Computer Exercise 1

Christian Glissov, s146996

October 9, 2018

DTU Compute - Institute for Mathematics and Computer science
Advanced Time Series Analysis 02427 October 9, 2018
Henrik Madsen & Erik Lindstrøm

Technical University of Denmark

Made in collaboration with Frederik Boe Hüttel, s151689, and Jonas Gottlieb Svendsen, s152992.

Contents

Question 1.1

In this exercise three different non-linear models will be simulated.

SETAR(2,1,1)

The first model is the SETAR model. From [2], (1.17) and (1.18), a SETAR model can be defined. The model used is a SETAR(2,1,1) model, it consists of 2 regimes, based on AR(1) processes and the regime shift is based on the previous observation,

$$X_t = \begin{cases} 4 + 0.5 \cdot X_{t-1} + \epsilon_{t,1} & X_{t-1} < 0 \\ -4 - 0.5 \cdot X_{t-1} + \epsilon_{t,2} & X_{t-1} \geq 0 \end{cases} \tag{1}$$

Where $\epsilon_{t,i} \in \mathbb{N}(0,1)$. Simulating (1) shows the following time series:

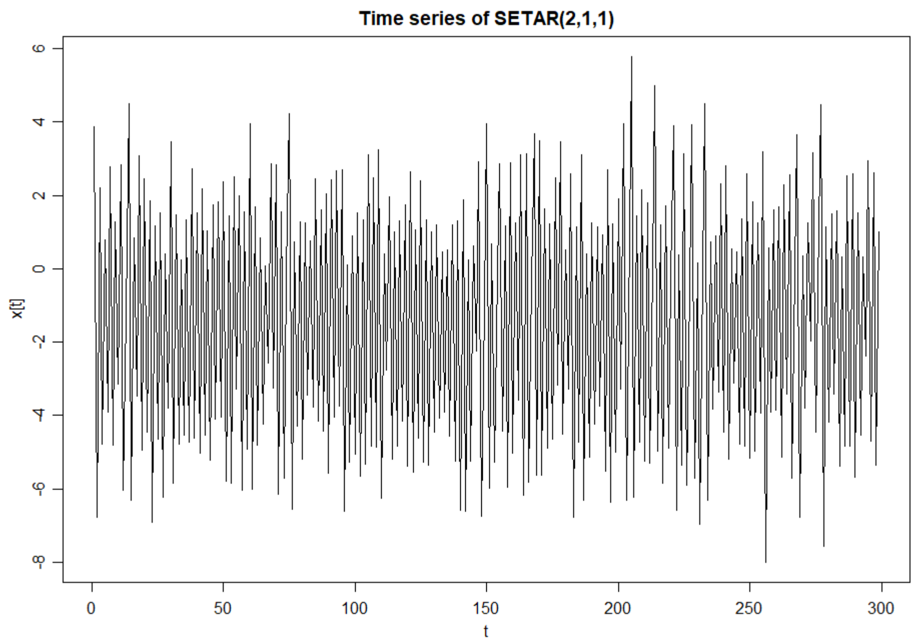

Figure 1: Simulation of the SETAR(2,1,1) in (1). Looks rather random, but it is created by two regimes.

Looking at the lag plot the regimes can be observed:

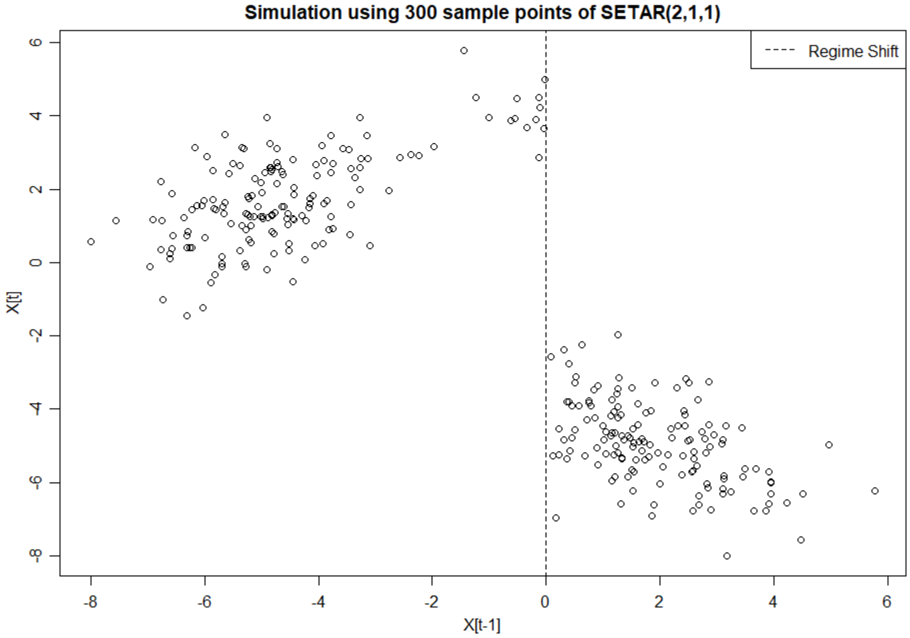

Figure 2: Clusters can now be observed, the dashed line is where the regime shift occurs and is clearly noticeable in the lag plot.

One might notice that there is a shift in the regime at 0, which was also specified in the model. Two clusters can almost be observed, both heavily affected by the noise term, this is due to slope of the AR(1) processes, which in this case is set fairly low as 0.5, if even lower the noise term will start to dominate the series. From (1.17) it can be seen that $a_0^{(J_t)}$ term will act as an intercept for the mean and will therefore the intercept will increase the distance between the clusters. Furthermore $a_i^{(J_t)}$ will act as weighting of the previous observations.

IGAR(2,1)

The second model is the IGAR(2,1) model. Once again from [2], (1.22) and (1.23), an IGAR(2,1) model can be defined, 2 regimes are defined, both including an AR(1) process,

$$X_t = \begin{cases} 4 + 0.5 \cdot X_{t-1} + \epsilon_{t,1} & p_1 < 0.5 \\ -4 - 0.5 \cdot X_{t-1} + \epsilon_{t,2} & p_2 \geq 0.5 \end{cases} \tag{2}$$

Where $\epsilon_{t,i} \in \mathbb{N}(0,1)$. The regimes shift of an IGAR model is determined by an independent stochastic variable, independent of the noise sources and governed by p, an independent stochastic parameter, as in (1.23) from [2]. Simulating Equation 2, with $p \in U(0,1)$, from the stochastic selection, the following time series is simulated:

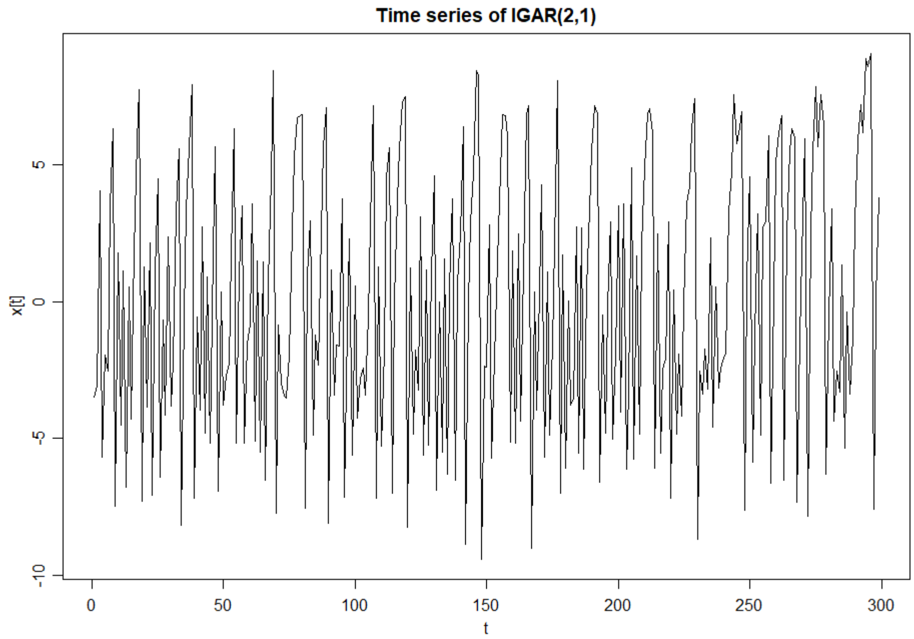

Figure 3: Simulation of a time series generated by an IGAR(2,1) model as defined in Equation 2, 300 sample points are used.

Once again the time series looks like white noise. However again by looking at the lag plot a different picture will appear:

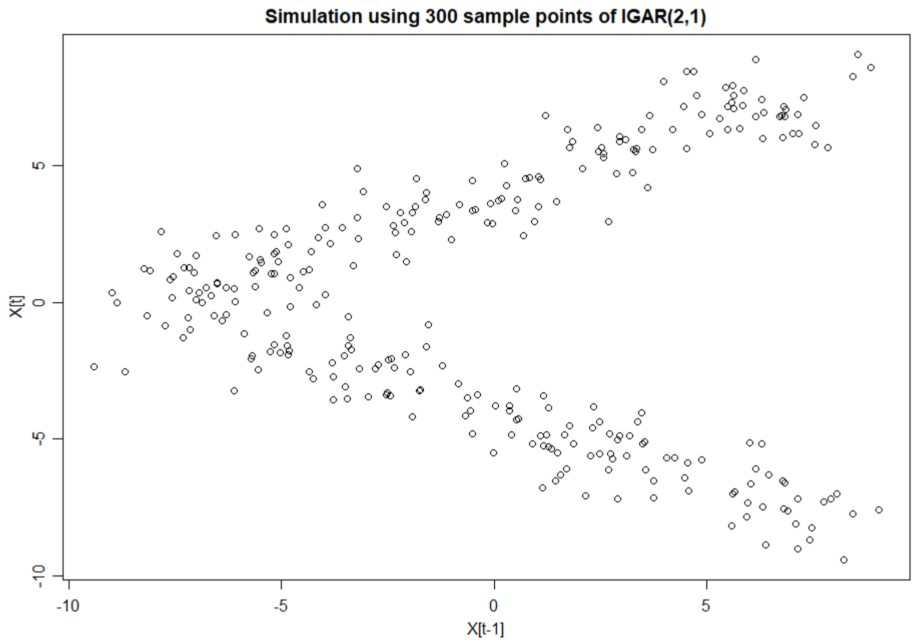

Figure 4: Looking at the lag plot, two symmetric regimes seem to be noticeable.

Figure 4 shows two regimes. The regimes look symmetric and fairly uniformly distributed, this is due to the regime shift being 0.5, which makes each regime being equally likely to be chosen.

MMAR(2,1)

The final model is the MMAR(2,1) model. The regime shift is governed by states and the transition probability matrix (t.p.m), for a MMAR(2,1) the t.p.m will look as follows [3]:

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} \\ p_{2,1} & p_{2,2} \end{bmatrix}$$

This matrix gives the probabilities of changing state with each row summing to 1. Each regime will then be chosen based on the current state. For this example the t.p.m will be defined as,

$$\mathbf{P} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}$$

This means there is 95% chance of staying in the current state and 5% of changing state. The Markov regime will then be defined as,

$$X_t = \begin{cases} 4 + 0.5 \cdot X_{t-1} + \epsilon_{t,1} & s_1 = 1 \\ -4 - 0.5 \cdot X_{t-1} + \epsilon_{t,2} & s_2 = -1 \end{cases} \tag{3}$$

Where $\epsilon_{t,i} \in \mathbb{N}(0,1)$ and s defines the state. Figure 5 shows the simulation of Equation 3 and the states:

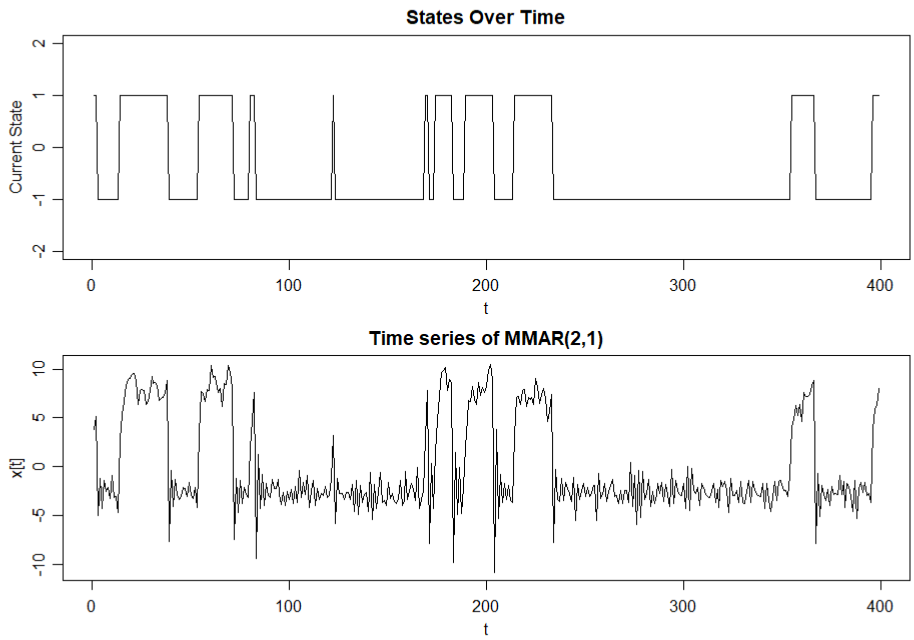

Figure 5: Simulation of a time series generated by an MMAR(2,1) model as defined in Equation 3, 400 sample points are used.

Due to the high probabilities in the diagonal of the t.p.m, the state will rarely change, this causes a rather inert Markov process as seen in Figure 5. In this case the states switches between -1 and 1.

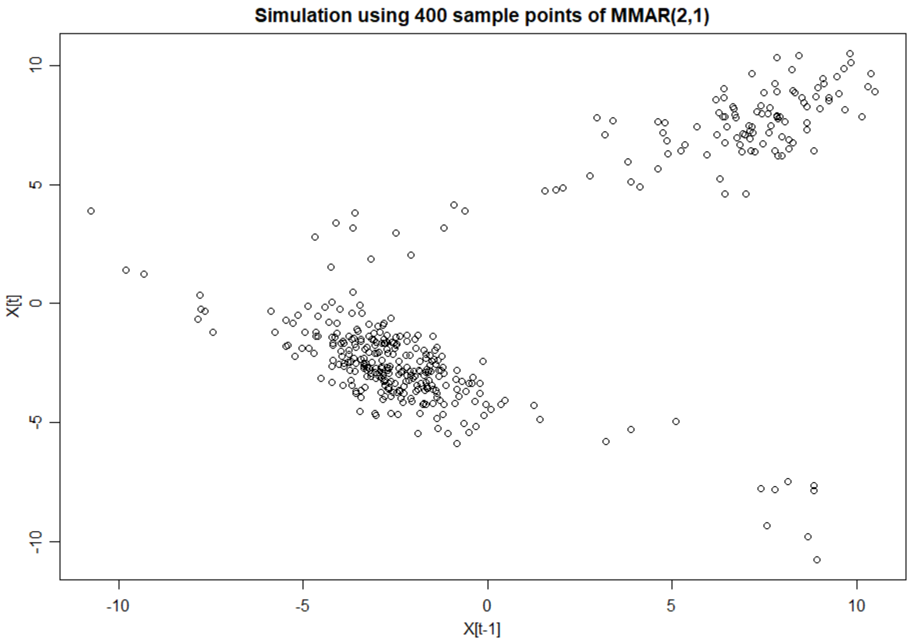

Figure 6: Looking at the lag plot, two regimes seem to be noticeable, where one of the regimes seem to contain a larger cluster of data than the other.

Looking at the lag plot in Figure 6 it is noticed that one of the clusters is larger than the other, this might be due to once again the low rate of change of the states, which makes data accumulate. To generate a simulation much alike the one for the IGAR(2,1) process the t.p.m would need to contain equal probabilities at each index, such as 0.5, this would create a rather symmetric process as in Figure 4.

Question 1.2

Based on the SETAR(2,1,1) process in Equation 1, the theoretical conditional mean can easily be found from eq. (2.1) in [2], that for a non-linear time series on the form of:

$$X_t = g(X_{t-1}, ..., X_{t-p}) + h(X_{t-1}, ..., X_{t-p})\epsilon_t.$$

The conditional mean is given by

$$M(x) = E\{X_t|X_{t-1} = x\} = g(X_{t-1}, ..., X_{t-p})$$

Therefore by looking at the model in Equation 1, $M(x)$ is easily identified:

$$M(x) = \begin{cases} 4 + 0.5 \cdot X_{t-1} & X_{t-1} < 0 \\ -4 - 0.5 \cdot X_{t-1} & X_{t-1} \geq 0 \end{cases} \tag{4}$$

In practice the theoretical mean of the process is unknown. Therefore the theoretical conditional mean has to be estimated. To do this kernels can be used. In this question an Epanechnikov kernel, [2] (2.15), will be used to fit the model. The conditional mean is then estimated using (2.34) in [2] and the Epanechnikov kernel. The results when trying different bandwidths can be seen in Figure 7

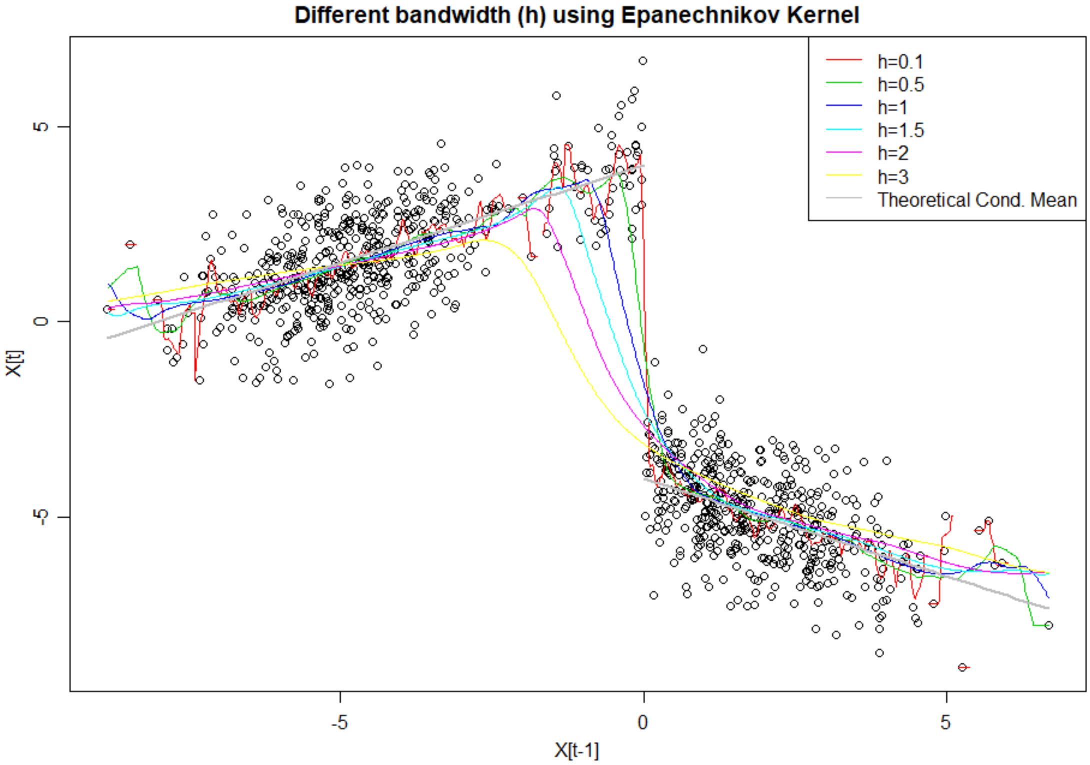

Figure 7: Estimated conditional mean with different bandwidths, using Epanechnikov kernel. The theoretical conditional mean is also shown. Number of simulations, $n = 1000$.

One might notice that when the bandwidth becomes too small the Epanechnikov will create discontinuities in the estimated conditional mean. This is due to the support $|u| \leq 1$, if $|u| > 1$, it will simply be set to 0. Since a small bandwidth h will not penalize the variance, the variance will increase and so will u, making the Epanechnikov returning only 0 values, giving NaNs when estimating the conditional mean.

If h is large it will reduce the variance, making it more biased towards the mean of the regimes. On the other hand if h is small the variance will increase, but the bias towards the mean will be small. The goal is to have a bandwidth with optimal tradeoff between bias and variance. Looking at Figure 7 a bandwidth between 0.5 to 1.5 seems to be good. To find an optimal bandwidth cross-validation of the MSE as given in (2.28) in [2]. Using the script *leaveOneOut.R* provided in the assignment and making small modifications to the script, the optimal bandwidth can be found as seen in Figure 8.

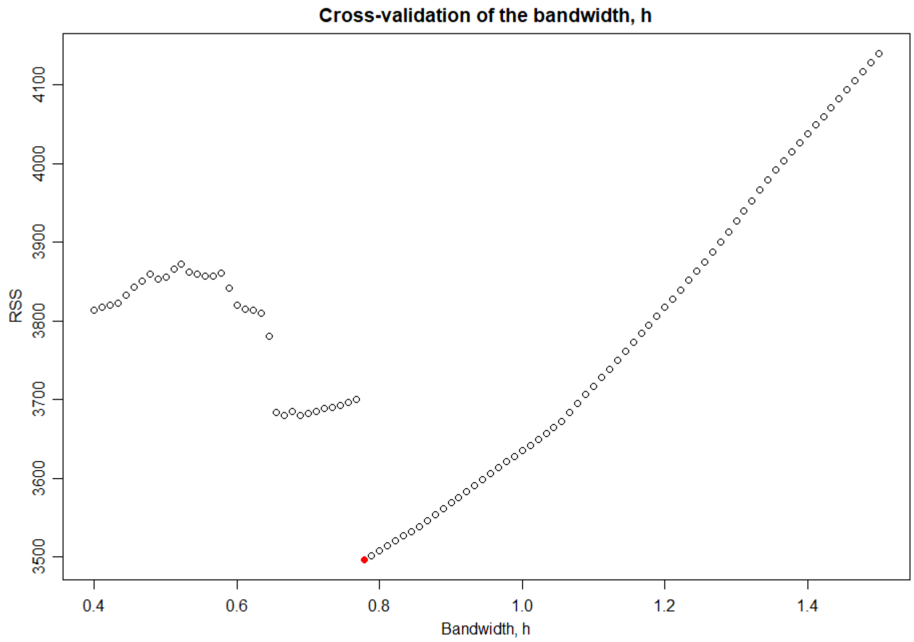

Figure 8: The red point is the optimal bandwidth, $h \approx 0.778$. The y-axis shows the residual sum of squares (RSS).

Once again using (2.34) and (2.36) from [2] to respectively estimate the conditional mean and variance. Confidence intervals can then be found ([1], "The Red Bible", (3.61)) and a fit using the optimal bandwidth is shown in Figure 9.

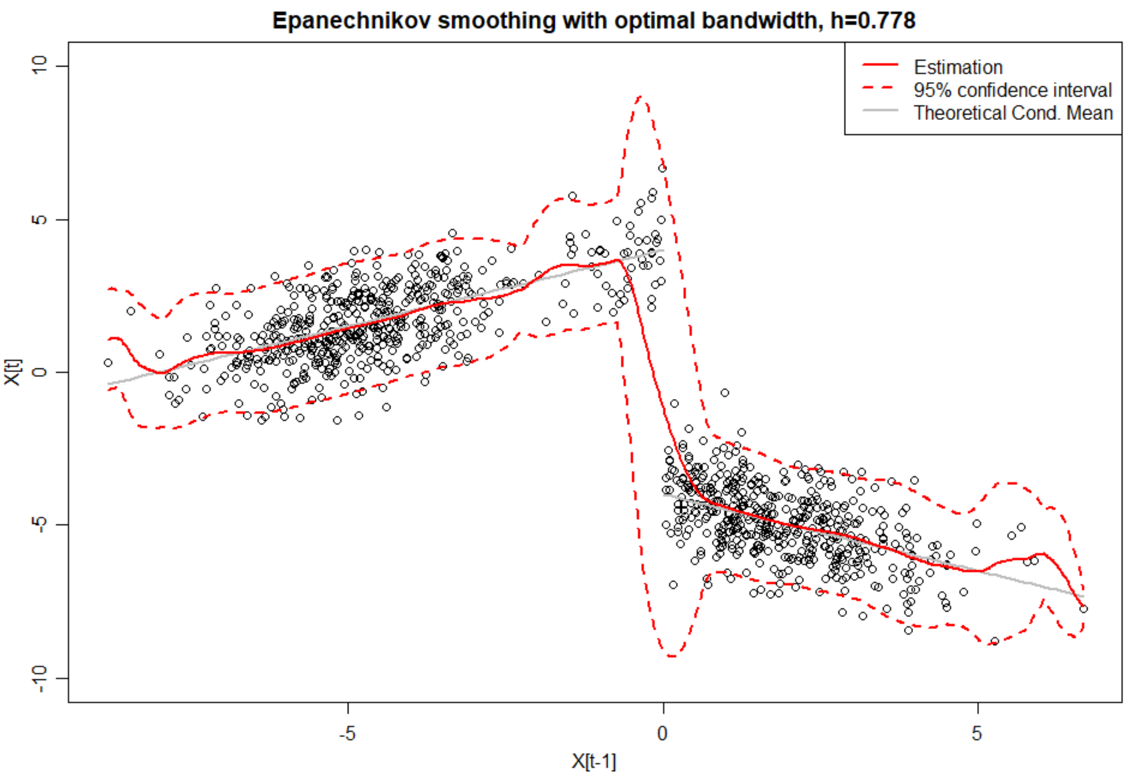

Figure 9: Estimation of the conditional mean of the SETAR(2,1,1) process.

The model captures the theoretical conditional mean of the data fairly well, except at the edges due to few observations or the abrupt regime shift.

Question 1.3

As mentioned in [2] chapter 3.2, there are several advantages to the cumulative versions of the conditional mean and variance. In this question the estimated cumulative conditional mean, $\hat{\Lambda}$, will be compared to the theoretical cumulative conditional mean, Λ. To find $\hat{\Lambda}$ equation (3.6) and (3.8) is used from [2], this has already been implemented in the script *cumulativeMeans.R*. To find the cumulative theoretical conditional mean, (3.4) is simply used.

$$\Lambda(\cdot) = \int_a^{\cdot} \lambda(x)dx$$

Where $\lambda(x)$ is simply given by the theoretical conditional mean. Therefore by doing a trivial integration of the conditional mean given in Equation 4 the following is found:

$$\Lambda = \begin{cases} 4(b-a) - 0.25(a^2 - b^2) & X_{t-1} < 0 \\ -4(b-a) + 0.25(a^2 - b^2) & X_{t-1} \geq 0 \end{cases} \tag{5}$$

Where attention is restricted to some fixed interval $[a, b]$. The cumulative theoretical conditional mean is shown in Figure 4 using *R*.

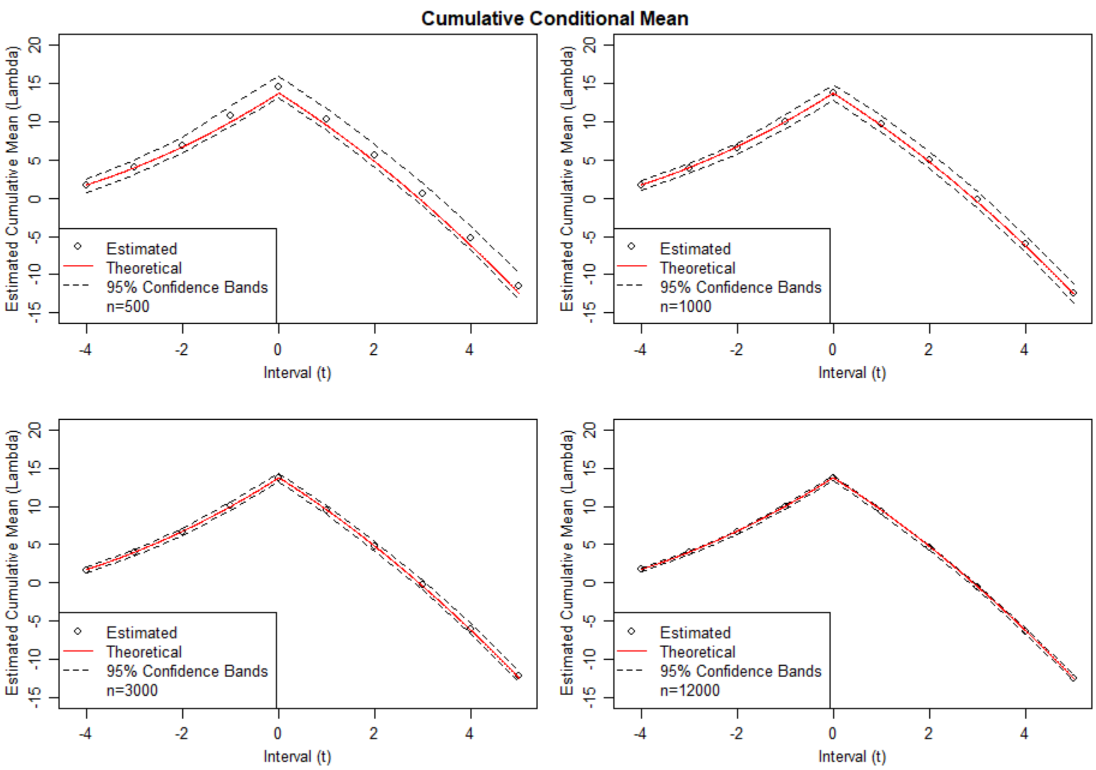

Figure 10: Different simulations of the estimated cumulative conditional mean, $n = \{500, 1000, 3000, 12000\}$. The abrupt break at $t = 0$ is due to $a_1^{(1)} = 0.5$ and $a_1^{(2)} = -0.5$ have different sign in Equation 1

From Figure 10 it is easily seen that when the number of simulations increases, $n \to \infty$, then the estimated cumulative conditional mean asymptotically converges to the theoretical cumulative conditional mean, $\hat{\Lambda} \to \Lambda$. The confidence bands also become slimmer, due to the decrease of uncertainty and this is also in line with what one should expect from (3.10) in [2]. It is therefore possible to estimate the theoretical cumulative conditional mean fairly well, given enough data is present.

Question 1.4

In this question an estimated heat loss function of the wind speed, $U_a(W_t)$, is desired. To model this problem the deterministic part of the model below is found:

$$\Phi_t = U_a(W_t)(T_t^i - T_t^e) + \epsilon_t \iff$$

$$U_a(W_t) = \frac{\Phi_t}{(T_t^i - T_t^e)}. \tag{6}$$

In Figure 11 a plot of $U_a(W_t)$ can be seen.

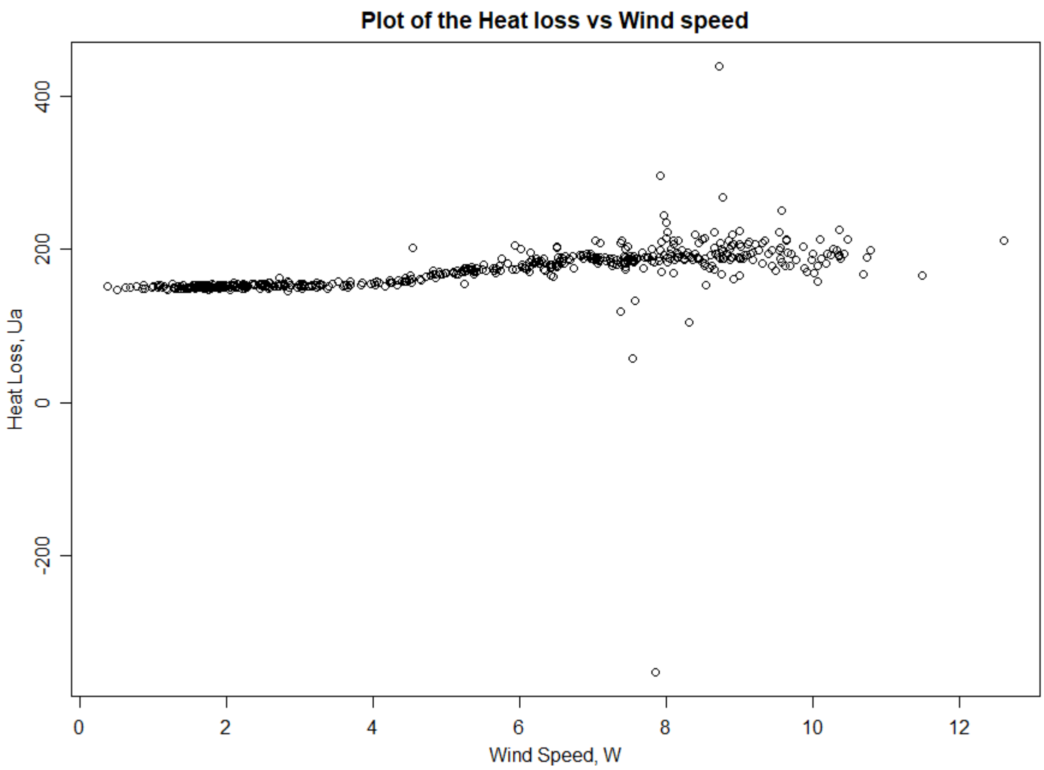

Figure 11: Plot of $U_a(W_t)$

Looking at Figure 11 it can be seen that certain outliers appear. These few observations will increase the variance and estimation of $U_a(W_t)$ a lot. It is therefore important to penalize them by using a weighted method. To do this a locally weighted polynomial regression can be used (LOESS). The LOESS function uses a gaussian kernel to do the smoothing. This is simply done by using the `loess` function in `R`. Before applying the function to the observations, the weighting of the least squares needs to be defined first. In Equation 6 it is observed that if $T_t^i - T_t^e \to 0$ then $U_a(W_t) \to \infty$. To penalize extreme values, the weighting will be based on the denominator of Equation 6.

$$\gamma = \frac{\left|T_t^i - T_t^e\right|}{\max\left(T_t^i - T_t^e\right)} \tag{7}$$

From Equation 7 it is clear that for observations with a small difference in the internal and external temperatures will in general be penalised a lot more.

The final thing is to estimate a bandwidth, this is once again done by using cross-validation as seen in Figure 12.

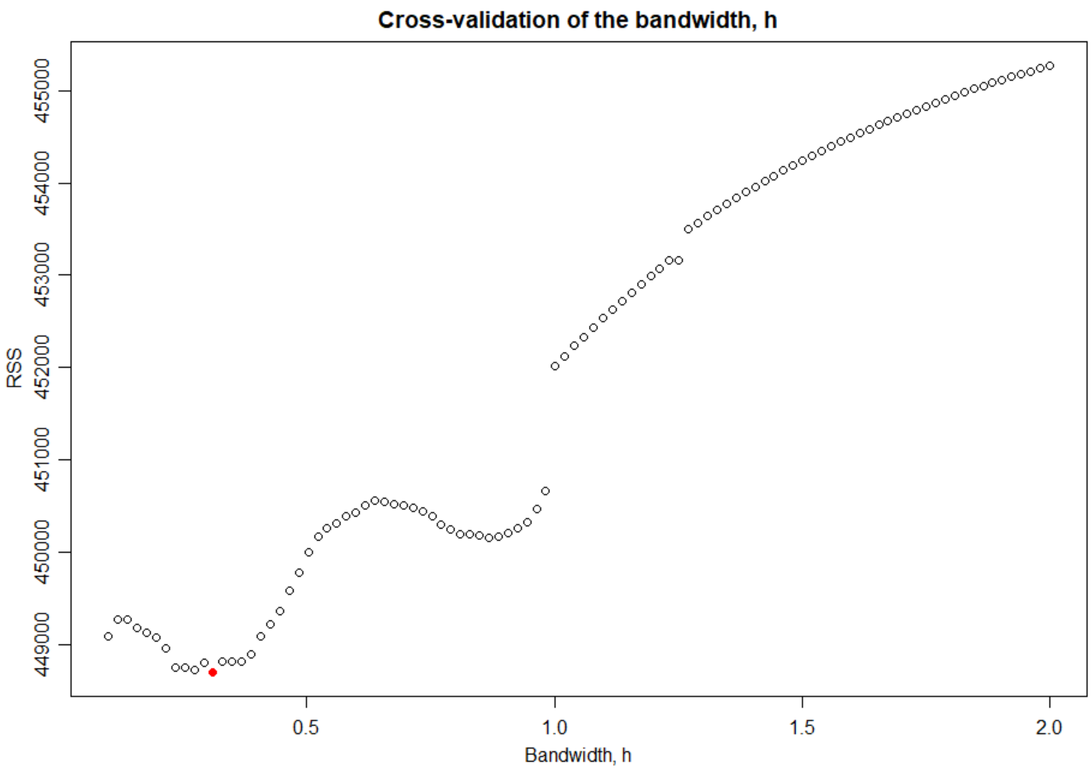

Figure 12: Cross-validation of LOESS function. The optimal bandwidth seems to be $h \approx 0.31$

Now the LOESS can be applied to the data to estimate $U_a(W_t)$. In Figure 13 the estimated $U_a(W_t)$ can be seen. Please note that the plot has been zoomed and the outlier in Figure 11 can't be seen.

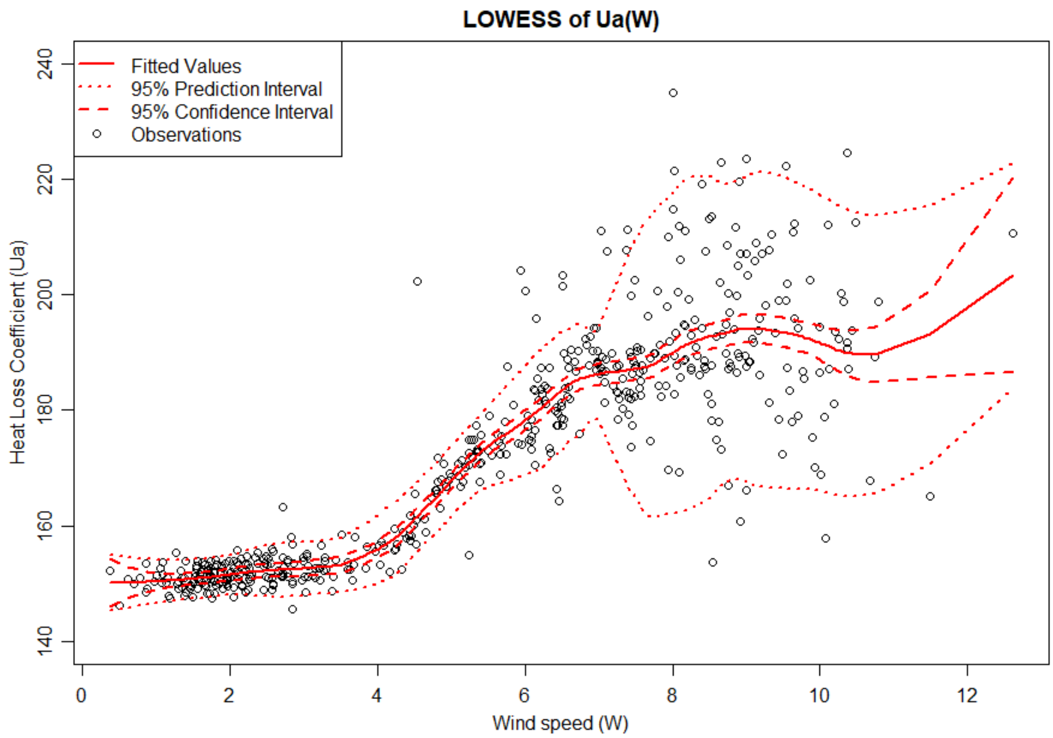

Figure 13: Estimated $U_a(W_t)$ with LOESS.

In Figure 13 the prediction interval is found by using the R package msir [4]. The confidence interval is found by using the predict function in R and calculating the 95% confidence interval by,

$$\hat{Y} \pm t_{\frac{\alpha}{2}} \cdot \sqrt{Var[\epsilon_t]} \tag{8}$$

With t being the t-distribution. Notice that the extreme observations seem to be penalised and will only have a small influence on the estimate. The estimate seems to capture the data well and due to the noisy data when $W \gtrapprox 6.5$ the prediction interval increases.

Question 1.5

Looking at the ACF and PACF of the data given for question 1.5 in Figure 14 it is clear that significant auto-correlation is present.

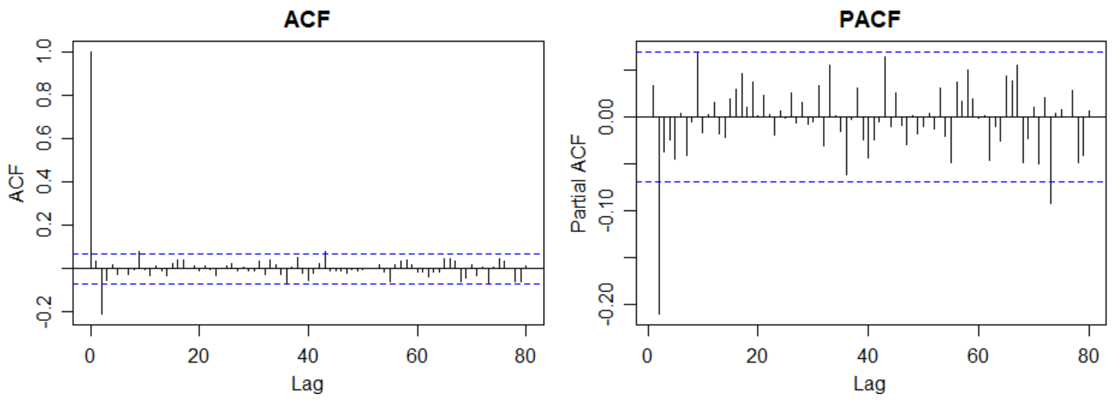

Figure 14: ACF and PACF of data.

Trying to model the data an ARIMA(0,0,2) is used:

$$X_t = 2.2343 - 0.2114\epsilon_{t-2} + \epsilon_t$$

This makes the residuals reasonably white, based on the ACF and PACF in Figure 15.

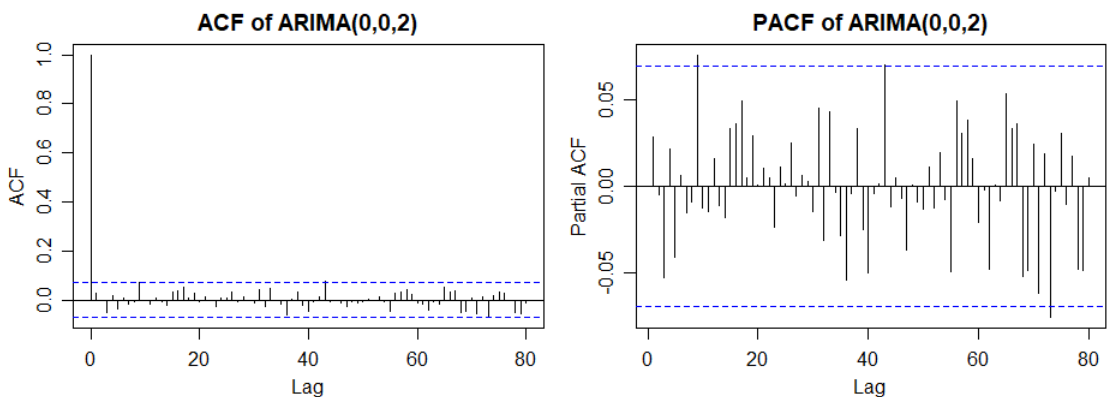

Figure 15: ACF and PACF of ARIMA(0,0,2) model.

However using lag dependent functions (LDF) on the residuals of the MA(2) model gives a different picture. In Figure 16 it can be seen that there are still significant lags, which seem to exponentially decrease every 2. lag. This indicates some non-linearity within the time series data, which is not captured by the MA(2) model.

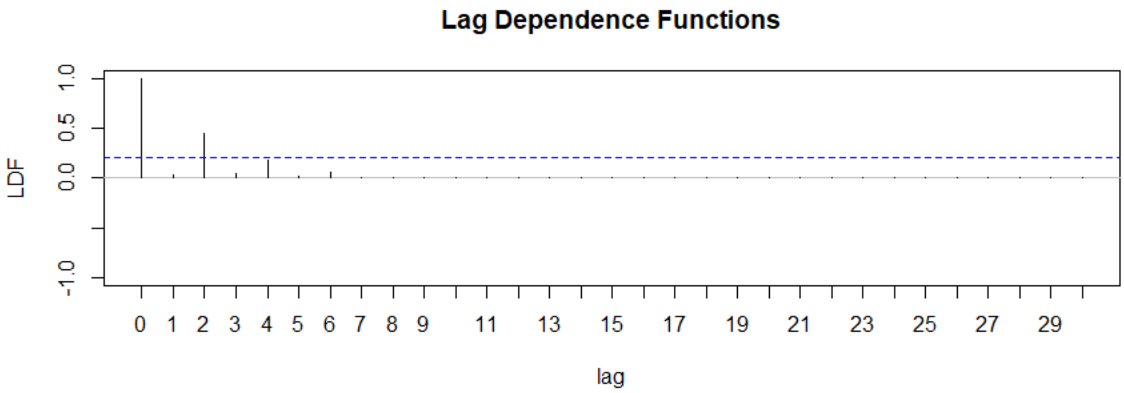

Figure 16

Looking at the residuals versus residuals plot in Figure 17, based on the significant lags from the LDF, non-linearity appear within the residuals. Residuals seem to be divided into regimes.

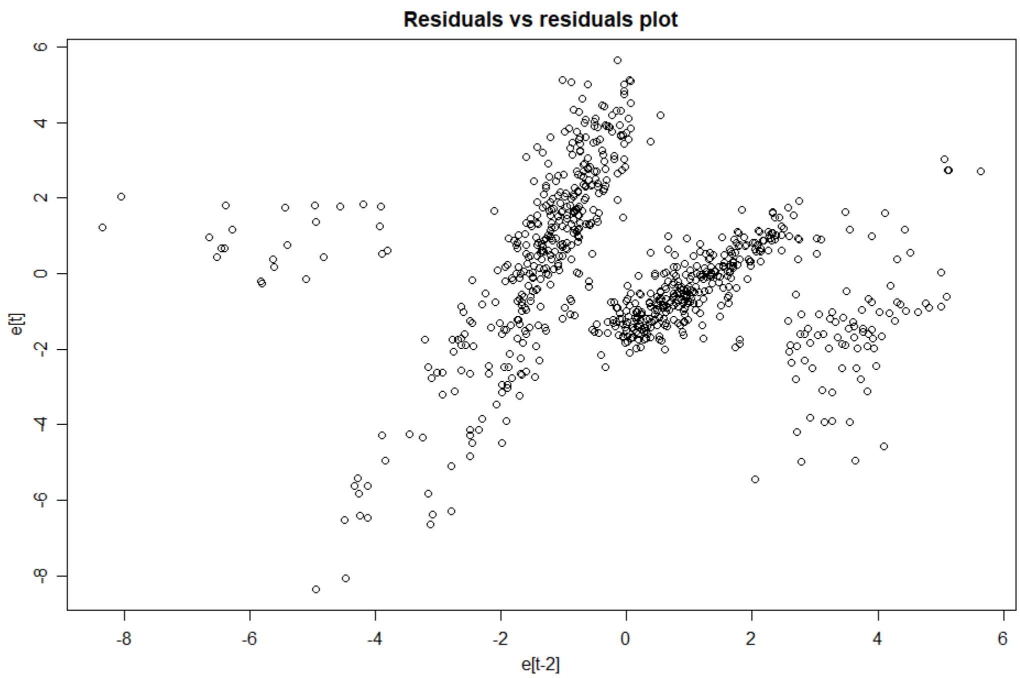

Figure 17: It seems like regimes appear within the residuals, indicating non-linearity.

The time series could therefore be generated by a SETAR(4,2,2) process and in Figure 18 a linear regression has been fitted to each supposedly regime. Where the regime shifts have simply been estimated by a guess based on the plot.

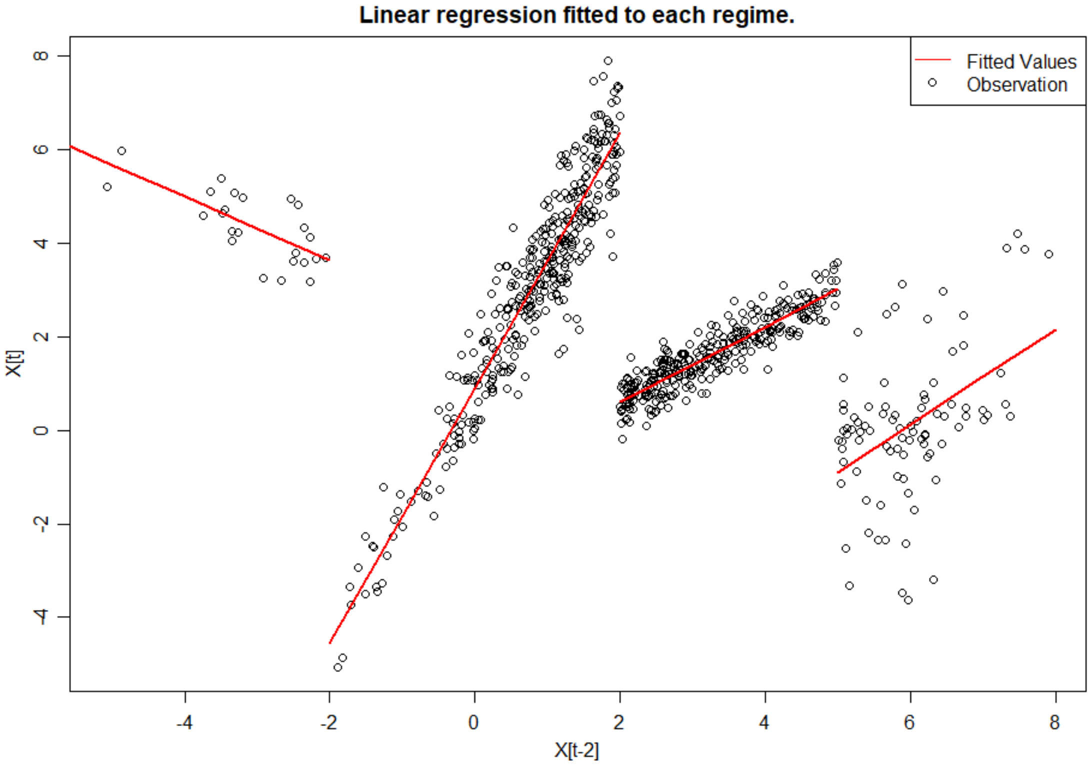

Figure 18: Current observation versus the 2-lagged observation. Linear regressions have been fitted to each cluster of data.

A possible better model structure might therefore be a SETAR(4,2,2) model consisting of only AR-processes. Based on the linear regressions a model could look like the following:

$$X_t = \begin{cases} 2.281 - 0.68X_{t-2} & X_{t-2} \leq -2 \\ 0.9 - 2.734X_{t-2} & -2 < X_{t-2} \leq 2 \\ -1.002 + 0.804X_{t-2} & 2 < X_{t-2} \leq 5 \\ -6 + X_{t-2} & X_{t-2} > 5 \end{cases} \tag{9}$$

Due to few observations in regime 1 and regime 4 it's difficult to see if it's simply noise or how observations have been simulated, the equation for regime 4 might just be noise. Therefore the model in Equation 9 is of course only a rough estimation of the true process and should mainly be seen as an example of the improved model structure. Another model that could be proposed is a non-linear model using kernel smoothers, LOESS. Fitting a LOESS with a span of 0.1 to the data gives the result in Figure 19.

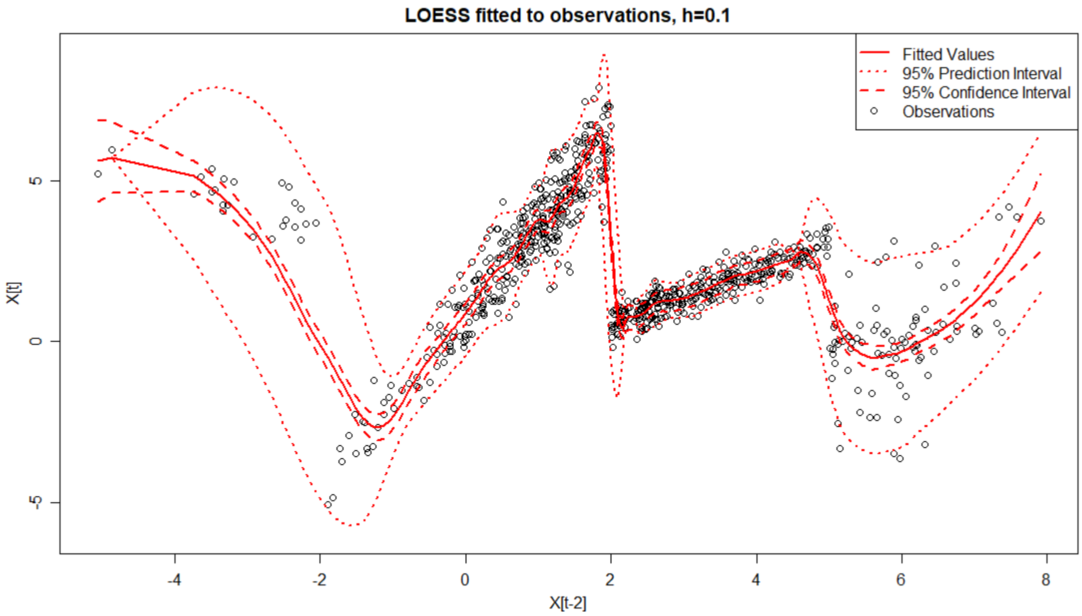

Figure 19: Each regime can be fitted using LOESS. The prediction interval seems to capture the data fairly well, however at places it seems discontinuous or very slim, perhaps because of the low bandwidth, h, used.

Looking at the ACF and PACF of the residuals, Figure 20, from the LOESS model, no significant lags can be found.

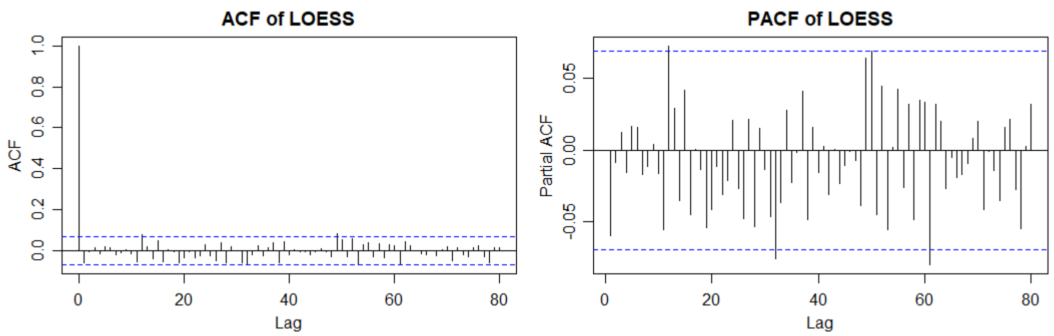

Figure 20: ACF and PACF of LOESS fit.

The same occurs for the LDF, where no lags are found to be significant in Figure 21.

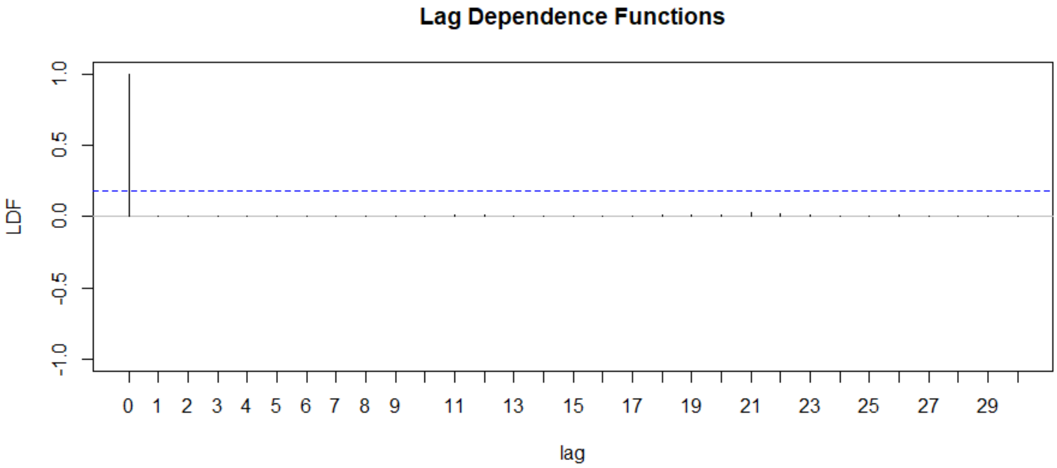

Figure 21: LDF of residuals from LOESS model.

LOESS can therefore also be proposed as a possible candidate for a better model structure.

References

[1] Henrik Madsen, *Time Series Analysis*, Chapman & Hall/CRC, 2008.

[2] Henrik Madsen & Jan Holst, *Modelling Non-Linear and Non-Stationary Time Series*, IMM, 2006, December.

[3] Walter Zucchini and Iain MacDonald, *Hidden Markov Models for Time Series, an introduction using R*, Chapman & Hall/CRC, 2009, Chapter 1.

[4] Model-based SIR for dimension reduction, Luca Scrucca, Computational Statistics & Data Analysis, 2011, 5(11), 3010-3026