SF2943 - Time Series Analysis Project
YETFI Group

Yage Hao - 980707-9547 | Ellinor Kalderén - 960509-2148 | Thomas Lundqvist 970103-4473 | Ioannis Athanasiadis - 930817-4896

Part A -Time Series Analysis of the Climate

Problem statement

In the context of this project, we were asked to choose and analyse a non-financial time series of our choice. We found Climate Change to be quite an intriguing topic and decided to analyze a times series related to that.

Software

To make the analysis of this dataset, the programming language R was used. R is made for data science and as such have many built in functions and supporting libraries for time series analysis and prediction. The libraries used were:

- *itsmr*
- *forecast*

Functions that were invoked:

- *acf* : Calculates the Autocovariance Function
- *auto.arima* : Automatically select parameters and fit a time series model
- *BIC* : Calculates the BIC criteria
- *yw* : Estimates AR coefficients using Yule-Walker
- *arima* : Fits a model specified by the given order onto the provided data
- *forecast* : Applied a generated model on the provided input series
- *diff* : Returns the difference between two vectors

Description of the dataset

We found the dataset titled as Climate Change: Earth Surface Temperature Data provided by the kaggle website [1] aligned both with the scope of this project and our interests. More specifically, the dataset contains data related to the average earth's surface temperature separated into different countries, states and cities. Additionally, the overall average surface temperature, across the globe, was also provided and is the time series which we chose to analyze. More specifically, the average temperature of the earth's surface was measured 12 times a year, once a month from 1750 to 2015. The dataset also contained the measurements uncertainty within a 95 % confidence interval. Due to the large uncertainty in the early years

measurements, indicated by the broad range of variance, we decided to exclude the early years and base our analysis from the year 1875 onwards. In Figure 1 we display the histogram of temperature measurements of our time series. Finally we split the time series into training and test sets with the purpose of evaluating how well the resulting model of our analysis generalizes to unseen data. The last 5 years [2011-2015] were used to construct the test time series and the rest of them [1875-2010] were used for training purposes.

Modeling

Stationarity analysis

First we need to conclude whether our time series is stationary or not. For that purpose, we plotted the time series for the last 141 years [1875 - 2015] as seen in Figure 2. Based on that figure, it is evident that there is a trend in our time series. Additionally, we also plotted the last 50 years in Figure 3 where a seasonality factor can be observed. The existence of seasonality can be also verified by the ACF shown in Figure 4, since there is a repeated pattern of period 12 in the autocorrelation of the temperature values. This observation is intuitive since the temperature is changing with the seasons on earth. Built upon these, we can conclude that our time series is not stationary due to the trend and seasonality.

After having identified that our time series is not stationary we need to transform it into a stationary one. From the raw time series data we can see that there seems to be a seasonality with period 12. Thus, the seasonality can be removed by taking the difference between the time series with a lag of 12. The existing trend is also removed, which generates the time series shown in Figures 5 and 6, where the red line indicates the mean. Furthermore, the ACF is shown in Figure 7 for 200 lags. Since the ACF for most lags are within the appropriate region we conclude that the time series is stationary enough to begin further analysis.

The Figures 5, 6 and 7 indicate that the statistical properties of the new time series do not change over time and thus it is stationary.

Parameter selection and modeling

After having transformed the initial non-stationary time series into a stationary one, the next step is to identify from which kind of process it has been realized from. In the context of this course, we have been introduced to the ARIMA(p,d,q) family of processes which account for a different number of time series categories based on the autoregressive order (p), the moving average order (q) and the order of differences (d). Considering that our time series has been transformed into a stationary one, setting d to zero should be a feasible option, and thus we are left with deciding what kind of ARMA(p,q) process our time series is.

Provided that the ACF and PACF are indicative of both the process' category and its autoregressive and moving average order, we calculated the sample ACF and PACF of artificial

generated ARMA(p,q) process for different p and q values which correspond to either strictly AR, MA or a combination of AR and MA process.

Based on Figure 8, we can compare the behavior of ACF and PACF of our time series to the ones of an AR, MA and an ARMA process. The descriptive feature of an MA process is that its ACF reaches to zero in relatively few time steps, as dictated by its order, while it takes many more time steps for its PACF to reach values close to zero. On the other hand, the PACF of an AR process reaches zero relatively fast, according to its order, and it takes considerably more time steps for its ACF to reach zero. Finally, in the case of ARMA processes which consist of both MA and AR factors, both the ACF and the PACF need relatively a lot of time steps for them to stop being significant. Based on these observations our time series looks as being closer to an ARMA.

In order to find the ideal order of p and q we employ a parameter search with p ranging from 1 to 15 and q from 1 to 15 and choose the model that results in the lowest AIC, since this metric accounts for both complexity and performance of the model. The parameter searching resulted in p = 4 and q = 13 setup to be the ideal. The order of the ARMA model generated by this procedure is quite intuitive since one would assume the future surface temperature to be highly correlated with the ones of the previous months, however capitalizing on temperatures from months exceeding the year by a lot would seem to be quite inefficient considering the additional complexity. In other words, the information related to accurately predicting the future temperature of the surface seems to be highly enclosed in the temperature of the few previous months. When it comes to the order of MA contribution, it is also intuitive since the significance of the PACF of our time series seems to be decreasing from roughly 13 lags onwards.

Results and Forecasts

The generated ARMA(4,13) model results in identical MSRE in the training and the test splits indicate that the model can generalize in novel inputs relatively well. Additionally, according to the qualitative results shown in Figures 9 and 10, we can say that our model managed to sufficiently fit the time series. Finally the generated model was also used to forecast future values as shown in Figure 11. The parameters fitted for this model can be found in Table 1.

Alternative Approach

ANN

Generally, an Artificial Neural Networks(ANN)-based approach could have been an effective alternative, since Neural Networks (NN) have shown promising potential in a number of fields ranging from computer vision to natural language processing. In our case, a simple multi-layer perceptron could have been trained to predict future predictions given a number of past observations. If we had chosen this approach, we would have had to pay attention to avoid overfitting on the training data which would have resulted in lacking performance in unseen

data. To address the problem of overfitting we could employ some kind of regularization techniques e.g. weight decay to account for the models' complexity and consequently its variance.

Another approach was to use the auto.arima function to determine the most appropriate model, this is further discussed in the appendix.

Difficulties and Improvements

During the modeling procedure, we firstly remove its trend and seasonality and convert it into a stationary process. However, after we created several models and checked the distributions of their residuals, we found the ACF are not close to zero with a 5% confidence interval suggesting it is not a white noise. Such autocorrelation made it difficult to choose models and parameters as they all failed passing the Ljung-Box test. To further investigate and make a more accurate model, we may need to consider the ARCH or GARCH models.

Conclusion

We concluded that the ARMA(4,13) was the most appropriate choice of model for our time series, based on our parameter search. The model was able to match our test data to a satisfactory level and we are therefore confident in our future predictions.

Part B - A discussion of the paper Modeling electricity loads in California: ARMA models with hyperbolic noise

Review:

In this report we have summeriesed and discussed the paper Modeling electricity loads in California: ARMA models with hyperbolic noise. The need for forecasting the energy demands is essential in order to ensure that enough energy is produced to support our needs. Additionally, transmitting and storing energy is both time consuming and costly which further highlights the need for efficient energy's demand forecasting. In their work [4], Nowicka-Zagrajek et al. proposed an approach which employs the hyperbolic noise assumption to model the electricity load. The intuition behind modeling the time series with hyperbolic noise instead of the standard white noise was that the authors observed that the residuals, after having removed the trend and the seasonality, displayed short-range correlations which indicates that the noise was not white by any means. Additionally, they have also noted a heavy-tailed trend in the error distribution which is indicative of a hyperbolic origin.

In this work, the authors utilized the time series provided by the University of California Energy Institute (UCEI) in which the electricity loads were recorded hourly for the period April 1 - 1998 to December 31- 2000. In this time series, the daily cycle was quite evident and for that reason, the time series was only perceived as having been recorded daily by averaging across the day. However, averaging across the day was not enough to completely remove the seasonality since there was a quite evident monthly and yearly seasonality as indicated by the spectral density. More specifically, in the spectral density that the authors provide, there are evident peaks in the frequencies corresponding to the week and the year. Given that removing non-stationarity is challenging to do so in cases where only few data points are available, they considered only the last 2 years from the time series during their analysis in order to reduce the effect of the trend and seasonality.

Having these 2-years long times series, the trend was removed by computing the moving average, while the weekly trend was removed by employing a specifically crafted linear filter.

As previously stated, the residuals are most often considered to be Gaussian iid$(0, \sigma^2)$ distributed. After plotting the residuals in a normal probability plot it was concluded that the residuals did not fit the line for Gaussian iid residuals nor the curved one describing α- stable residuals. Instead it showed heavy tails and was best approximated as hyperbolic. This was motivated by plotting the empirical probability density function on a semi-logarithmic scale and the hyperbolic probability distribution function, which coefficients were estimated using the

maximum likelihood method. The Kolmogorov statistic was used for comparison and the hyperbolic distribution could not be rejected on a 1 % confidence level.

The best fitted model for the data was discovered to be an ARMA(1,6) model with an AICC of 1956.294. The goodness of fit for the model was determined using four different statistical methods to determine the randomness of the residuals: Portmanteau, Turning Point, Difference Sign and Rank. Neither of these test detect deviation from iid noise at a 5 % level.

The model was tested on two months of data, from 1st of January to the 28th of February, by making a day ahead prediction of the test data. The results of this prediction are then compared to the actual load of the system and the predictions made by California System Operator (CAISO). In a Mean Squared Error (MSE) sense the CAISO model was better, however the authors argue that this is not the best take away. It is evident that the MSE is high because of two predictions that are off in the first and second lag, which create outliers. In a Mean Absolute Error (MAE) sense the authors model outperforms the CAISO model as this error metric does not place as much weight on these outliers. The big outliers were that the ARMA(1,6) model incorrectly corresponds to American holidays New years eve and President day. When the model is tested on the period 3 January to the 28 of February both of the error metrics improve. They also fine tuned the model by applying it to an adaptive scheme, which slightly improved the error metrics even more.

Discussion:

The authors provided a well justified analysis related to the electricity load forecasting. More specifically they have successfully transformed the initial non-stationary time series into a stationary one as indicated by the sample AC and PAC function of the residuals. They provided reasonable arguments for using the ARMA family of processes as well as employing hyperbolic noise. Additionally, they also argued why using non-linear models such as ARCH or GARCH was out of the question. Furthermore, they successfully tested various numbers of different AR and MA orders and they evaluated them based on the AICC metric which is similar to the one we used in the PART-A but penalizes extensive overfitting more, compared to the base AIC.

When it comes to the generality of their approach, the authors tested their model on real data as well, and it turned out that their approach outperformed the official forecast operator of California (CAISO). This is a strong indication that their model managed to incorporate the relevant past information without overfitting.

In challenging problems like this one, one would spontaneously consider applying Artificial Neural Networks to model the problem. The authors provide a detailed summary of the ANNs' capabilities in modeling time series and they concluded that they often resulted in overfitted models.

To some extent using a time series of only a few steps might be somewhat problematic but in this case we did not consider it to be too big of a problem since they explicitly tried to model the

short-term forecasting. However using a bigger time series even for that would have been most likely beneficial.

When it comes to the weaknesses of the authors' analysis, we had trouble identifying any apart from the way the initial time series was transformed into a stationary one. We felt like their trend and seasonality removing approach might be too fine tuned for the specific portion of the time series and a significantly different stationarity transformation approach might be needed for a different part of the timeseries. Finally, omitting a portion of the available data can also be considered as a weakness since neglecting as much data as the 30% of the available one might be too much. Although including these data would result in a less stationary time series, it might have resulted in a more capable model due to fitting in a more difficult time series.

Modeling the loads in this way seems to have been successful in this case but whether this exact model can be applied today is questionable since the electricity loads keep increasing every year and depend on demographic and economic growth [3]. If it would be applied in order to forecast future loads one would most likely need to reevaluate the model.

Overall this was a well argued and well written paper, the author provided all the necessary details for one to replicate their analysis and on top of that, their model outperformed the well established CAISO when tested in novel data related to short-time electricity load forecasting.

Peer Review - Carbon Dioxide (CO2) concentration below the surface of the water in the Bay of Bengal

M. Ayed Tarek and M. El Mendili Mohammed

We are reviewing the paper: “Carbon Dioxide (CO2) concentration below the surface of the water in the Bay of Bengal" by M. Ayed Tarek and M. El Mendili Mohammed. In general, the report was clear and informative with plenty of graphs and good presentation of results and analysis. This gave the report an overall professional feel.

In detail, the report is well structured and composed of four parts which are introduction of the dataset, data exploration with stationarity analysis, model fitting and residual analysis part and forecasting. The dataset they used is a climate time series of Carbon Dioxide concentration, which is meaningful to study on for climate changes and global temperature increasing issues. In the data exploration part, they made comparisons of different differencing methods and how they influenced the stationarity. They finally chose an ARIMA model to fit their time series, which is straightforward for climate time series and has good interpretability. And as the figure in the forecast part shows, the model is significantly accurate.

However, we suppose the report can be further studied to some extent. For example, the QQ-plot of residual checking is not satisfactory to us as it shows heavy tails on both sides, which was commented on in their presentation.

Some of the required parts of the project are not shown. Alternative models and approaches were missing as well as possible difficulties that were encountered during the project. The final parameter of the time series, a description of the software, as well as the functions used to analyze the data were also missing.

Besides, some tiny improvements can be done on figure plotting. For example, some of the figures are missing x, and y labels, see for example figure 1. The thick line width also made it hard to see the data curves within clusters of data. This is more of a preference and is somewhat okay. In the end we feel that the project was very well executed and presented.

References

[1] Climate Change: Earth Surface Temperature Data. (2017, May 1). Kaggle. https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data

[2] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2nd ed.). OTexts.

[3] Kavalec, Chris, Nicholas Fugate, Tom Gorin, Bryan Alcorn, Mark Ciminelli, Asish Gautam, Glen Sharp, and Kate Sullivan. 2012. California Energy Demand Forecast 2012-2022 Volume 1: Statewide Electricity Demand and Methods, End-User Natural Gas Demand, and Energy Efficiency. California Energy Commission, Electricity Supply Analysis Division. Publication Number: CEC-200-2012-001-CMF-VI.

[4] Nowicka-Zagrajek, J. and Weron, R., 2021. Modeling electricity loads in California: ARMA models with hyperbolic noise.

Appendix

auto.arima()

We also tried to apply the auto.arima() function which can automatically fit a time series model based on AIC and BIC criterion regardless of stationarity. Around this function, we applied three different scenarios and reached three different models.

In the first case, we applied the auto.arima() function to our preprocessed stationary time series. The function automatically fits a MA(2) model. Then we did residual analysis to check if it is a white noise. Although in this case the p-value equals 0.1045 which is quite significant, the ACF plot shows a peak on lag 12 indicating some dependencies.

In the second case, we directly applied the auto.arima() function to our original time series which shows an increasing trend and an obvious seasonal performance. In this circumstances, the auto.arima() function actually fits a SARIMA (seasonal arima) model, i.e. ARIMA(3,0,0)(1,1,1)[12] which means the general part of the time series follows a AR(3) model while the seasonal part follows a ARIMA(1,1,1) model with lag 12. The results of residual check are not satisfied as well since the p-value is too small (less than 0.1%) and the ACF are not close to zero.

Inspired by the above two models, we manually choose the parameters p and q of the ARMA model and fitted an ARMA(4,1) model by using the arima() function. However, in this case, the model still failed to pass the Ljung-Box test as the ACF plot showed, but it provides a reasonable p-value, i.e. 0.042.

To further check and decide which model is better, we do forecasting using data of the last five years which we left for validation. As Figure 12 shows, only the ARIMA(3,0,0)(1,1,1)[12] provides a reasonable prediction and the plot between predicted values and actual values shows a linear trend which suggests our ARIMA model is good.

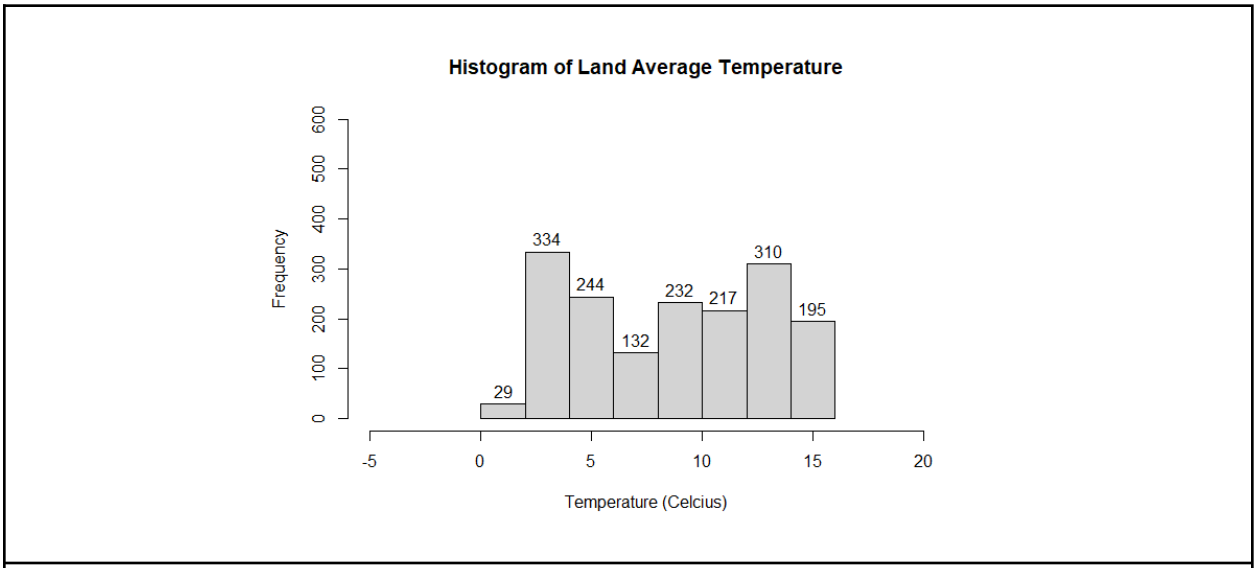


Figure 1. Surface temperature histogram of our time series

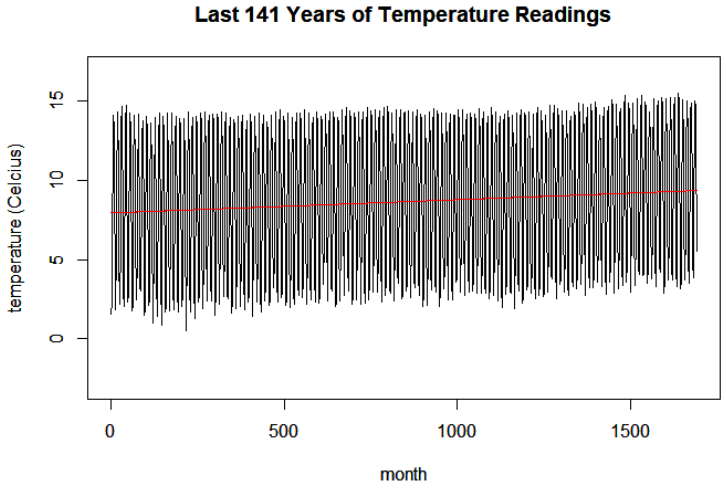


Figure 2. Trend (red) in the time series (black)

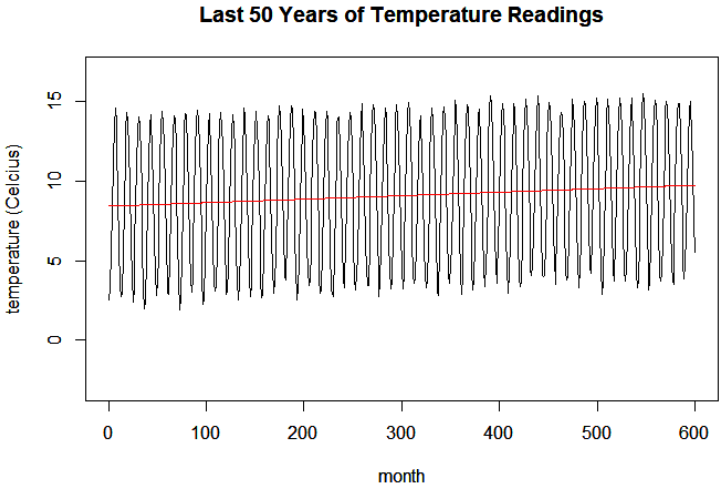


Figure 3. Seasonality in the time series (black)

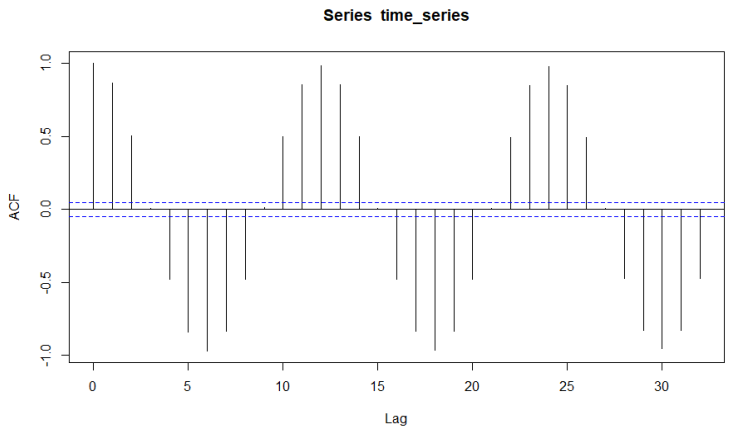


Figure 4. Auto-correlation function of the time series

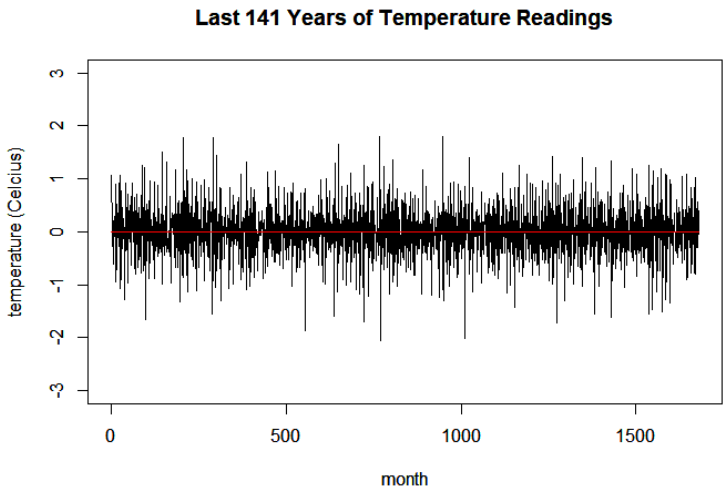


Figure 5. Trend (red) in the time series (black)

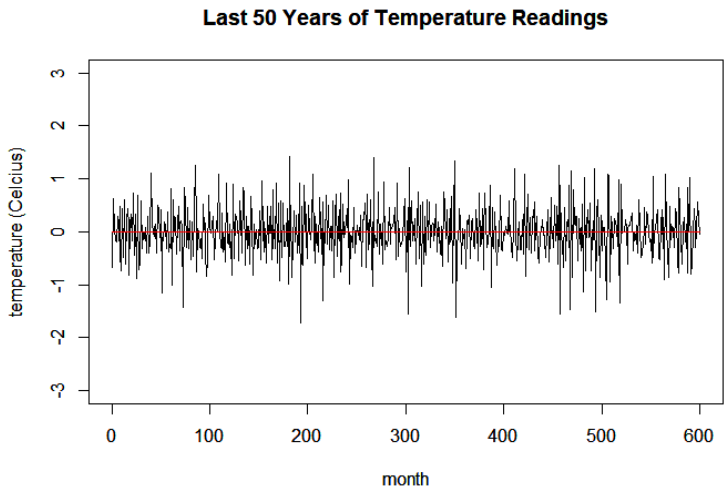


Figure 6. Seasonality in the time series (black)

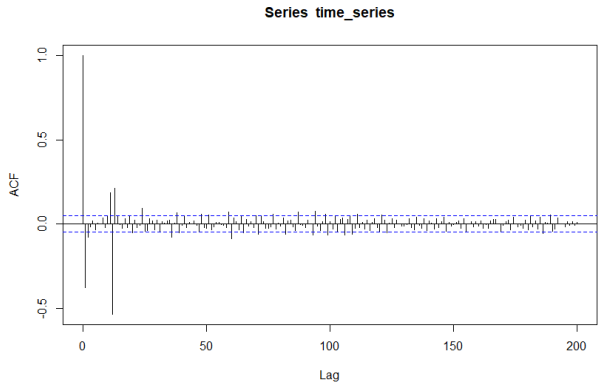
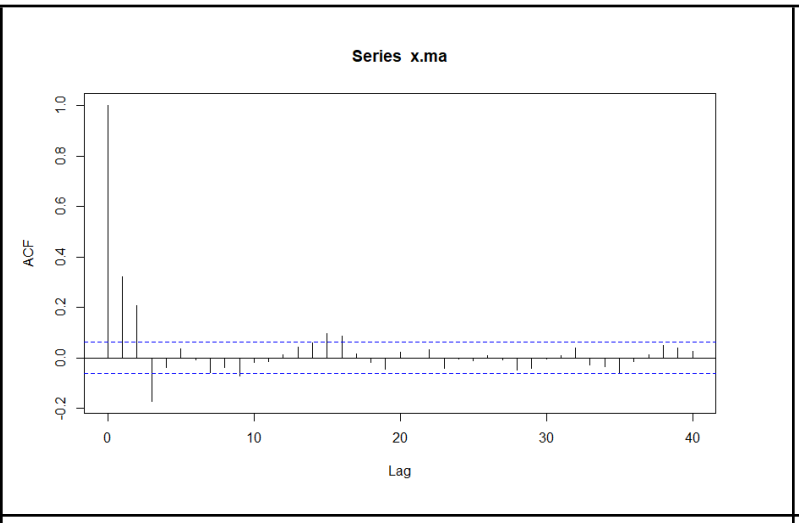
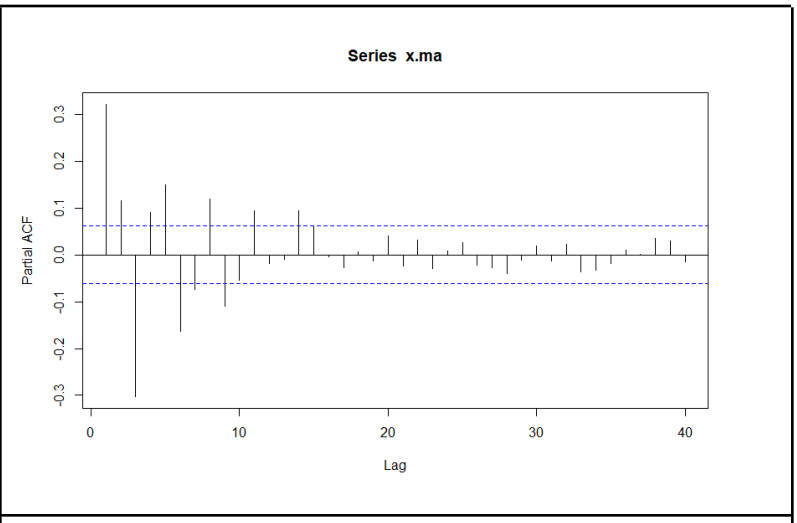
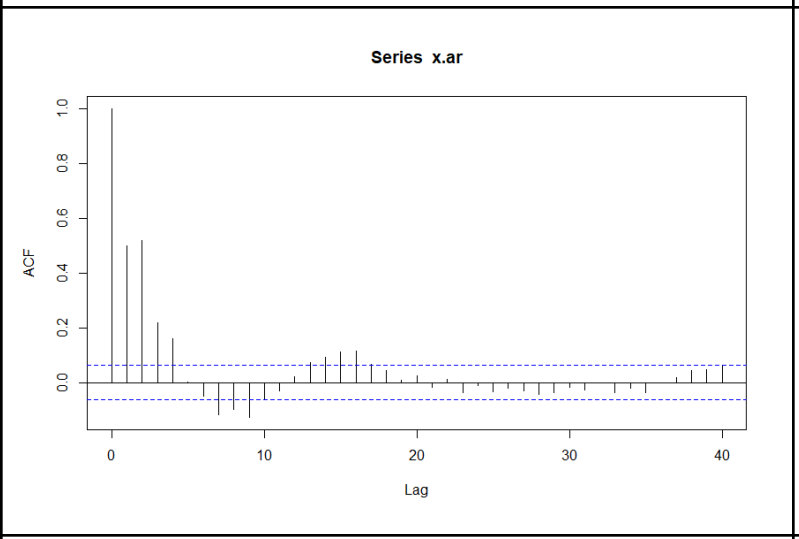
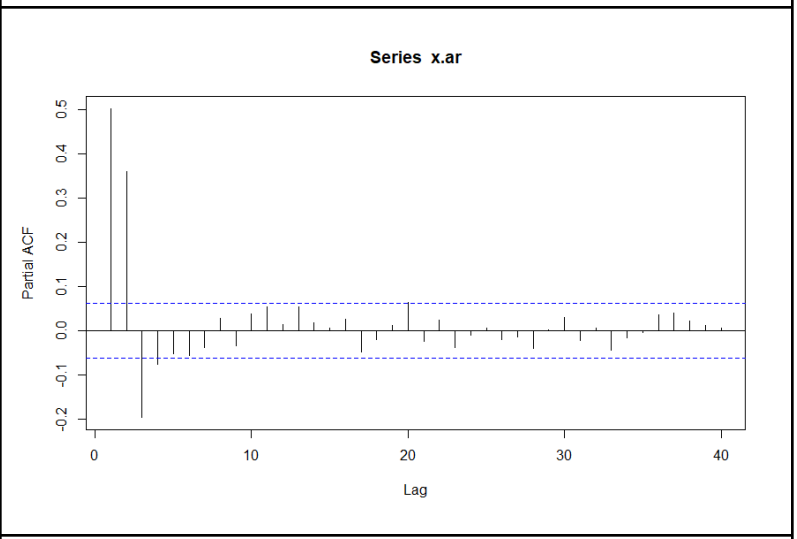


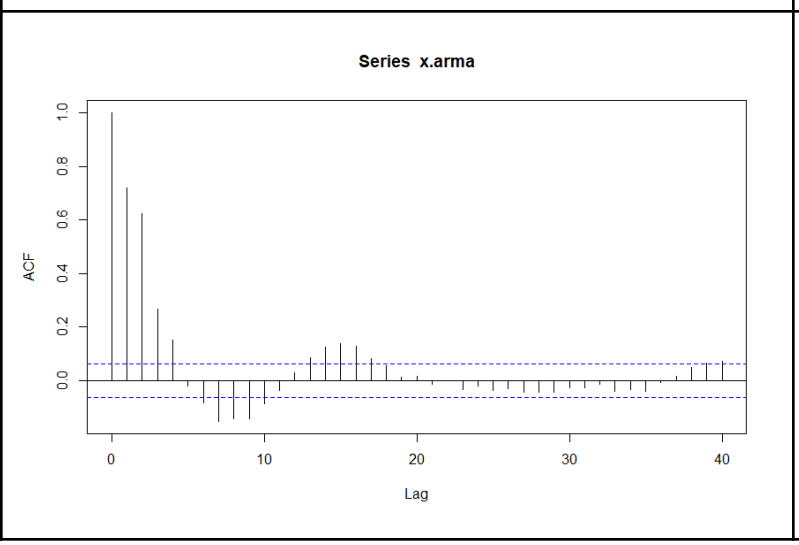
Figure 7. Auto-correlation function of the time series

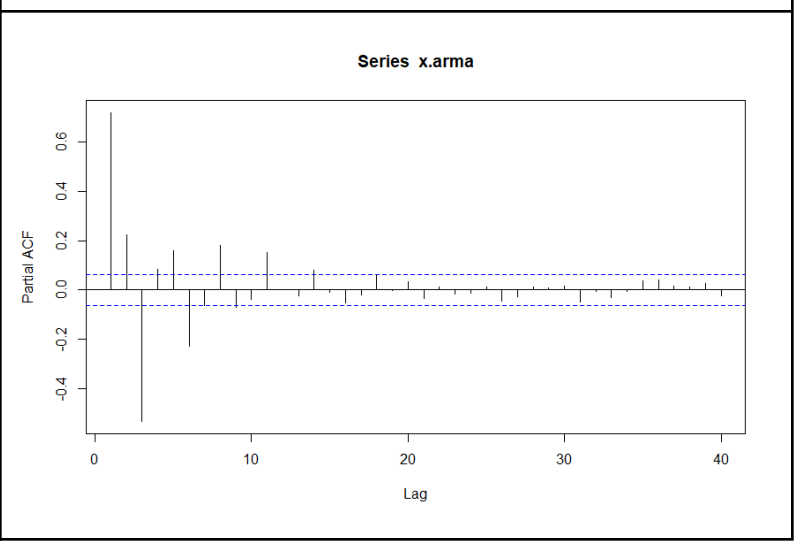

Series x.ma
ACF
Lag

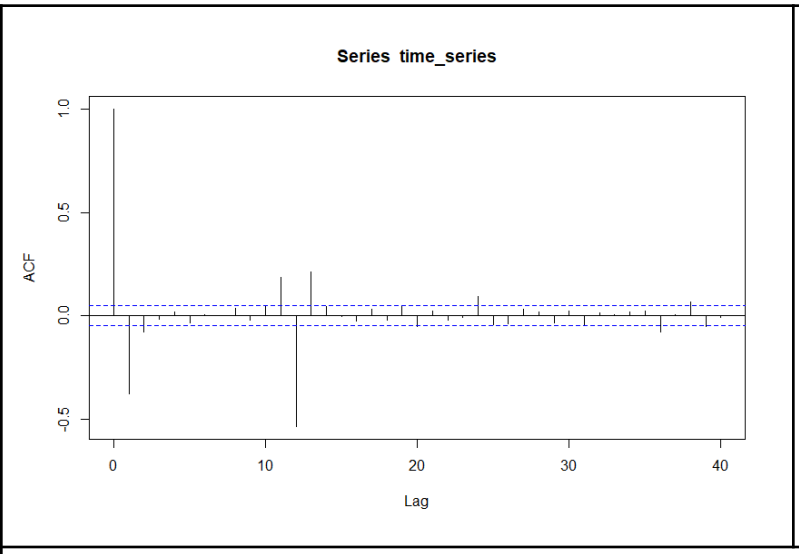

Series x.ma
Partial ACF
Lag


Series x.ar
ACF
Lag


Series x.ar
Partial ACF
Lag


Series x.arma
ACF
Lag


Series x.arma
Partial ACF
Lag



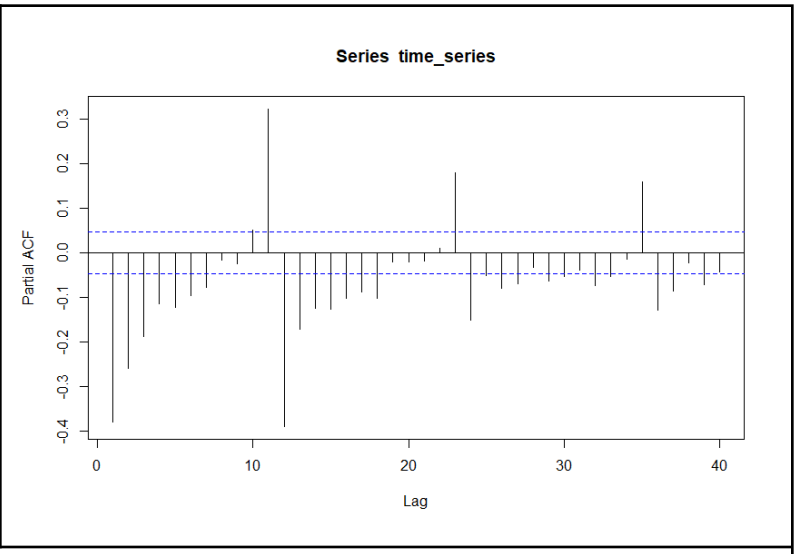


Figure 8. *ACF and PACF for MA(4), AR(4), ARMA(4,4) and our time series*

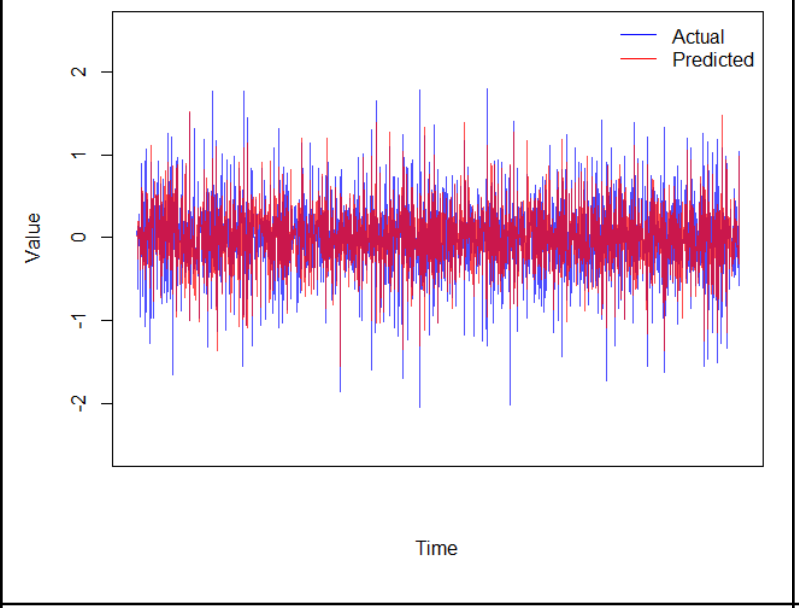


Figure 9. The generated ARMA(4,13) applied on the train-split - Mean Squared Residual Error : 0.09067194

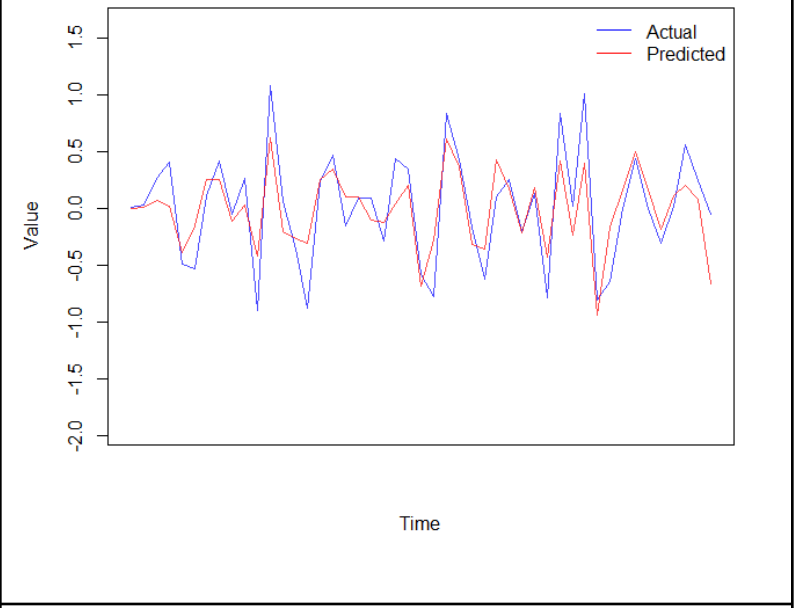


Figure 10. The generated ARMA(4,13) applied on the test-split - Mean Squared Residual Error : 0.07901927

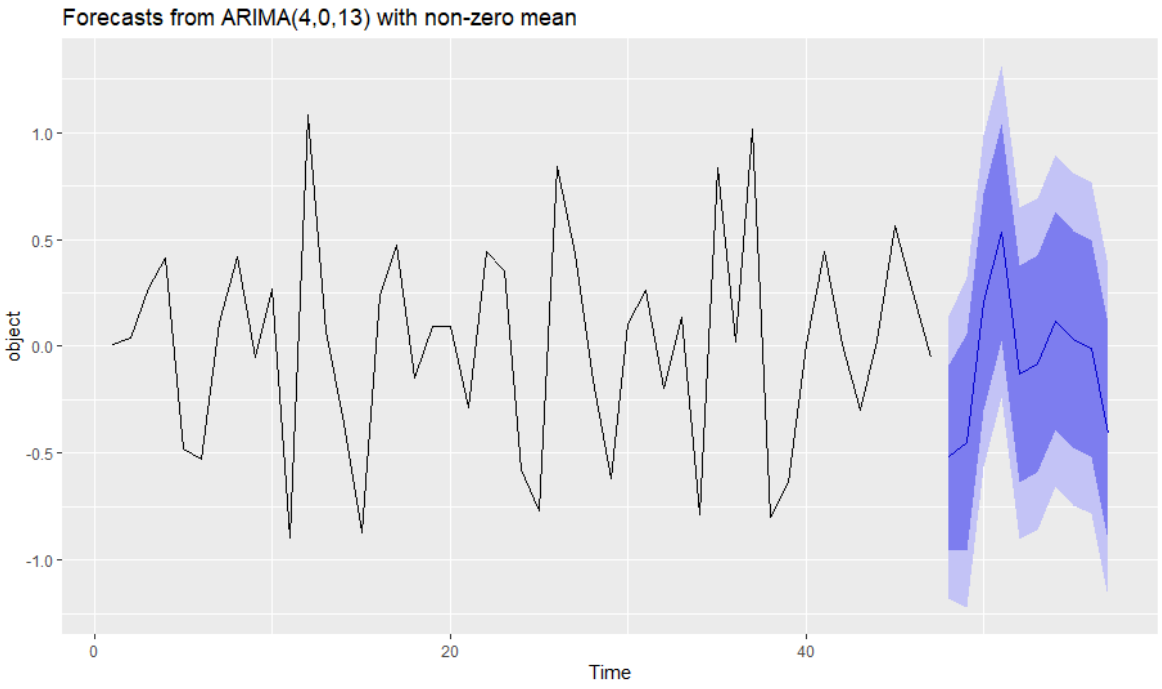


Figure 11. Forecasting on the test split with ARIMA(4,0,13)

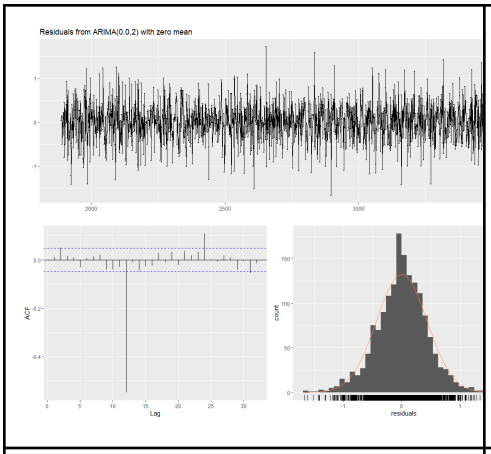


Figure 12. Residual analysis of MA(2)

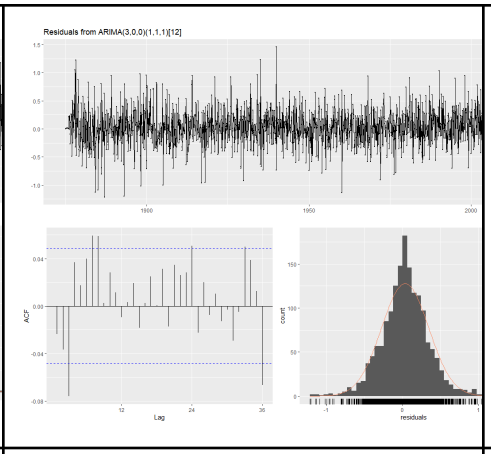


Figure 13. Residual analysis of ARIMA(3,0,0)(1,1,1)[12]

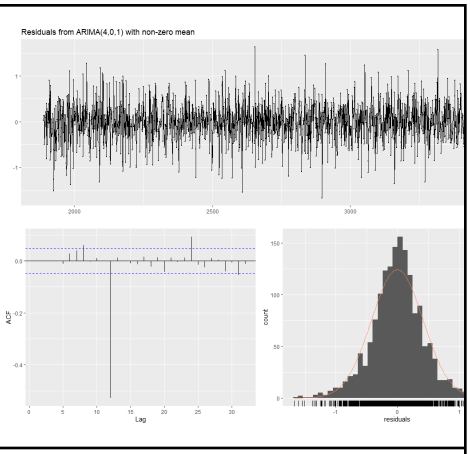


Figure 14. Residual analysis of ARMA(4,1)

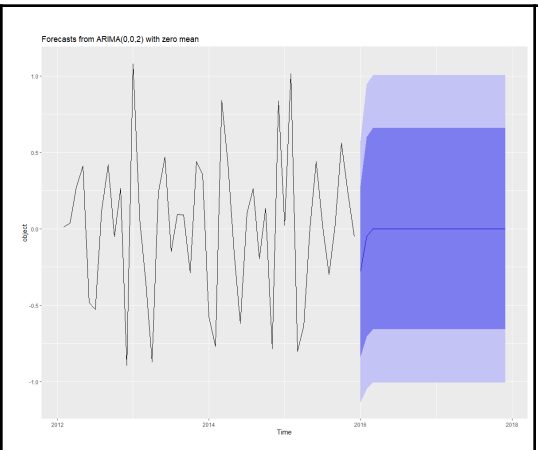


Figure 15. Forecast of MA(2)

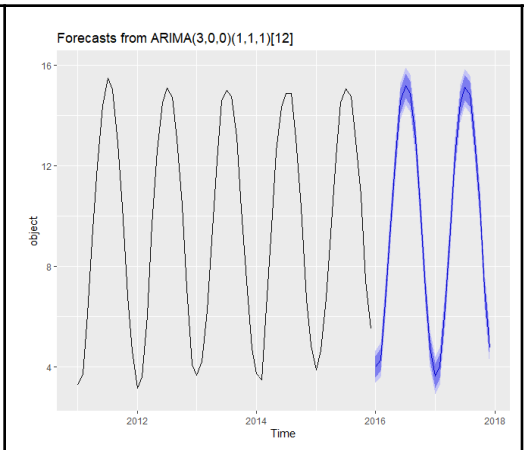


Figure 16. Forecast of ARIMA(3,0,0)(1,1,1)[12]

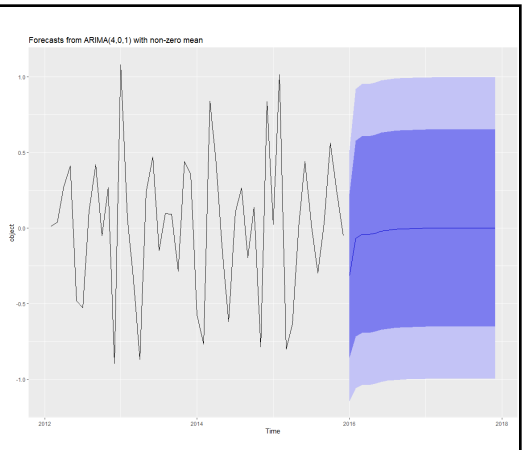


Figure 17. Forecast of ARMA(4,1)

AR 1	AR 2	AR 3	AR 4	MA 1	MA 2	MA 3	MA 4	MA 5	MA 6	MA 7	MA 8	MA 9	MA 10	MA 11	MA 12	MA 13
0.3290	0.1	0.0648	0.0886	-0.9817	-0.005	-0.0189	0.0031	-0.0056	0.0111	0.0164	0.0078	0.0119	-0.0124	0.0071	-1.0029	0.9692

Table 1. ARMA(4,0,13) generated coefficients