# SF2943 - Time Series Analysis Project
# YETFI Group

Yage Hao | Ellinor Kalderén | Thomas Lundqvist | Ioannis Athanasiadis

## Part A -Time Series Analysis of the Climate

### Problem statement

In the context of this project, we were asked to choose and analyse a time series of our choice. We found Climate Change to be quite an intriguing topic and we 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 are:

- *itsmr*
- *forecast*

Functions that are 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 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 measured from the earth's surface was measured 12 times a year, once a month from 1750 to 2015. The dataset also contained the measurement 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 Fig 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 [1800-2010] 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. Furthermore, 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 the 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 searching 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 on the time series. Finally the generated model was also used to forecast future values as shown in Figure 11.  The parameters used 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 potentials 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.

# 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 stationarity. Around this function, we applied on 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 *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.

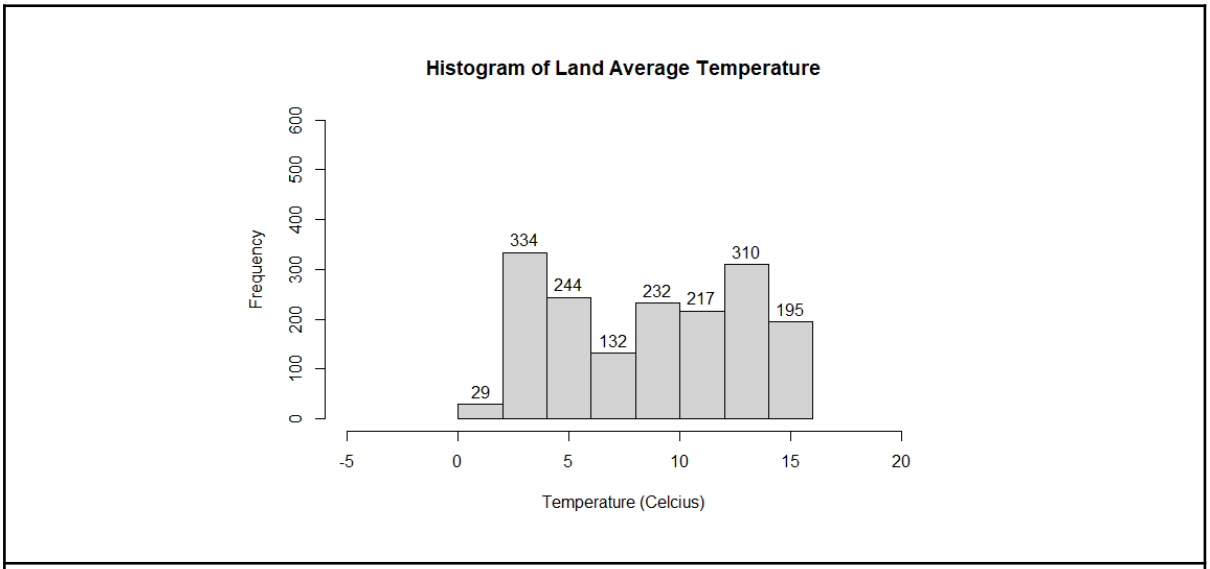



Figure 1. Surface temperature histogram of our time series

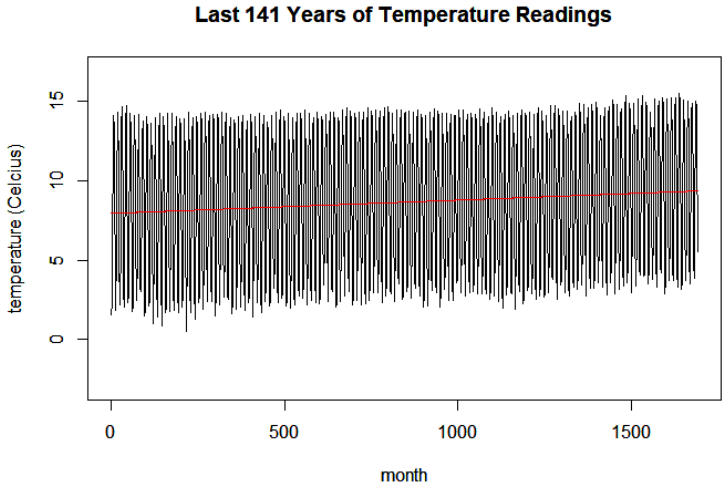



Figure 2. Trend (red) in the time series (black)

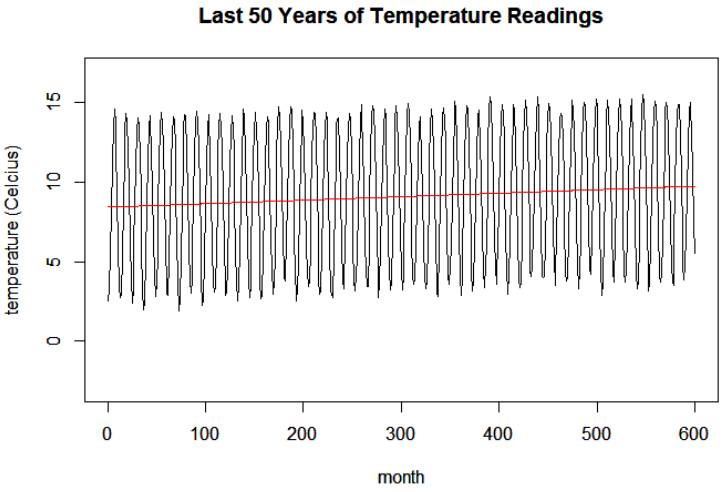



Figure 3. Seasonality in the time series (black)

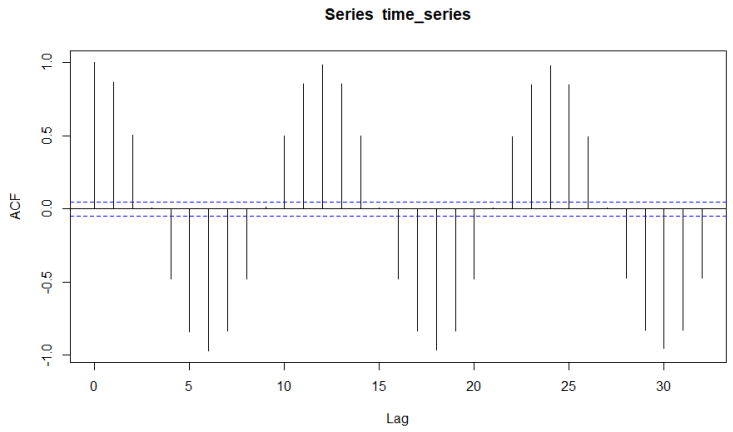



Figure 4. Auto-correlation function of the time series

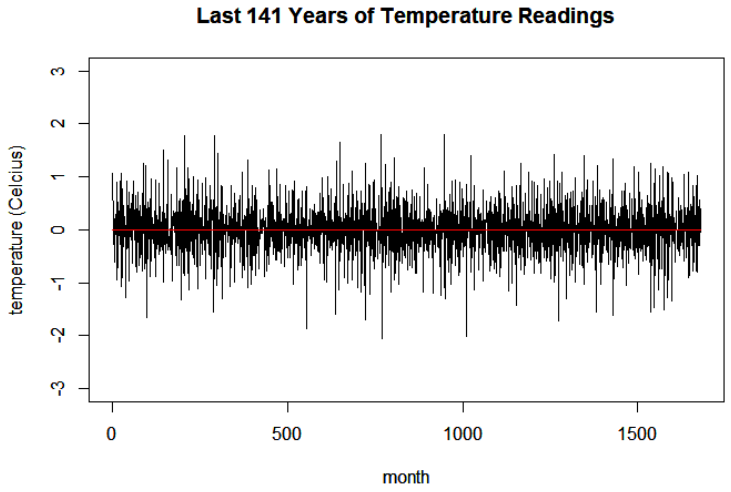



Figure 5. Trend (red) in the time series (black)

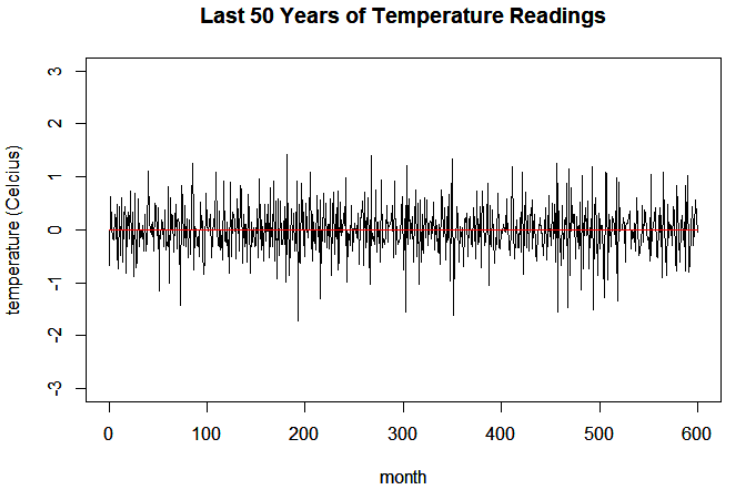



Figure 6. Seasonality in the time series (black)

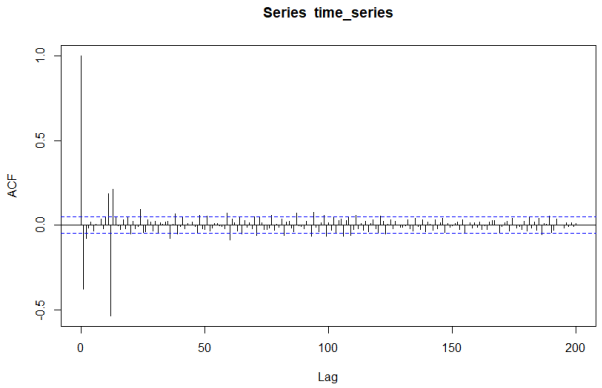
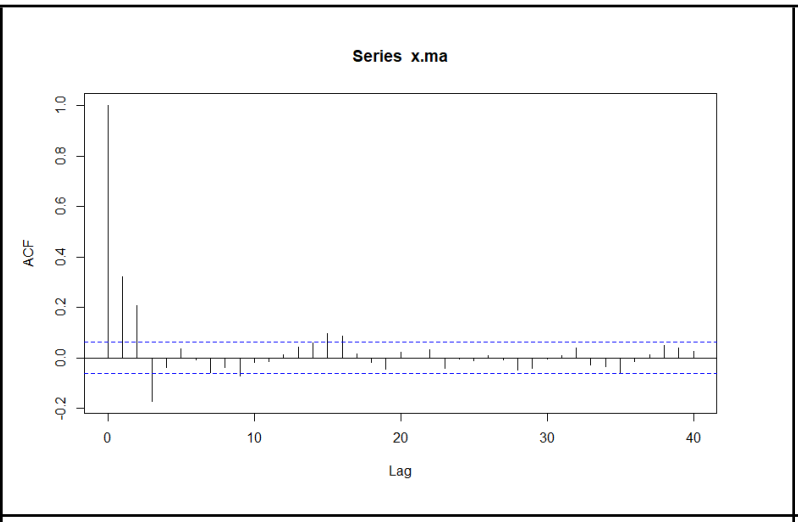
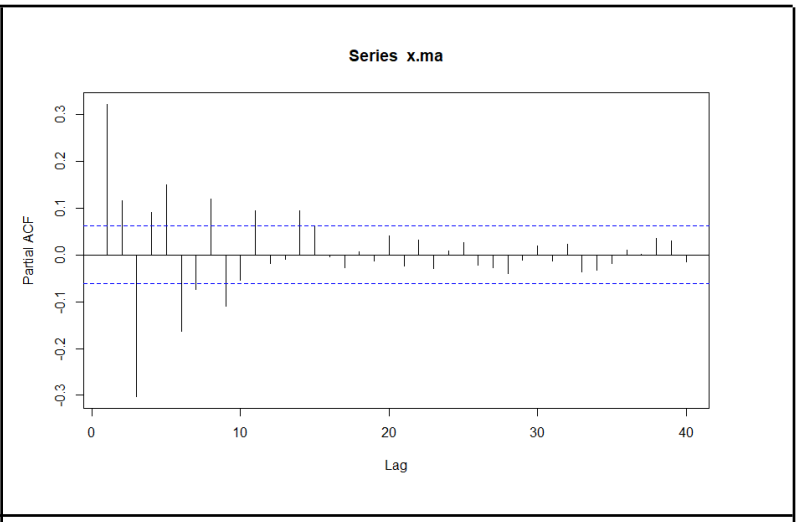
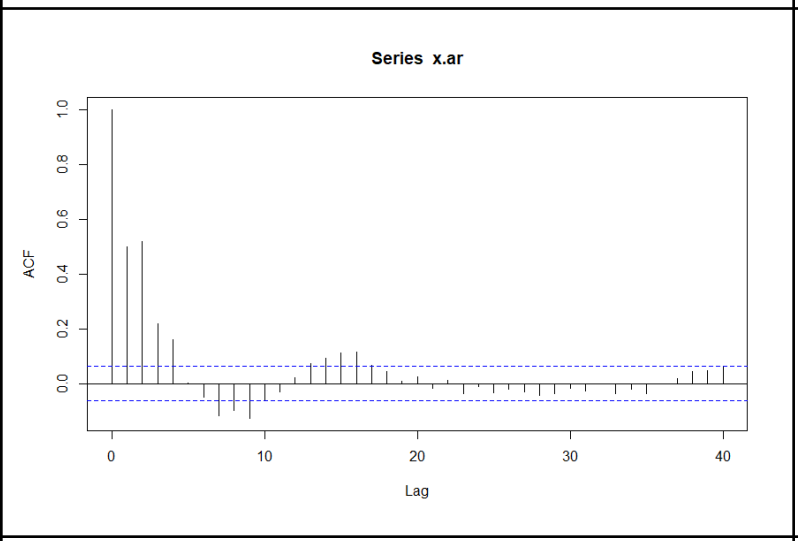
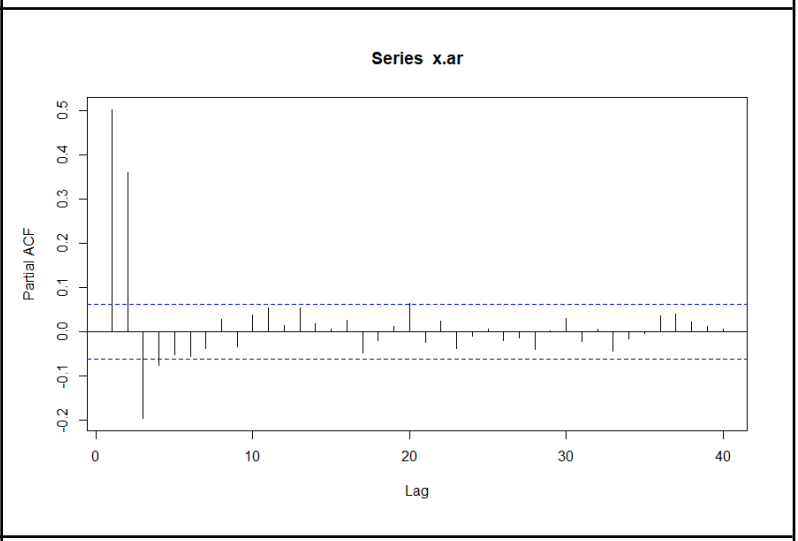


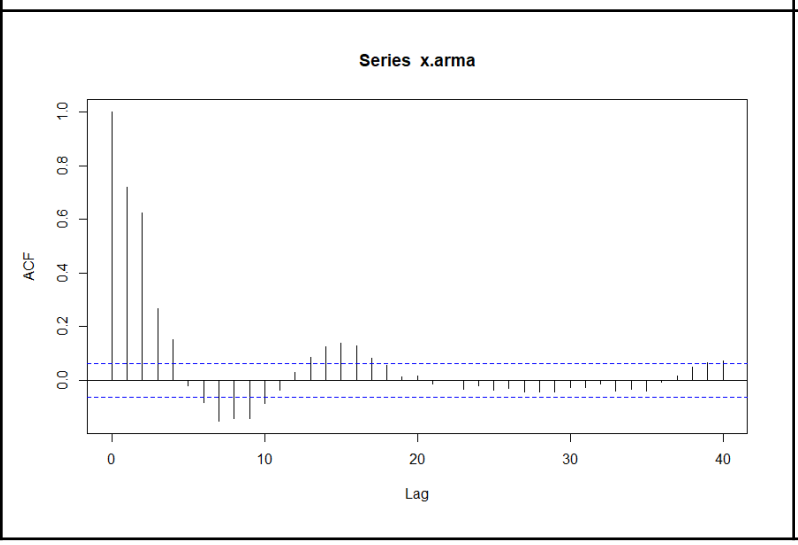

Figure 7. Auto-correlation function of the time series

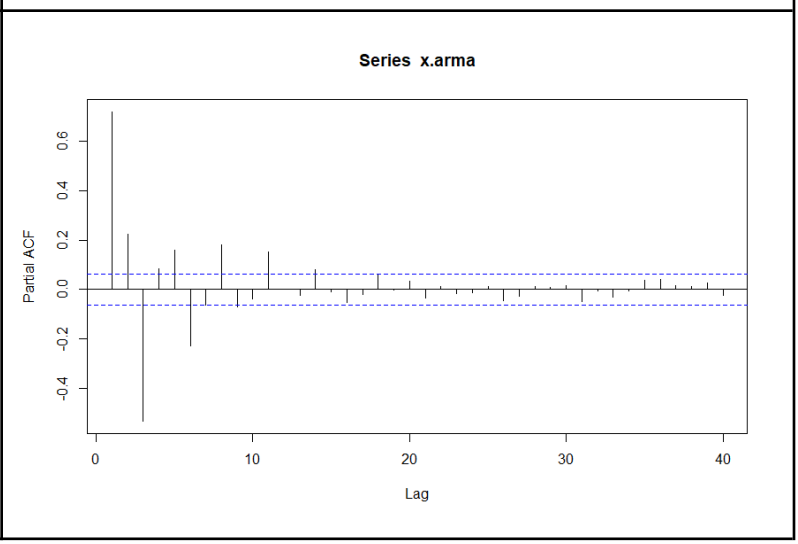

Series x.ma
ACF
Lag

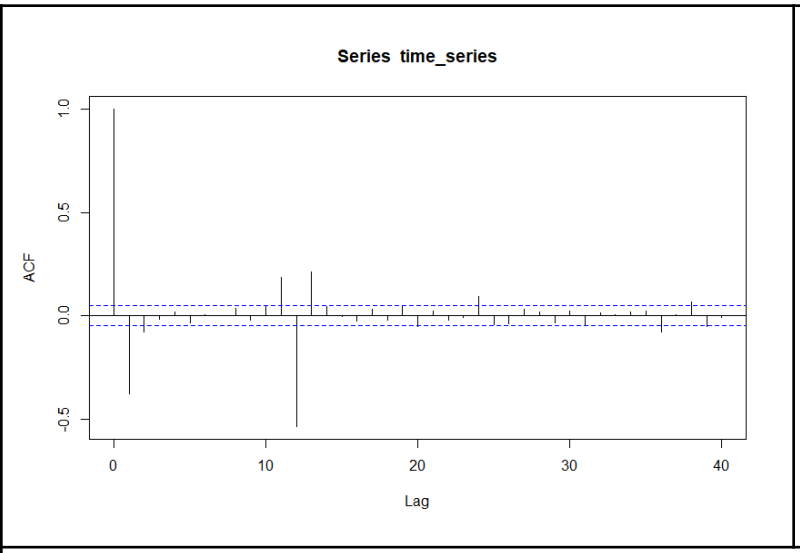

Series x.ma
Partial ACF
Lag


Series x.ar
ACF
Lag


Series x.ar
Partial ACF
Lag


Series x.arma
ACF
Lag


Series x.arma
Partial ACF
Lag



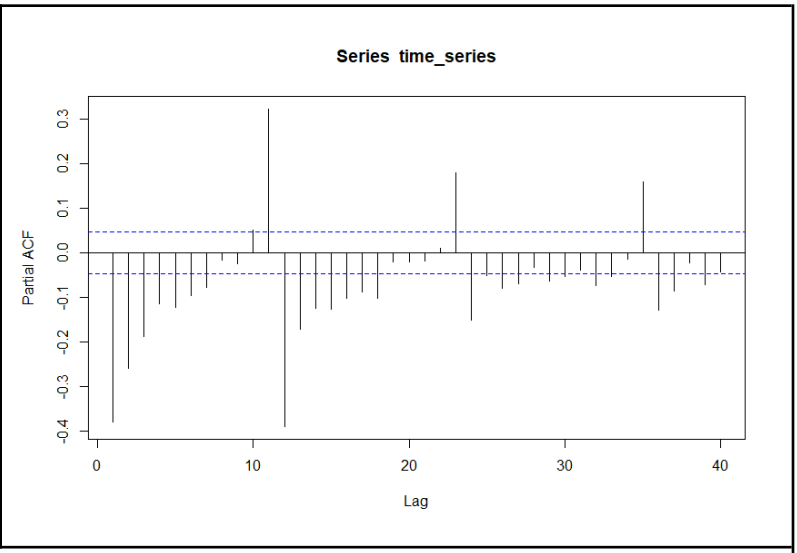




*Figure 8. ACF and PACF for MA(4), AR(4), ARMA(4,4) and our time series*

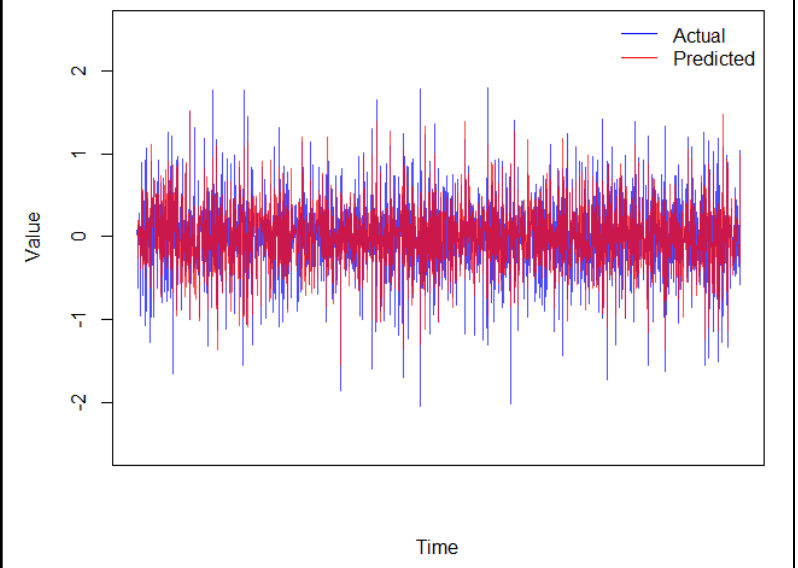



Figure 9. The generated ARMA(4,13) applied on the train-split - Mean Squared Residual Error : 0.09067194

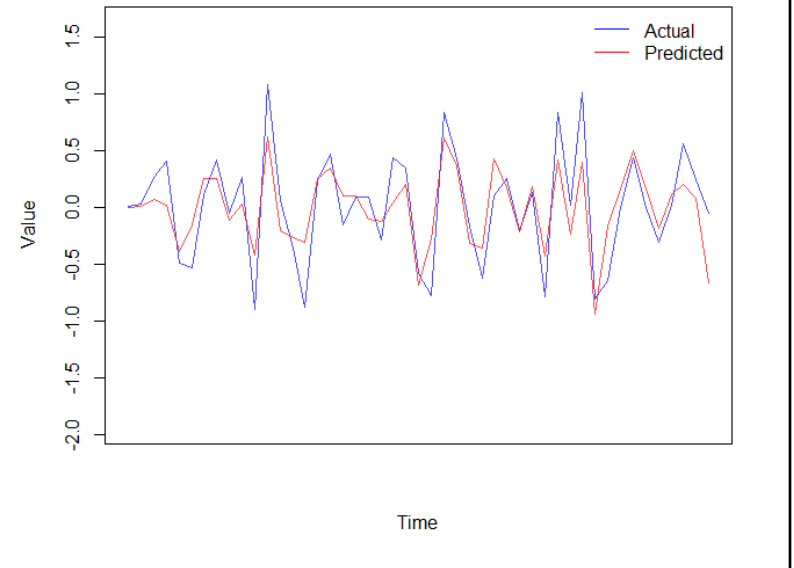



Figure 10. The generated ARMA(4,13) applied on the test-split - Mean Squared Residual Error : 0.07901927

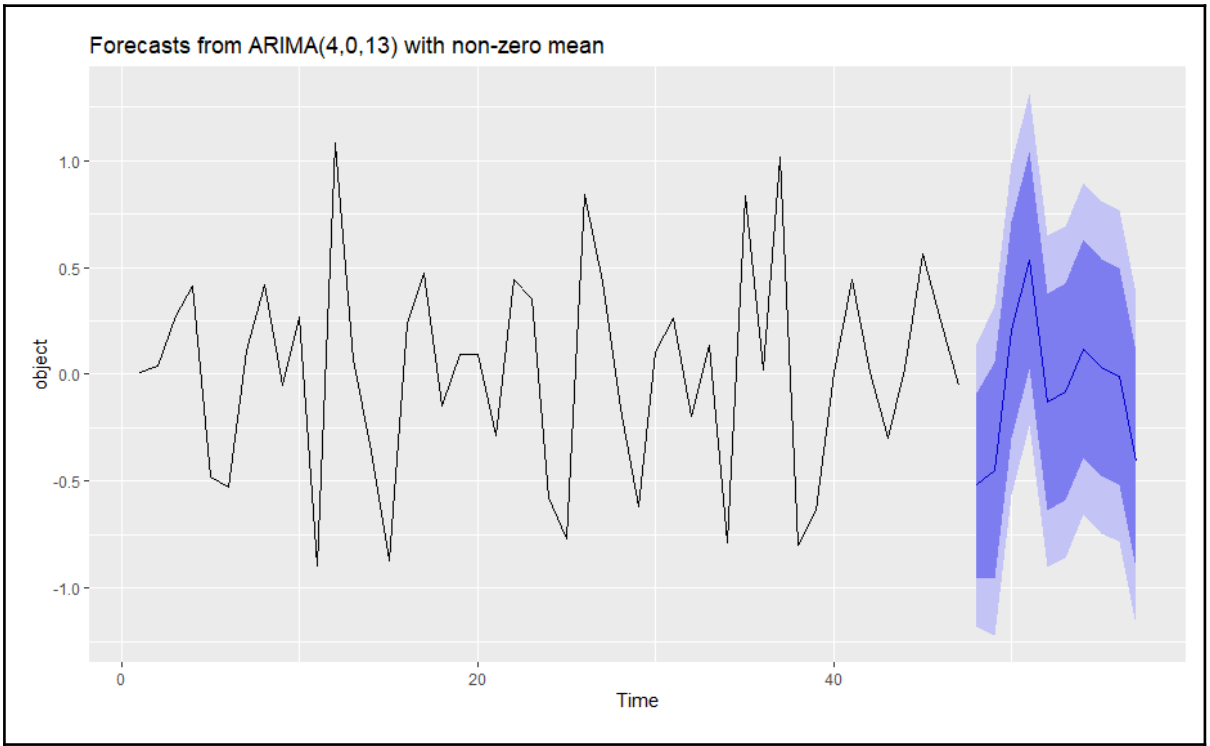



Figure 11. Forecasting on the test split with ARIMA(4,0,13)

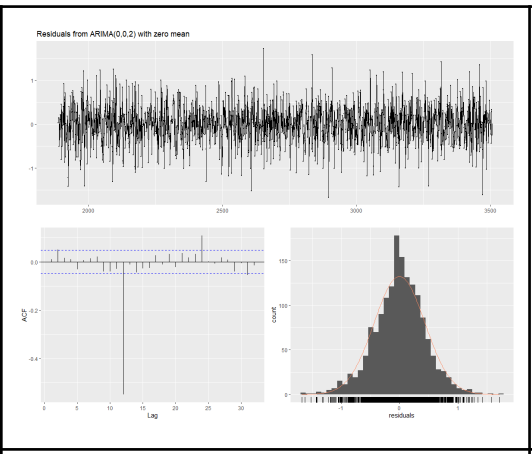



Figure 12. Residual analysis of MA(2)

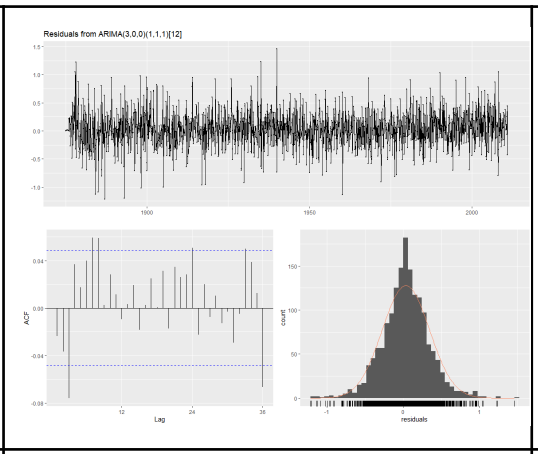



Figure 13. Residual analysis of ARIMA(3,0,0)(1,1,1)[12]

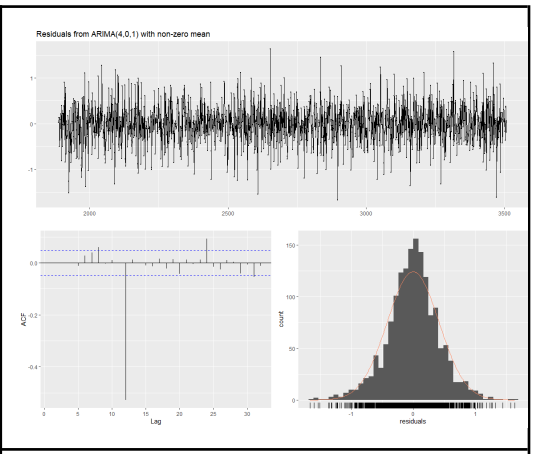



Figure 14. Residual analysis of ARMA(4,1)

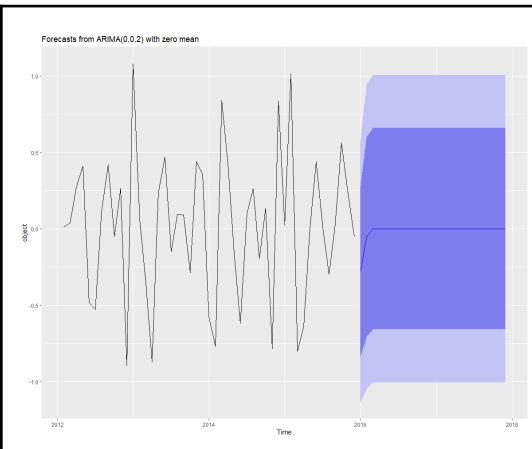



Figure 15. Forecast of MA(2)

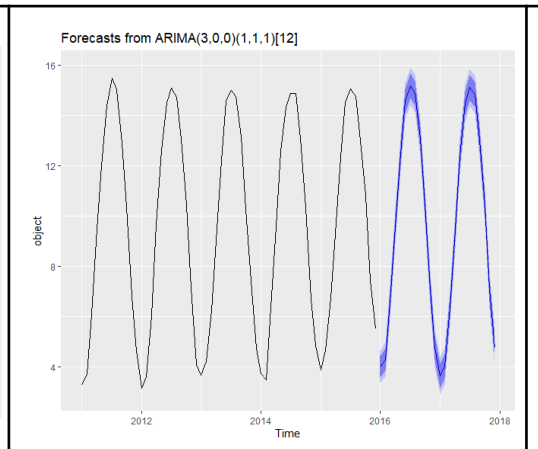



Figure 16. Forecast of ARIMA(3,0,0)(1,1,1)[12]

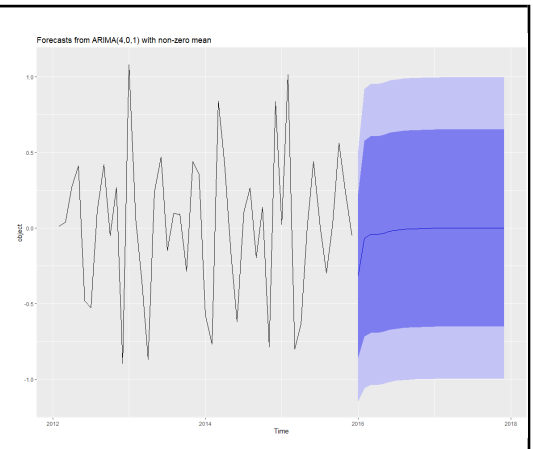



Figure 17. Forecast of ARMA(4,1)

| 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

## 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.**