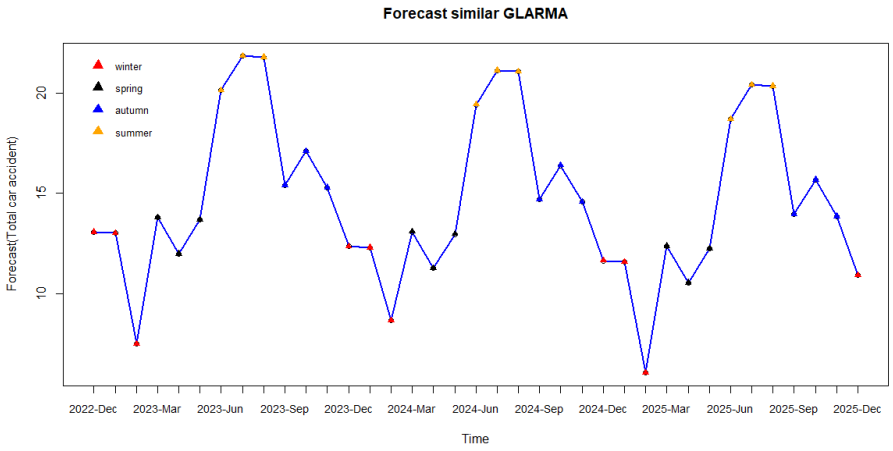

Figure 15: Forecast of the ARMAX model exactly similar to the GLARMA model

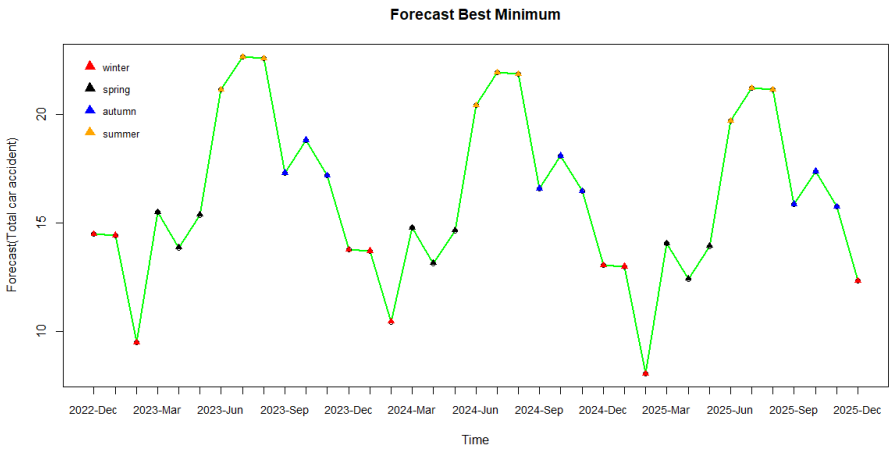

Figure 16: Forecast of ARMAX model with best minimum lags

Appendix2

```
#########################################################
#Rcode*****************************************************
library(glarma)
library(ggplot2)
data_new=read.csv("FinnishAccidents.csv",na.strings = "..")
#View(data_new)
head(data_new)
data_new1=data_new[0:239,]
View(data_new1)
head(data_new)
summary(data_new)
str(data_new)
# Create an empty data frame
df <- data.frame(month = rep(month.name,120), year= rep
    (2003:2022, each = 12), days = numeric(120))
# Iterate through the years and months and get the number of
    days
counter = 1
for(y in 2003:2022){
  for(i in 1:12) {
    start_date <- as.Date(paste0(y,"-", i, "-01"))
    if (i == 2) {
      if (y %% 4 == 0 && (y %% 100 != 0 || y %% 400 == 0)) {
        end_date <- as.Date(paste0(y,"-", i, "-29"))
      } else {
        end_date <- as.Date(paste0(y,"-", i, "-28"))
      }
    } else if (i %in% c(4,6,9,11)) {
      end_date <- as.Date(paste0(y,"-", i, "-30"))
    } else {
      end_date <- as.Date(paste0(y,"-", i, "-31"))
    }
    df$days[counter] <- as.numeric(end_date - start_date) + 1
    df$year[counter] <- y
    df$month[counter] <- month.name[i]
    counter <- counter + 1
  }
}
# Print the data frame
print(df)
head(df)
View(df)
new_df<-df[0:239,2:3]
head(new_df)
accident_data <- cbind(data_new1, new_df)
head(accident_data)
View(accident_data)
```

```
boxplot(accident_data$Total)
boxplot(accident_data$speed.limit..50km.h.or.less)
boxplot(accident_data$year)
boxplot(accident_data$days)
str(accident_data)
# Print the new data frame
#View(accident_data)
log_days=log(accident_data["days"])
log_days<-log_days[0:239,]
length(log_days)

accident_data <- cbind(accident_data, log_days)
head(accident_data)
summary(accident_data)
month <- gsub("^\\d+M(\\d+)$", "\\1", accident_data$X)
month
# Convert month to numeric and replace non-numeric values with
    NA
month <- as.numeric(month, na.rm = TRUE)

#(First trial)Add a new column "Season" with 0 for spring and
    1 for winter
accident_data$winter <- ifelse(month %in% c(1:2, 12), 1, 0)
accident_data$autumn <- ifelse(month %in% c(9:11), 1, 0)
accident_data$spring <- ifelse(month %in% c(3:5), 1, 0)
#****
head(accident_data)
View(accident_data)
#accident_data
############################## Fitting trend line
obs <- ts(accident_data$Total, start = c(2003, 10), deltat = 1
    /12)
plot(obs)
F1=1:length(obs)
F2=log(F1)
#######################################################
# Add a new column "Season" with 1 for September and October,
    and 1 for November, December, January,
#February, and March. All other months will have 0.
#accident_data$Season <- ifelse(month %in% c(9, 10), 1, ifelse
    (month %in% c(11, 12, 1:3), 1, 0))
#table(accident_data$Season, month)
# Print the updated data frame
print(accident_data)
head(accident_data1)
View(accident_data1)

boxplot(accident_data[2])
head(accident_data1)
y <- accident_data[, 2] #predictive variable
```

```
y1 <- accident_data1[, 2]
head(y1)
head(y)
#accident_data1 <- cbind(accident_data1,1)
months_number<-1:239 # inclusion of month number
accident_data1<-cbind(accident_data1,months_number)
#accident_data1<-cbind(accident_data1,1)
head(accident_data1)
View(accident_data1)
#accident_data1<-cbind(accident_data1,1) # adding intercept
#colnames(accident_data1)[6]="Intercept"
#colnames(accident_data1)#********************
#View(accident_data1)
X1 <- as.matrix(accident_data1[,c(6,7,8,9,12)])# matrix of the
    dependent variable
#X1 <- as.matrix(accident_data1[,c(6,7,8,9,12,13)])
View(X1)
X2<-cbind(X1,1) # adding intercept
colnames(X2)[6]="Intercept"
colnames(X2)
#X <- as.matrix(cbind(accident_data[,c(4,6,7,8)], accident_
    data[,7]+accident_data[,8]))# matrix
#of the dependent variable
#View(X2)
X2
head(X2)
#summary(accident_data)# log of days does not work well(year
    not significant while using)
head(accident_data1[,c(6,7,8,9,12)])
#head(X2)
#View(X2)
length(X)
length(y)
head(y)
#Glarma model
head(X2)
colnames(X2)[6]="Intercept"
glarma_model1.2 <- glarma(y1,X2 ,phiLags=c(7,11), type = "
    NegBin",method = "FS",residuals = "Score",
alphaInit = 0, maxit = 30, grad = 1e-6)
summary(glarma_model1.2)

#windows(width=9,height=6)
par(mfrow = c(2,2))
set.seed(321)
plot(glarma_model1.2)
plot(glarma_model1.2,which=7:10)
###**************************************Box test************
    **********************
library(stats)
```

```
library(zoo)

rt <- normRandPIT(glarma_model1.2)$rt
Box.test(rt, lag = 25, type = "Ljung-Box", fitdf = 0)#p-value
    = 0.8523


####************************************************************
par(mfrow = c(1,1))
plot(obs, ylab = "Total accident",lty = 2, main = "Observed Vs
    GLARMA")
lines(fitted, lwd = 2,col="red")
##************************************************************
par(mfrow = c(1,1))
plot(obs, ylab = "Total accident",lwd=2, main = "Total car
    accident",col="black")
points(obs,col="red",pch = 19)
#*************************************Forecast************
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
library(zoo)
#####################################Generate New Data
    ##############################
# Create an empty data frame
df1 <- data.frame(month = rep(month.name,12), year= rep
    (2022:2025, each = 12), days = numeric(48))
# Iterate through the years and months and get the number of
    days
counter = 1
for(y in 2022:2025){
  for(i in 1:12) {
    start_date <- as.Date(paste0(y,"-", i, "-01"))
    if (i == 2) {
      if (y %% 4 == 0 && (y %% 100 != 0 || y %% 400 == 0)) {
        end_date <- as.Date(paste0(y,"-", i, "-29"))
      } else {
        end_date <- as.Date(paste0(y,"-", i, "-28"))
      }
    } else if (i %in% c(4,6,9,11)) {
      end_date <- as.Date(paste0(y,"-", i, "-30"))
    } else {
      end_date <- as.Date(paste0(y,"-", i, "-31"))
    }
    df1$days[counter] <- as.numeric(end_date - start_date) + 1
    df1$year[counter] <- y
    df1$month[counter] <- month.name[i]
    counter <- counter + 1
  }
}
# Print the data frame
print(df1)
```

```
head(df1)
#View(df1)
new_df1<-df1[12:48,]
View(new_df1)
#head(new_df1) # number of days
accident_data4 <- new_df1
head(accident_data4)
View(accident_data4)
str(accident_data4)
# Print the new data frame
#View(accident_data)
log_days4=log(accident_data4["days"])
log_days4<-log_days4[1:37,]
length(log_days4)
accident_data4 <- cbind(accident_data4, log_days4)
head(accident_data4)
View(accident_data4)
month <- gsub("^\\d+M(\\d+)$", "\\1", accident_data4$month)
month
# Convert month to numeric and replace non-numeric values with
    NA
#month <- as.numeric(month, na.rm = TRUE)
accident_data4$month <- tolower(accident_data4$month)
# Define a function to map month names to corresponding
    numeric values
month_to_number <- function(month) {
  months <- c("january", "february", "march", "april", "may",
      "june",
              "july", "august", "september", "october", "
                  november", "december")
  match(month, months)
}
# Convert month to numeric
month <- sapply(accident_data4$month, month_to_number)
#accident_data4$month <- as.numeric(accident_data4$month, na.
    rm = TRUE)
#Add a new column "Season" with 0 for spring and 1 for winter
accident_data4$winter <- ifelse(month %in% c(1:2, 12), 1, 0)
accident_data4$autumn <- ifelse(month %in% c(9:11), 1, 0)
accident_data4$spring <- ifelse(month %in% c(3:5), 1, 0)
#****
head(accident_data4)
View(accident_data4)
#accident_data
month_number4<-240:276
month_number4
# inclusion of month number
View(accident_data4)
accident_data4<-cbind(accident_data4,month_number4)
head(accident_data4)
```

```
View(accident_data4)
accident_data4<-cbind(accident_data4,1)
colnames(accident_data4)[9]="Intercept"
colnames(accident_data4)
View(accident_data4)
X4 <- as.matrix(accident_data4[,c(4,5,6,7,8,9)])# matrix of
    new independent variable
#View(X4)
########################################End new data generate
    ########
########################***Monthy forecast GLARMA****
    ################
#View(X4)
XT1 <- as.matrix(X4[1:36,])
colnames(XT1)[6]="Intercept"
forecast_glarma5 <- forecast.glarma(glarma_model1.2, n.ahead
    =36, newdata=XT1,newoffset = rep(0,36))
forecast_glarma5
print(forecast_glarma5$mu)
glarma_forecast<-forecast_glarma5$mu
print(glarma_forecast)
#################################################################
#*****************************Train Test Model validation**
##############################%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
##################forecast one year ahead for GLARMA Model#
True_accident <- vector()
length(seq(108, 239-11, by = 1))
prediction_start_vector=seq(108, 239-11, by = 1)
True_accident <- rep(NA, length(prediction_start_vector))
Forecast_accident_glarma <- rep(NA, length(prediction_start_
    vector))
for (i in 1:length(prediction_start_vector)){
  new_element <- sum(accident_data1[prediction_start_vector[i
      ]:(prediction_start_vector[i] + 11),2])
  #  True_accident <- c(True_accident, new_element)
  True_accident[i] <-  new_element
  model1.2<-glarma(y1[1:(106+i)],X2[1:(106+i),] ,phiLags=c
      (7,11), type = "NegBin",method = "FS",
  residuals = "Score",alphaInit = 0, maxit = 30, grad = 1e-6)
  allX1<-X2[(i+107):(i+107+11),]
  mu<-forecast.glarma(model1.2, n.ahead=12, newdata=allX1,
      newoffset = rep(0,12))$mu
  #1:(prediction_start_vector[i]-1)
  #prediction_start_vector[i]:(prediction_start_vector[i] +
      11)
  Forecast_accident_glarma[i]<-sum(mu)
}
print(Forecast_accident_glarma)
print(True_accident)
plot(True_accident)
```

```
line(Forecast_accident_glarma)
##################################End yearly forecast for
    glarma########
#############################################################
##################################Forecast yearly for simillar
    GLARMA####
View(accident_data4)
View(accident_data1)
True_accident <- vector()
length(seq(108, 239-11, by = 1))
prediction_start_vector=seq(108, 239-11, by = 1)
True_accident <- rep(NA, length(prediction_start_vector))
Forecast_simillar_glarma <- rep(NA, length(prediction_start_
    vector))
for (i in 1:length(prediction_start_vector)){
  # i=2
  new_element <- sum(accident_data1[prediction_start_vector[i
      ]:(prediction_start_vector[i] + 11),2])
  #  True_accident <- c(True_accident, new_element)
  True_accident[i] <-  new_element
  F6<-cbind(accident_data1[,c(6,7,8,9,12)],1)
  simillar_glarma1.3<-arima(accident_data1$Total[1:(106+i)],
      fixed=c(0,0,0,0,0,0,NA,0,0,0,NA,NA,NA,NA,NA,NA,NA),
      order=c(11,0,0), optim.control=list(maxit=1000),
      transform.pars = FALSE,xreg=F6[1:(106+i),],include.mean=
      FALSE)
  new_data <- cbind(as.matrix(accident_data1[(i+107):(i
      +107+11),c(6,7,8,9,12)]),1)
  prediction<-predict(simillar_glarma1.3, newxreg = new_data,h
      =1)$pred
  Forecast_simillar_glarma[i]<-sum(prediction)
}
print(Forecast_simillar_glarma)
#################################
#################################forecast Best minimum
    ###################################
True_accident <- vector()
length(seq(108, 239-11, by = 1))
prediction_start_vector=seq(108, 239-11, by = 1)
True_accident <- rep(NA, length(prediction_start_vector))
Forecast_best_minimum <- rep(NA, length(prediction_start_
    vector))
for (i in 1:length(prediction_start_vector)){
  new_element <- sum(accident_data1[prediction_start_vector[i
      ]:(prediction_start_vector[i] + 11),2])
  #  True_accident <- c(True_accident, new_element)
  True_accident[i] <-  new_element
  #F5
  #head(F5)
  simillar_glarma1.4<-arima(accident_data1$Total[1:(106+i)],
```

```
      order=c(10,0,5), optim.control=list(maxit=1000),xreg=F6
      [1:(106+i),],include.mean=FALSE)
  #allX1<-X2[(i+107):(i+107+11),]
  new_data <- cbind(as.matrix(accident_data1[(i+107):(i
      +107+11),c(6,7,8,9,12)]),1)
  prediction1<-predict(simillar_glarma1.4, newxreg = new_data,
      h=1)$pred
  #1:(prediction_start_vector[i]-1)
  #prediction_start_vector[i]:(prediction_start_vector[i] +
      11)
  Forecast_best_minimum[i]<-sum(prediction1)
}
print(Forecast_best_minimum)

##############################True_calculation_simpler##########
#True_accident <- vector()
#for (i in seq(2, length(True) - 10, by = 12)) {
#  new_element <- sum(True[i:(i + 11)])
#  True_accident <- c(True_accident, new_element)
#}
#print(True_accident)
#####################################################################
#####################################################################
##################################%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#####################**********Compare with the ARMAX
    model*****
head(accident_data1[,c(6,7,8,9,12)])
F5<-cbind(accident_data1[,c(6,7,8,9,12)],1)
# include the intercept line
head(F5)
dim(F5)
#p=F5[1:239,]
#p=accident_data1[1:239,nrow=6]
#head(p)
#dim(p)
#length(accident_data1$Total)
#trend2=arima(accident_data1$Total,xreg=F5,include.mean=FALSE)
    #,optim.control=list(maxit=1000))
#res2=residuals(trend2)
#plot(res2)
#acf(res1,30)
#pacf(res1,30)
#trend2_curve=(accident_data1$Total)-res2
#trend2_curve
#plot(trend2_curve)
#TSGraphs = function(series, lags=30 ){
#  layout (1:3)
#  par(mar = c(3, 4, 1, 1))
#  plot(series)
#  acf(series,lags)
```

```
#   pacf(series,lags)
#   layout(1)
#}
#Z1 = ts(trend2_curve,start=c(2003,1),frequency=12)
#TSGraphs(trend2_curve)
#TSGraphs(diff(trend2_curve))
#trend2_curve=Z1
#fixed=c(0,0,0,0,0,0,0,NA,0,0,0,NA,NA)
#*****************************ARMAX model comparision********
    *************************
#####################################################################
head(accident_data1[,c(6,7,8,9,12)])
F5<-cbind(accident_data1[,c(6,7,8,9,12)],1)
# include the intercept line
head(F5)
dim(F5)
#####################################################################
#*****************************ARMAX with Same lags same
    condition as glarma model*****************************
library(forecast)
m1.10<-arima(accident_data1$Total,fixed=c(0,0,0,0,0,0,NA
    ,0,0,0,NA,NA,NA,NA,NA,NA,NA), order=c(11,0,0), optim.
    control=list(maxit=1000),transform.pars = FALSE,xreg=F5,
    include.mean=FALSE)
#m1.10<-arima(accident_data1$Total,fixed=c(0,0,0,0,0,0,NA
    ,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA), order=c(25,0,0), optim.control=list(maxit
    =1000),transform.pars = FALSE,xreg=F5,include.mean=FALSE)
m1.10
AIC(m1.10)
errors_similar_GLARMA = residuals(m1.10)
acf(errors_similar_GLARMA,24)
pacf(errors_similar_GLARMA,24)
head(accident_data4[,c(4,5,6,7,8,9)])
new_data <- data.frame(F5 = as.matrix(accident_data4[,c
    (4,5,6,7,8,9)]))
#head(new_data)
# Generate a forecast using the ARMAX model and the new data
peediction11 <- predict(m1.10, newxreg = new_data,h=1)
pred1=peediction11$pred
pred1
#Box.test
Box.test(errors_similar_GLARMA,lag=20,type="Ljung-Box",fitdf =
    2)
#*****************************************************************
    **
#*************************best ARMAX model with the minimum
    lags***************
m1.2 <- arima(accident_data1$Total, order=c(11,0,5), optim.
    control=list(maxit=1000),xreg=F5,include.mean=FALSE)
```

```
errors_best_minimum= residuals(m1.2)
acf(errors_best_minimum,24)
pacf(errors_best_minimum,24)
AIC(m1.2)
predict2=predict(m1.2,newxreg = new_data)
pred2=predict2$pred
pred2
#Box.test
Box.test(errors_best_minimum,lag=20,type="Ljung-Box",fitdf =
    2)
#################################################################
#**************************************RMSE*****************
length(True_accident)
length(Forecast_accident_glarma)
mean(True_accident)
mean(Forecast_accident_glarma)
# Compute the RMSE of your same parameter of glarma model
rmse1 <-sqrt(mean((True_accident - Forecast_simillar_glarma)
    ^2))
print(rmse1)#25.79226

# compute RMSE of best model with minimum lags
rmse2 <- sqrt(mean((True_accident - Forecast_best_minimum)^2))
print(rmse2)#36.06311

#Compute the RMSE of our glarma model
rmse3 <- sqrt(mean((True_accident - Forecast_accident_glarma)
    ^2))
print(rmse3)#18.81097

#*************************************************************
#************************Monthly Forecast comparison plots of
     three model *************
#*************************************************************
pred1<-ts(pred1,start = c(2022,12), deltat = 1/12)
pred1
pred2<-ts(pred2,start = c(2022,12), deltat = 1/12)
pred2
glarma_forecast<-ts(glarma_forecast,start = c(2022,12), deltat
     = 1/12)
glarma_forecast
Total_accident<-ts(accident_data$Total,start = c(2003,1),
    deltat = 1/12)
#index <- as.yearmon(index(pred1))

################**********************
plot(Total_accident, xlim=c(2003,2026),ylab="Total accident",
    type = "l", lty = 1, col =
"black", lwd = 2,Main="Forecast comparision")
lines(x = pred1, type = "l", lty = 1, col = "blue", lwd = 3)
```

```
lines(x = pred2, type = "l", lty = 1, col = "green", lwd = 3)
lines(x = glarma_forecast, type = "l", lty = 1, col = "red",
    lwd = 3)
points(Total_accident, pch = 19, col = "black")

title(main = "Monthly Forecast GLARMA Models")

for (i in seq_along(index)[-1]) {
  grid(col = "gray", lty = "dotted")
  x1 <- as.numeric(index[i - 1])
  x2 <- as.numeric(index[i])
  y1 <- par("usr")[3]
  y2 <- par("usr")[4]
  lines(c(x1, x2), c(y1, y2), col = "gray", lty = "dotted")
}

legend("topright",legend("topright",
       legend = c("Similar GLARMA (Blue)", "Best Minimum (
           Green)","GLARMA(red)", "Observation (Black)"),
       legend = c("Similar GLARMA (Blue)", "Best Minimum (
           Green)","GLARMA(red)", "Observation (Black)"),
       col = c("blue", "green","red", "black"),
       lty = 1, pch = c(NA,NA,NA,19), pt.cex = 1.8,
       cex = 0.8, x.intersp = 0.5,
       xjust = 2, yjust =2, bty = "n",
       lwd = c(2, 3, 3))


##########################################################

##################plot of the montly forecast comparision
    ###########################################

#start_date <- as.Date("2012-01-01")
#end_date <- as.Date("2023-05-28")
#date_sequence <- seq(start_date, end_date, by = "month")
#index <- as.Date(index)

plot(True_accident,xlab="Time (month)",ylab="Forecast",ylim=c
    (85,330), type = "l", lty = 1, col = "black",
lwd = 2)
lines(x = Forecast_simillar_glarma, type = "l", lty = 1, col =
     "blue", lwd = 3)
lines(x = Forecast_best_minimum, type = "l", lty = 1, col = "
    green", lwd = 3)
lines(x = Forecast_accident_glarma, type = "l", lty = 1, col =
     "red", lwd = 3)
points(True_accident, pch = 19, col = "black")
points(Forecast_accident_glarma, pch = 19, col = "red")
```

```
#axis(1, at = index, labels = format(index, "%Y-%b"), cex.axis
    = 0.9, las = 2)

title(main = "yearly Forecast Comparison")
legend("topleft",
       legend = c("Forecast simillar glarma(Blue)", "Forecast
           best minimum(Green)",
       "Forecast glarma(red)", "True accident (Black)"),
       col = c("blue", "green","red", "black"),
       lty = 1, pch = c(NA,NA,19,19), pt.cex = 1.8,
       cex = 0.8, x.intersp = 0.5,
       xjust = 0.5, yjust =0.5, bty = "n",
       lwd = c(2, 3, 3))
#
   ######################################################################


#########################%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#**********************plotting the Forecast with points in
    seasons **************************
#########################**************
    ############################
#*******************************Forecast season GLARMA********
    *******************************
#install.packages("scales")
library(scales)
library(zoo)
glarma_forecast<-ts(glarma_forecast,start = c(2022,12), deltat
    = 1/12)
glarma_forecast
library(scales)
# Convert index to year-month format
index <- as.yearmon(index(glarma_forecast))

# Plot forecast with months in X-axis
plot(glarma_forecast, ylab = "Forecast(Total car accident)",
    lty = 1,
     main = "Forecast GLARMA", xaxt = "n")
lines(glarma_forecast, col = "red",lwd = 2)
points(glarma_forecast)

# Add X-axis with months and years
#axis(1, at = index, labels = format(index, "%Y"), cex.axis =
    0.9)
axis(1, at = index, labels = format(index, "%Y-%b"), cex.axis
    = 0.9)
# Add diagonal lines to X-axis
#for (i in seq_along(index)[-1]) {
#  grid(col = "gray", lty = "dotted")
```

```
#  x1 <- as.numeric(index[i - 1])
#  x2 <- as.numeric(index[i])
#  y1 <- par("usr")[3]
#  y2 <- par("usr")[4]
#  lines(c(x1, x2), c(y1, y2), col = "gray", lty = "dotted")
#}


winter_month <- c("December", "January", "February")
spring_month <- c("March","May","April")
autumn_month <- c("September", "October","November")
summer_month <- c("June", "July", "August")

points(glarma_forecast,
        pch = ifelse(format(index, "%B") %in% winter_month, 17,
                     ifelse(format(index, "%B") %in% spring_
                         month, 17,
                            ifelse(format(index, "%B") %in%
                                autumn_month, 17,
                                   ifelse(format(index, "%B") %
                                       in% summer_month, 17, 1)
                                       ))),
        col = ifelse(format(index, "%B") %in% winter_month, "
            red",
                     ifelse(format(index, "%B") %in% spring_
                         month, "black",
                            ifelse(format(index, "%B") %in%
                                autumn_month, "blue", "orange")
                                )),
        bg = ifelse(format(index, "%B") %in% autumn_month, "
            blue", NA))


# Add legend for winter, spring, autumn, and summer points
legend(x = "topleft", y = 1, legend = c("winter", "spring", "
    autumn", "summer"),
        pch = c(17, 17, 17, 17),
        col = c("red", "black", "blue", "orange"),
        pt.cex = 1.5, cex = 0.8, x.intersp = 0.5, bty = "n")
##%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%
#******************forecast seasons simillar glarma**********
    *
pred1<-ts(pred1,start = c(2022,12), deltat = 1/12)

# Convert index to year-month format
index <- as.yearmon(index(pred1))

# Plot forecast with months in X-axis
plot(pred1, ylab = "Forecast(Total_car_accident)", lty = 2,
```

```
      main = "Forecast similar GLARMA", xaxt = "n")
lines(pred1, col = "blue",lwd = 2)
points(pred1)

# Add X-axis with months and years
axis(1, at = index, labels = format(index, "%Y-%b"), cex.axis
    = 0.9)

# Add winter and summer points
winter_month <- c("December", "January", "February")
spring_month <- c("March","May","April")
autumn_month <- c("September", "October","November")
summer_month <- c("June", "July", "August")

points(pred1,
       pch = ifelse(format(index, "%B") %in% winter_month, 17,
                    ifelse(format(index, "%B") %in% spring_
                        month, 17,
                           ifelse(format(index, "%B") %in%
                               autumn_month, 17,
                                  ifelse(format(index, "%B") %
                                      in% summer_month, 17, 1)
                                  ))),
       col = ifelse(format(index, "%B") %in% winter_month, "
           red",
                    ifelse(format(index, "%B") %in% spring_
                        month, "black",
                           ifelse(format(index, "%B") %in%
                               autumn_month, "blue", "orange")
                           )),
       bg = ifelse(format(index, "%B") %in% autumn_month, "
           blue", NA))


# Add legend for winter, spring, autumn, and summer points
legend(x = "topleft", y = 1, legend = c("winter", "spring", "
    autumn", "summer"),
       pch = c(17, 17, 17, 17),
       col = c("red", "black", "blue", "orange"),
       pt.cex = 1.5, cex = 0.8, x.intersp = 0.5, bty = "n")

#########################################################
#**********Forecast season Best Minimum**********
    **********
pred2<-ts(pred2,start = c(2022,12), deltat = 1/12)

# Convert index to year-month format
index <- as.yearmon(index(pred2))

# Plot forecast with months in X-axis
```

```
plot(pred2, ylab = "Forecast(Total car accident)", lty = 2,
     main = "Forecast Best Minimum", xaxt = "n")
lines(pred2, col = "green",lwd = 2)
points(pred2)

# Add X-axis with months and years
axis(1, at = index, labels = format(index, "%Y-%b"), cex.axis
    = 0.9)
# Add diagonal lines to X-axis


# Add winter and summer points
winter_month <- c("December", "January", "February")
spring_month <- c("March","May","April")
autumn_month <- c("September", "October","November")
summer_month <- c("June", "July", "August")

points(pred2,
       pch = ifelse(format(index, "%B") %in% winter_month, 17,
                    ifelse(format(index, "%B") %in% spring_
                        month, 17,
                           ifelse(format(index, "%B") %in%
                               autumn_month, 17,
                                  ifelse(format(index, "%B") %
                                      in% summer_month, 17, 1)
                                  ))),
       col = ifelse(format(index, "%B") %in% winter_month, "
           red",
                    ifelse(format(index, "%B") %in% spring_
                        month, "black",
                           ifelse(format(index, "%B") %in%
                               autumn_month, "blue", "orange")
                           )),
       bg = ifelse(format(index, "%B") %in% autumn_month, "
           blue", NA))


# Add legend for winter, spring, autumn, and summer points
legend(x = "topleft", y = 1, legend = c("winter", "spring", "
    autumn", "summer"),
       pch = c(17, 17, 17, 17),
       col = c("red", "black", "blue", "orange"),
       pt.cex = 1.5, cex = 0.8, x.intersp = 0.5, bty = "n")

#####################################################
```

Non-exclusive licence to reproduce thesis and make thesis public

I, Dhruba R. Gnawali, 1. herewith grant the University of Tartu a free permit (non-exclusive license) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, GLARMA time series modeling of counts, supervised by Märt Möls.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Dhruba R. Gnawali
06/06/2023