ACR2Full

Martin Cavarga

14 February, 2020

Advanced Methods of Time Series Analysis Applied to Quarterly Estimates of Unemployment Rate

Introduction

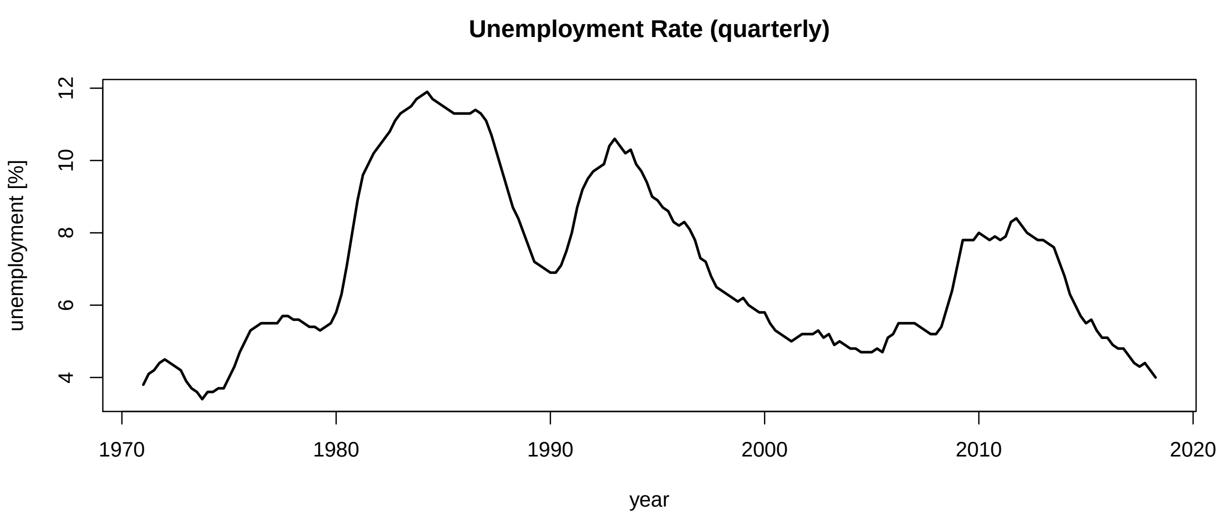
The chosen source of data is the Labour Force Survey (LFS) quarterly estimates of unemployment rate in the UK since March 1971, up to March 2018.

---

1. Elementary Modeling by an AR Process

We begin by extracting the data from a downloaded file

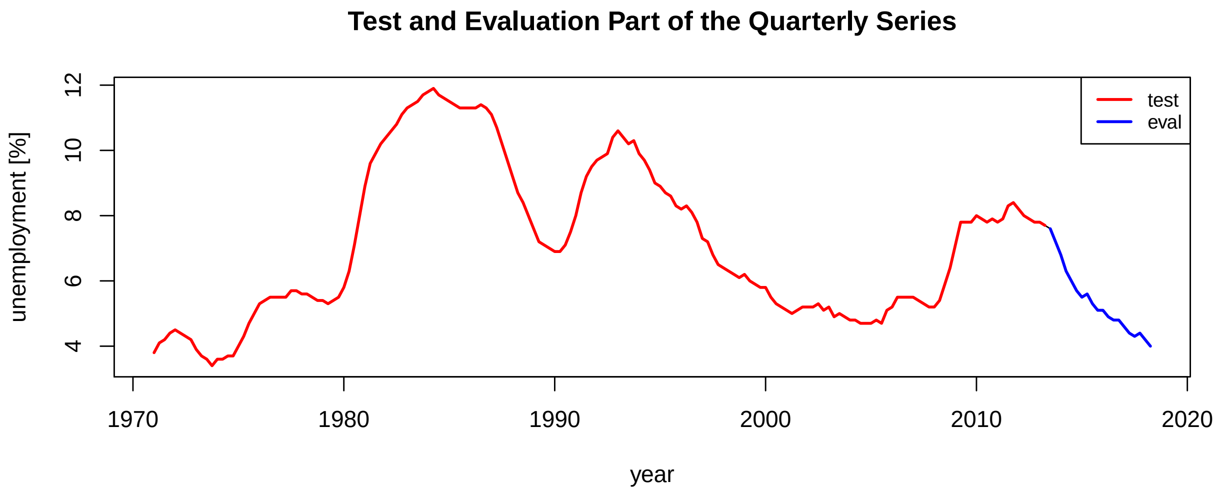
1.1: Data Plot



Now even thought we are working with annual data there should not be any seasonal components or trend since the results do not depend on periodic observable phenomena, but rather the complex economic situation over multiple decades. Also the data may include exponentially decaying decrease in unemployment, but only after year 1980, which would suggest a regime-switching stochastic process.

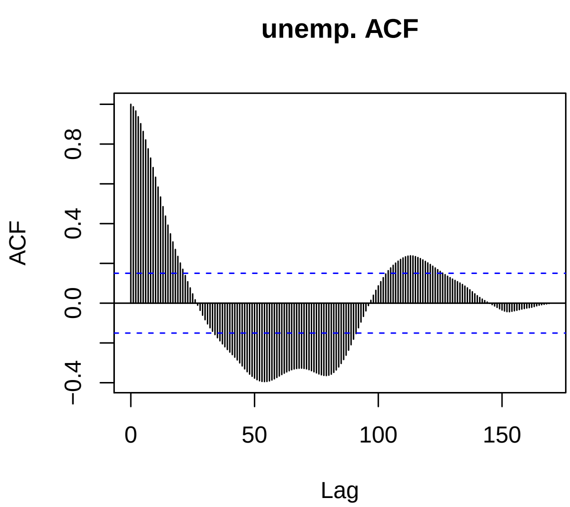
1.2: Test Part and Evaluation Part of the Time Series

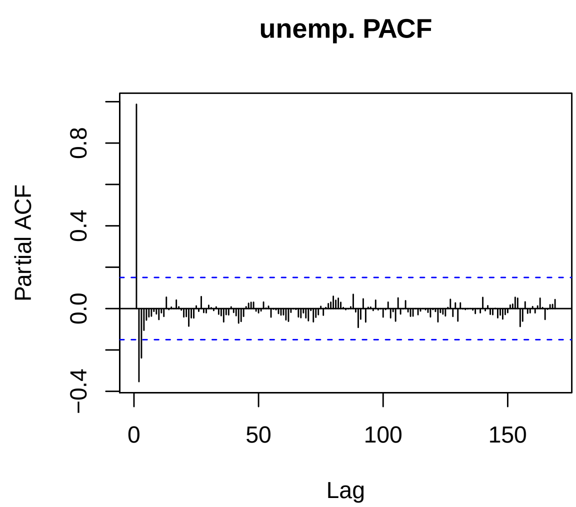
Now we separate the time series into test part, where a suitable model of a stochastic process will be found, and the evaluation part, where predictions given by such model are evaluated. Since our dataset contains quarterly data, we choose the length of the evaluation part of the time series as $L = 4k$ where k is an arbitrary (and sufficiently small) positive integer. We put $k = 5$ for 5 years' length of the evaluation series.



1.3: Mean, Variance, ACF, and PACF of the Test Part

```
##      Min.      Max.      Mean    Median  Variance
##  3.400000 11.900000  7.188824  6.900000  5.661827
```

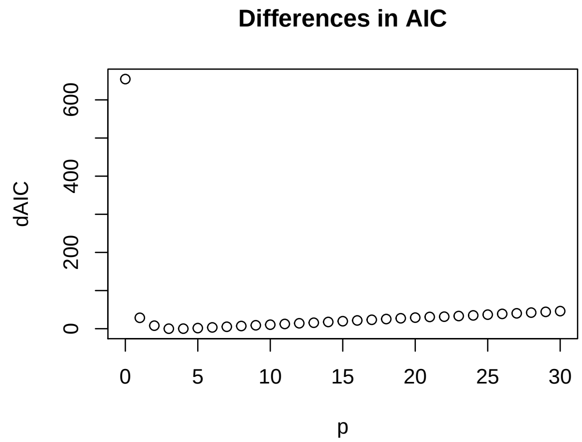


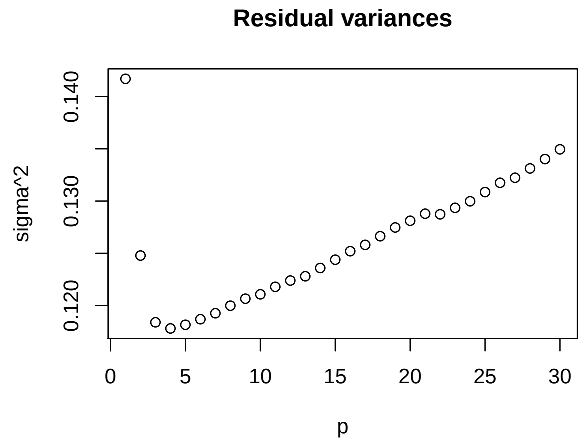


As we mentioned in section 1.1, the underlying process which gave rise to the observed results is aperiodic, yet it is undoubtedly a process with memory. Unemployment rate strongly depends (aside from other important aspects) on its own history which might extend generations into the past. The results are, however, significantly influenced by external phenomena, such as the global economic crisis in late 2000's.

1.4: Finding a Suitable AR Model

Since the economic situation and the job market remembers its past, we choose a simple $AR(p)$ process with parameter p corresponding to the number of steps after which the process still "remembers" its past. We search for the model with the lowest AIC (Akaike's Information Criterion). And by plotting the `AIC` parameter, we obtain differences $AIC_{min} - AIC_k$ for all models.



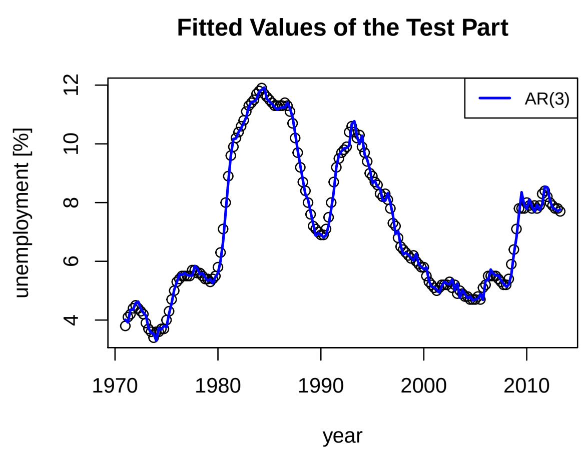


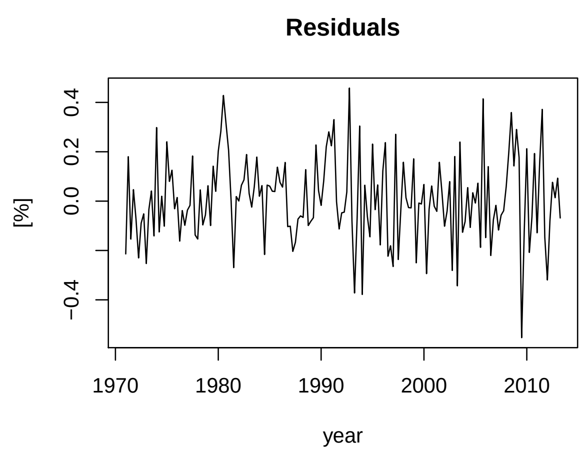
As we can see in the figures, the lowest variance of residues corresponds to an $AR(3)$ process:

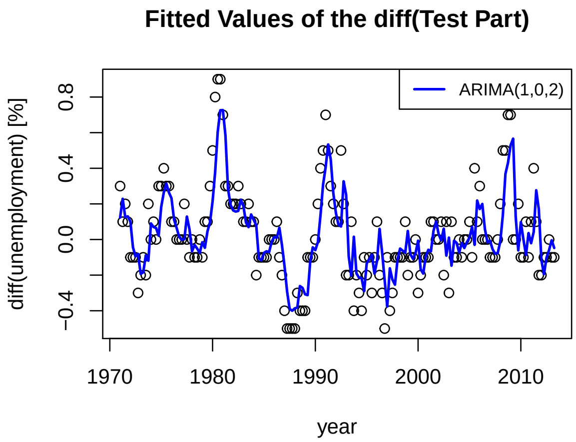
$$X_t = (1.252 \pm 0.075)X_{t-1} + (-0.034 \pm 0.123)X_{t-2} + (-0.238 \pm 0.075)X_{t-3} + \varepsilon_t \ , \quad \hat{\sigma}_\varepsilon = 0.0358847$$

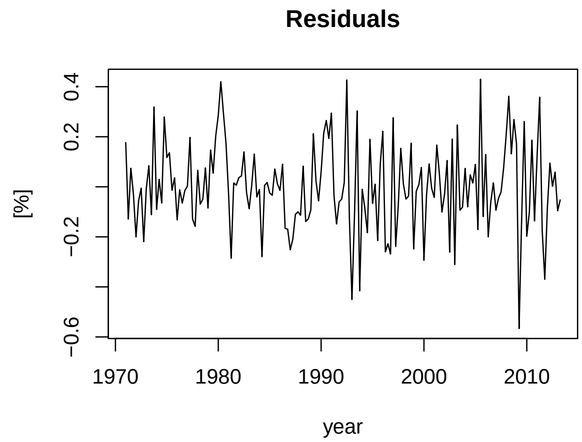
While testing the validity of our linear model, we can easily use an automatic search procedure `auto.arima()` from the `forecast` package:

Which shows that the best fit would be produced by an $ARIMA(1, 1, 2)$. The fact that the raw data is likely integrated with $d = 1$ suggests that we should consider modeling 1st differences of the series instead.



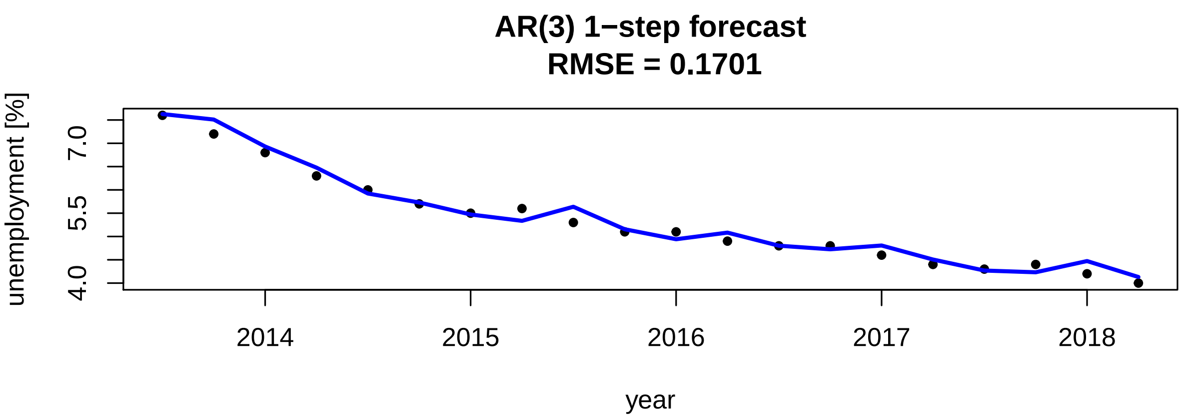






When we do, in fact, take differences of the time series (as shown on the bottom plot), an ARIMA search procedure returns model $ARIMA(1, 0, 2)$ as a best fit.

1.5: 1-Step Predictions Over the Evaluation Part



1.6.: Conclusion

Since it has very low rate of local oscillation, but does not have an easily predictable systematic pattern, the analyzed unemployment rate time series seems to be well-estimated by an $AR(3)$ process with low prediction errors. Automatic search procedure `auto.arima()` determined the best fit to be an $ARIMA(1, 1, 2)$, i.e. we are most likely dealing with an integrated process of first order. Hence, from now on, we will carry out our analysis on first differences of the series.

However, as we mentioned earlier we might be dealing with a 'regime-switching' process, which will be examined by suitable linearity/nonlinearity tests in the following chapters.

---

2. Finding the Parameters of a SETAR Model

The unemployment time series might be a result of a regime-switching process. Naturally, the behavior of the unemployment rate in a given country should depend on the current economic situation. The change in the local economy can be described, for example, via a set of "thresholds" which determine whether the stochastic process changes its regime. The regime of a stochastic process is defined by a unique ARMA or any other linear stochastic process with unique parameters.

Since we are dealing with differences, i.e. rate of change of unemployment rate, we can also interpret the model regimes as regimes of "growth" and "decrease" in unemployment, since undoubtedly, higher unemployment rate induces even more unemployment due to the lack of job opportunities.

We begin by finding the parameters of a Self-Exciting Threshold Autoregressive (SETAR) process, that is: a process whose regime is described by a random variable determined by the very process itself, more specifically its history of up to d steps behind, which in an essence means that the process "influences its regime" up to d time steps into the future.

For the purposes of this analysis we consider only 2 regimes, namely the regime of "job crisis" when the unemployment rate may fluctuate or drop more wildly compared to the regime of "job stability" when the unemployment rate stabilizes or grows.

2.1: Useful Functions

First, we define a, so called, "indicator function" which essentially returns a boolean value from a given input process value `x` and threshold value `c`:

$$I(z_t, c) = \begin{cases} 0 & \text{if } z_t \le c \\ 1 & \text{if } z_t > c \end{cases}$$

Afterwards, we define the basis vector for a single regime as:

$$Y_t = (1, X_{t-1}, X_{t-2})^\top$$

which can then be used in the basis for two regimes

Then we need a deterministic skeleton of the model:

$$F(z_t, \theta) = \begin{cases} \phi_{1,0} + \phi_{1,1} X_{t-1} + ... + \phi_{1,p_1} X_{t-p_1} & \text{if } z_t \le c \\ \phi_{2,0} + \phi_{2,1} X_{t-1} + ... + \phi_{2,p_2} X_{t-p_2} & \text{if } z_t > c \end{cases}$$

and the last group of functions we need for the upcoming procedure are functions for the information criteria of a SETAR model:

$$AIC_{SETAR} = \sum_{j=1}^{m} (n_j \log \hat{\sigma}_{\varepsilon,j} + 2(p_j + 1)) \ , \quad BIC_{SETAR} = \sum_{j=1}^{m} (n_j \log \hat{\sigma}_{\varepsilon,j} + \log n_j (p_j + 1)) \ , \ m = 2$$

2.2: The Estimation of Parameters of a SETAR Model

Given a dataset `x` and parameters `p` (AR order), `d` (SETAR delay), and the threshold `c` we find the coefficients of a SETAR model with these parameters by performing a multivariate linear regression (a custom `EstimSETAR` method). The coefficient vector `PhiParams` is the vector of unknowns of a linear system with matrix $\mathbf{X}$ and a right-hand-side vector $\mathbf{y}$ given by the time series. Although for higher values of `p` the inversion of matrix $\mathbf{X}^\top \mathbf{X}$ (with dimensions $(2p+2) \times (2p+2)$) might be computationally demanding, we will determine the covariance matrix, i.e.: $(\mathbf{X}^\top \mathbf{X})^{-1}$ using a function `inv` from the `matlib` package.

After performing this procedure for multiple parameters, i.e.: searching the discrete parameter space, we further process the model with minimum residual square sum. For that we'll use a postprocessing method, in which we determine the following model attributes: n_1, n_2 (regime data counts), $\hat{\sigma}_{\varepsilon,1}$, $\hat{\sigma}_{\varepsilon,2}$ (regime residual variances), and information criteria (AIC, BIC).

The model should contain the following attributes in total:

```
## List of 17
##  $ p          : num 2
##  $ d          : num 2
##  $ c          : num 0
##  $ data       : num [1:170] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 170
##  $ PhiParams  : num [1:6] 0 0.4124 0.4661 0.0692 0.8179 ...
##  $ PhiStErrors: num [1:6] 0 0 0.0534 0.0694 0.0155 0.0448 ...
##  $ residuals  : num [1:168, 1] 0.1002 -0.1157 -0.2168 -0.0703 -0.0121 ...
##  $ resSigmaSq : num 0.028
##  $ name       : chr "SETAR(2,2,0)"
##  $ nReg       : num 2
##  $ n1         : num 102
##  $ n2         : num 68
##  $ resSigmaSq1: num 0.0213
##  $ resSigmaSq2: num 0.039
##  $ AIC        : num -597
##  $ BIC        : num -578
```

It should be noted that for some values of `p` and `d` the indices of arrays in the algorithms might get out of the range of regularity for the linear system. For that reason we implement exceptions for the outputs of `EstimSETAR` in the following algorithm.

2.3: SETAR Parameter Estimation Procedure

To answer the question: 'how does one find the right parameters `p`, `d` and `c` for their desired SETAR model?', we implement a procedure for a grid of sampled parameters:

$$p = 1, 2, 3, 4, 5, 6, 7\ ,\ d = 1, ..., \ p \quad c = -0.3, -0.2927, -0.2853...0.4032, 0.4105, 0.4179, 0.4252$$

After iterating through thresholds c, we pick the model with the lowest residual variance.

After we have a set of models in their original, we choose 12 models with the lowest BIC (Bayesian Information Criterion):

```
##    p d         c  n1  n2       AIC       BIC resSigmaSq
## 3  2 2  0.102875 129  41 -616.5615 -602.8413 0.02624694
## 5  3 2  0.102875 129  41 -606.5288 -588.2352 0.02668915
## 4  3 1  0.300650 153  17 -601.5380 -586.0834 0.02777303
## 1  1 1  0.000325 102  68 -595.3267 -585.6377 0.02955918
## 2  2 1  0.205425 142  28 -596.8388 -583.9747 0.02908718
## 8  4 2  0.102875 129  41 -600.5090 -577.6421 0.02681811
## 7  4 1  0.300650 153  17 -595.8224 -576.5041 0.02800968
## 10 4 4  0.205425 142  28 -597.1284 -575.6882 0.02698219
## 6  3 3 -0.197450  34 136 -592.0333 -574.2772 0.02778604
## 15 5 5  0.102875 129  41 -600.5783 -573.1380 0.02542239
## 14 5 4  0.205425 142  28 -597.2758 -571.5476 0.02618782
## 12 5 2  0.102875 129  41 -598.8347 -571.3944 0.02590032
```

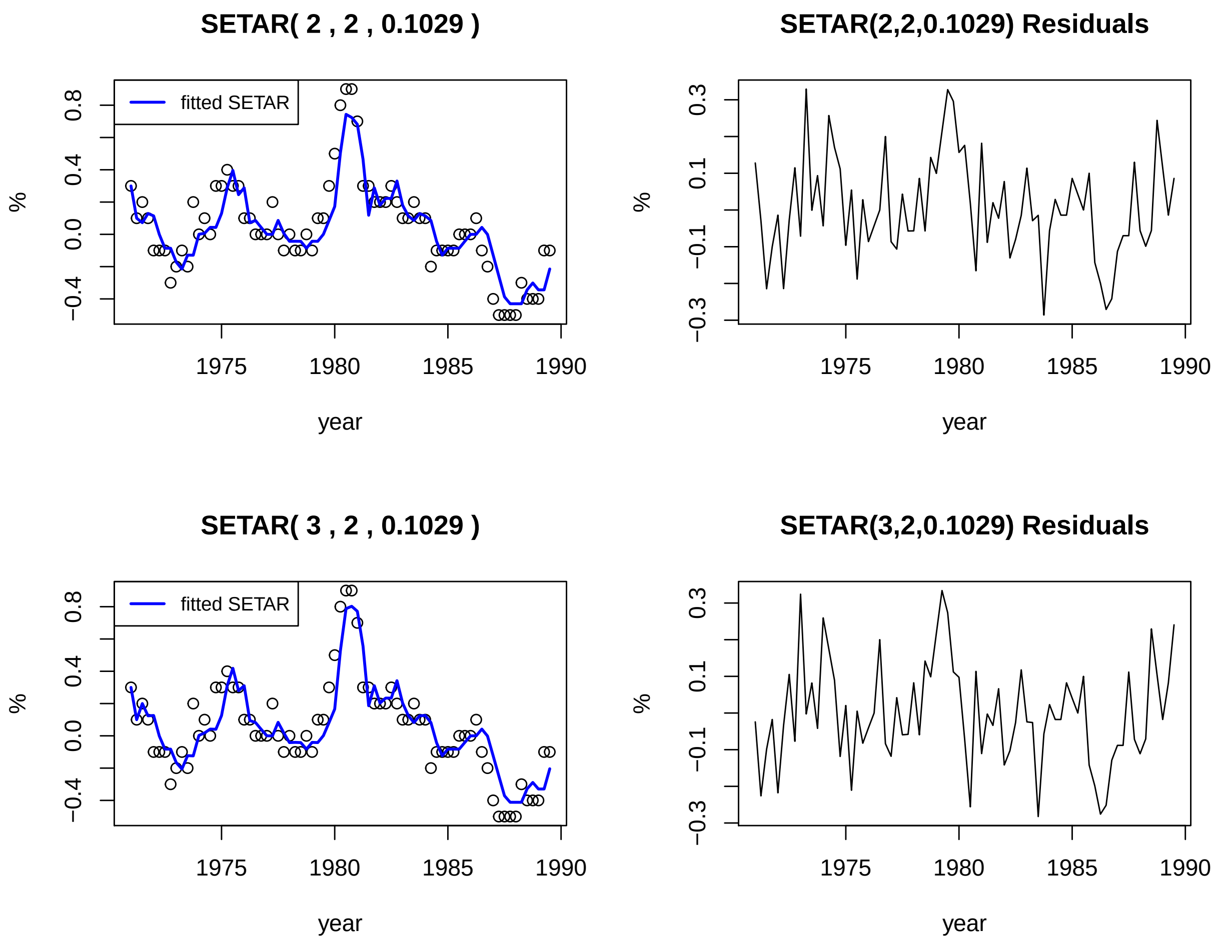
and we can also include standard errors of the estimated regression coefficients (of at least the first 3 models):

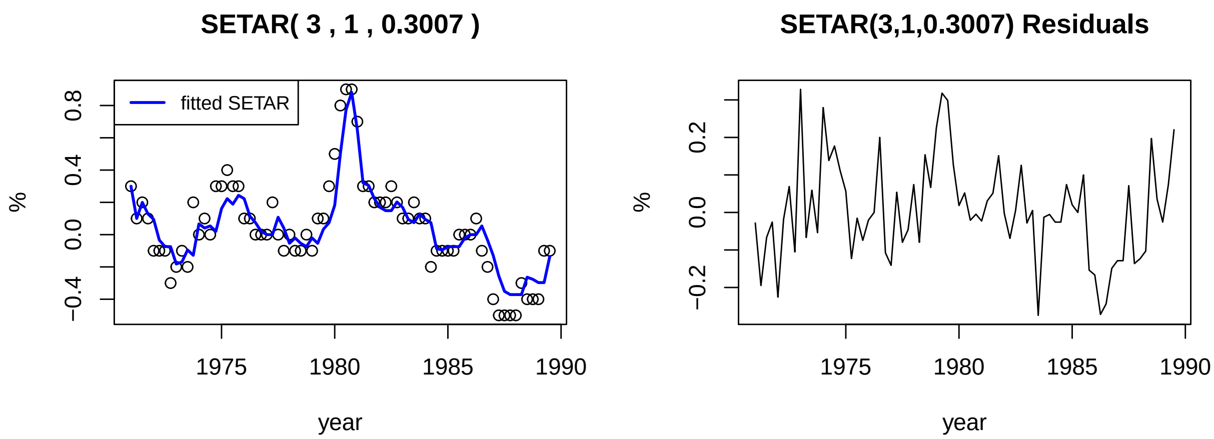
```
## $`SETAR(2,2,0.103)`
##          [,1]       [,2]       [,3]       [,4]       [,5]        [,6]
## Phi         0 0.43170625 0.42834406 0.09138806 1.08026643 -0.42454326
## stdError    0 0.04164731 0.05038936 0.02581381 0.05901015  0.07802637
##
## $`SETAR(3,2,0.103)`
##          [,1]       [,2]       [,3] [,4]       [,5]       [,6]        [,7] [,8]
## Phi         0 0.41811454 0.40547459    0 0.08019403 1.07869738 -0.31056937    0
## stdError    0 0.04340129 0.05441896    0 0.02619414 0.05963855  0.09969198    0
```

```
##
## $`SETAR(3,1,0.301)`
##           [,1]       [,2]       [,3] [,4]       [,5]      [,6]      [,7]
## Phi          0 0.53829523 0.20456086    0 -0.1620930 1.1391822 0.6438414
## stdError     0 0.04368799 0.04233384    0  0.0792829 0.1808141 0.1619481
##                 [,8]
## Phi       -0.9934059
## stdError   0.1556738
```

Note, that we intentionally set coefficients, whose standard errors exceed half of their estimated value, to zero.

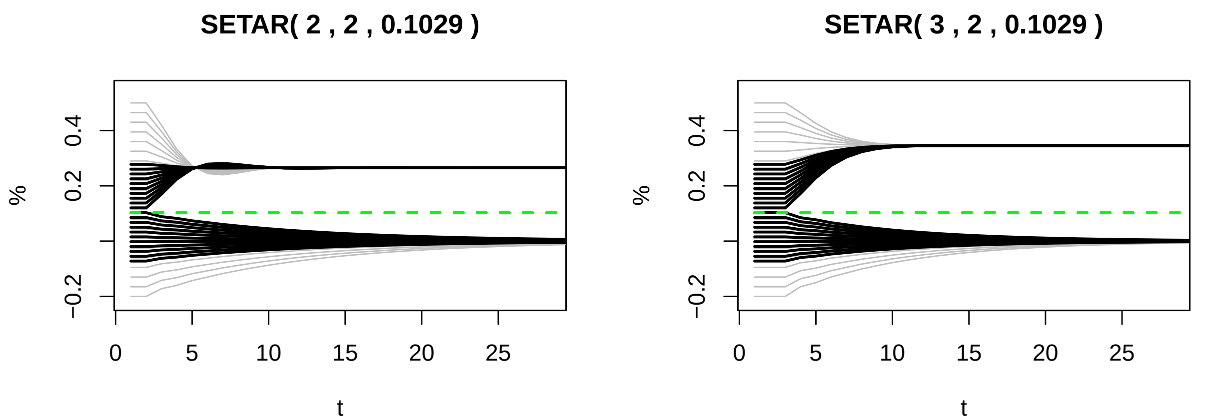
We can now visualize the results of the top 3 models:

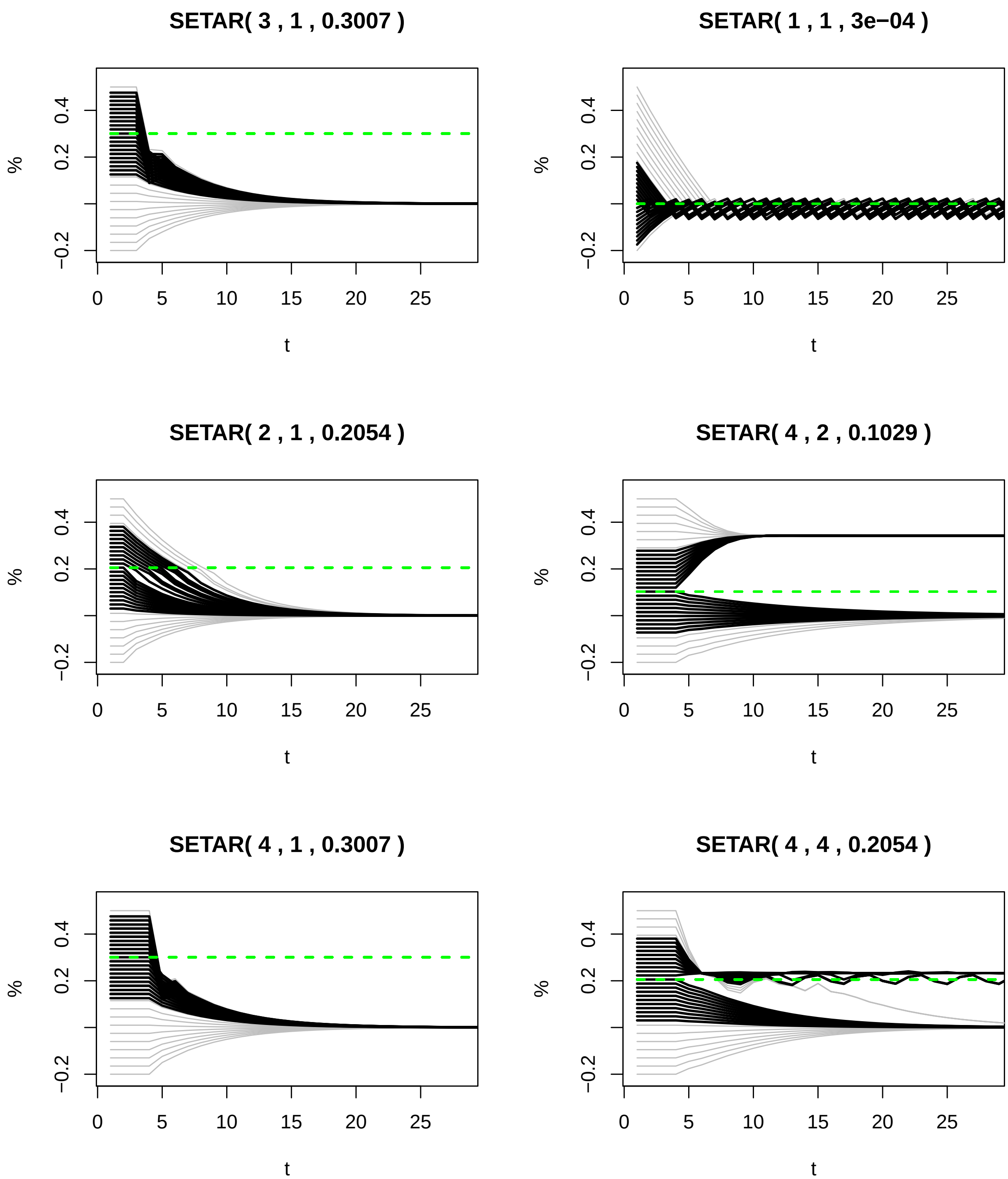


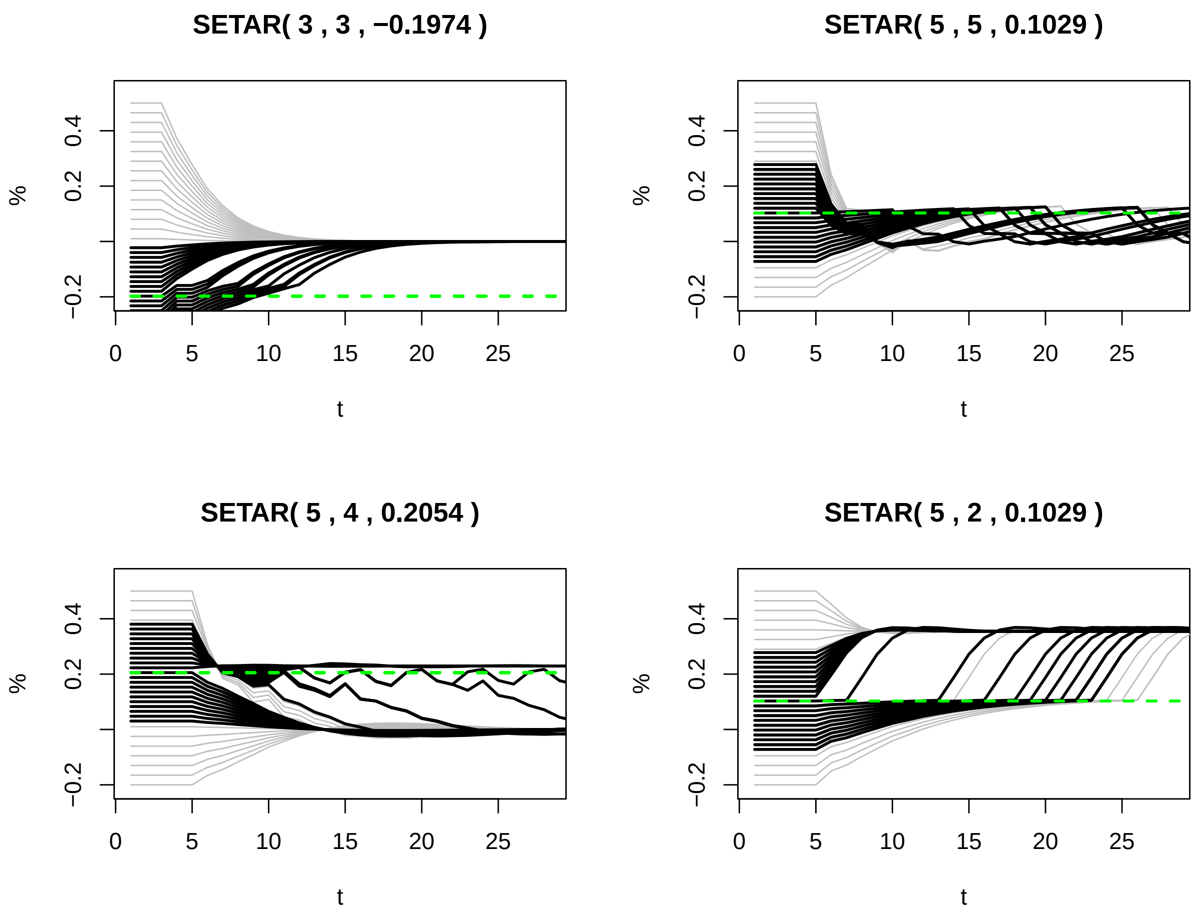


2.4: SETAR Equilibria and Equilibrium Simulations

It is also essential to find out whether the skeletons of the selected SETAR models have some equilibria. The estimation of the exact equilibria of the piecewise-linear skeletons with $p = 1$ is straightforward: We find the fixed points of the skeletons by finding the intersections between their graphs and the identity line $\mathrm{id}x = x$, given the model parameters (coefficients). However, the results of our search have mostly higher AR degrees, thus we will need to determine the models' equilibria using a more general method, namely letting the model skeletons evolve with multiple input initial conditions.




SETAR(3 , 1 , 0.3007)
SETAR(1 , 1 , 3e−04)
SETAR(2 , 1 , 0.2054)
SETAR(4 , 2 , 0.1029)
SETAR(4 , 1 , 0.3007)
SETAR(4 , 4 , 0.2054)
%
t
−0.2
0.2
0.4
0
5
10
15
20
25

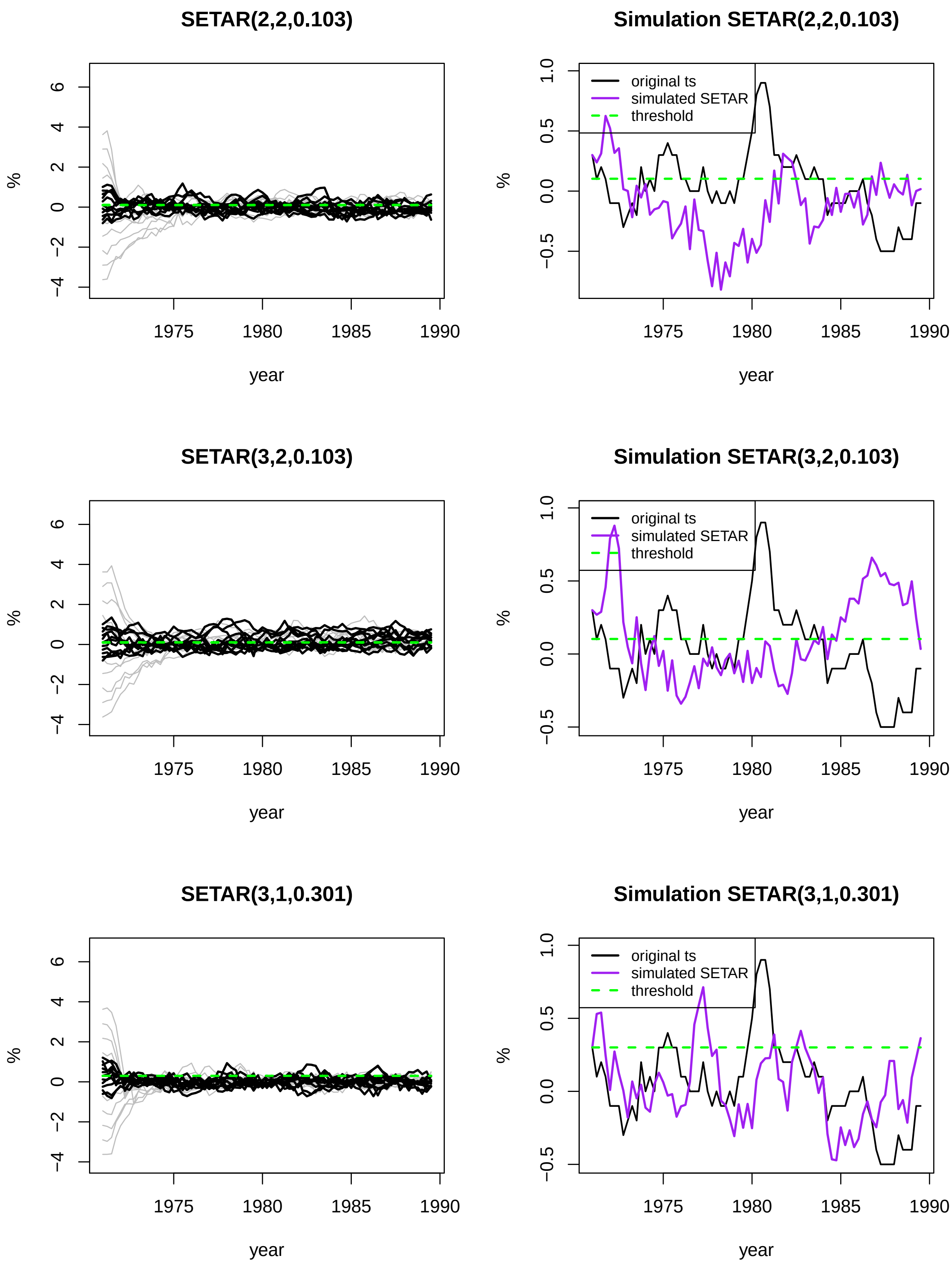



```
## $`SETAR( 5 , 2 , 0.103 ) equilibria`
## [1] 0.0000 0.2654
##
## $`SETAR( 5 , 2 , 0.103 ) equilibria`
## [1] 0.0000 0.3459
##
## $`SETAR( 5 , 2 , 0.103 ) equilibria`
## [1] 0
```

Note that we generated an additional set of trajectories (black) close to the threshold *c*.

As we see, the trajectories of the top 3 models gravitate towards 0 in all models, but in the first and second model they can end up in one more position, close to zero. Fourth, eight and the following models seem to have a limit cycle behavior, in which their values can spontaneously switch to another equilibrium, and even oscillate between them.

It might also be interesting to see how the trajectories evolve when we add an iid noise on top of the model skeleton. We carry out multiple simulations with initial conditions close to the threshold. The added noise will have the same deviance as the residual square sum.



The trajectories of all of the first three models seem to gravitate toward 0 significantly fast (or alternatively: towards their threshold values which are close to zero as well). The relatively low oscillation rate of the original time series suggests that the differences of this time series will, at most, fluctuate around 0. The change between the 'high' and 'low' regimes does not seem very significant, at leat on the larger scale. The validity of the model will be tested in chapter 3.

2.6: Conclusion

The results of the SETAR Parameter Estimation Procedure in section 2.3 show that the 3 best 2-regime SETAR models are:

```
##     unlist.results.
## 1 SETAR(2,2,0.103)
## 2 SETAR(3,2,0.103)
## 3 SETAR(3,1,0.301)
```

The first model with the lowest `BIC` (Bayesian Information Criterion) has the most accurate estimation of its 4 regression parameters:

$$X_t = \begin{cases} (0.4317 \pm 0.0416)X_{t-1} + (0.4283 \pm 0.0504)X_{t-2} + \varepsilon_t & \text{if } X_{t-2} \leq 0.1029 \\ (0.0914 \pm 0.0258) + (1.0803 \pm 0.059)X_{t-1} + (-0.4245 \pm 0.078)X_{t-2} + \varepsilon_t & \text{if } X_{t-2} > 0.1029 \end{cases}$$

$\hat{\sigma}_\varepsilon^2 = 0.0262$, $\hat{\sigma}_{\varepsilon,1} = 0.0202$, $\hat{\sigma}_{\varepsilon,2} = 0.0474$

3: Tests of Linearity/Nonlinearity of SETAR models

We need to make sure a non-linear model (SETAR, for example) is really suitable for describing the process. In order to find out, we test the null hypothesis that a linear model is more suitable than a non-linear one. In the case of a 2-regime model we are looking for, so called, nuisance parameters, i.e.: $H_0 : \Phi_1 = \Phi_2$ where Φ_1 and Φ_2 are the parameters of the low and the high regime respectively.

3.1: Hansen's Conditions

Hansen proposed three conditions to test whether a SETAR model can be tested for linearity using the so called Likelihood-Ratio (LR) test.

If all three of Hansen's conditions are satisfied the model can be tested using a specialized LR test.

```
##                    cond1 cond2 cond3
## SETAR(2,2,0.103)  FALSE  TRUE FALSE
## SETAR(3,2,0.103)  FALSE  TRUE FALSE
## SETAR(3,1,0.301)  FALSE  TRUE FALSE
## SETAR(1,1,0)       TRUE  TRUE  TRUE
## SETAR(2,1,0.205)   TRUE  TRUE  TRUE
## SETAR(4,2,0.103)  FALSE  TRUE FALSE
## SETAR(4,1,0.301)  FALSE  TRUE FALSE
## SETAR(4,4,0.205)  FALSE  TRUE FALSE
## SETAR(3,3,-0.197)  TRUE  TRUE  TRUE
## SETAR(5,5,0.103)  FALSE  TRUE FALSE
## SETAR(5,4,0.205)  FALSE  TRUE FALSE
## SETAR(5,2,0.103)  FALSE  TRUE FALSE
```

It appears that only the fourth, fifth, and the nineth model can be tested using the LR test. The rest will have to be assessed using the Lagrange Multiplier (LM) test.

3.2: LR and LM Tests

In this section we perform the basic procedures for the LR (Likelihood Ratio), and LM (Lagrange Multiplier) tests, and afterward we interpet the results:

```
##                model Hansen Cond.      TS      CV p-value linearity
## 3   SETAR(2,2,0.103)        FALSE 16.9682  7.8147   7e-04  rejected
## 5   SETAR(3,2,0.103)        FALSE 16.3438  9.4877  0.0026  rejected
## 4   SETAR(3,1,0.301)        FALSE  8.3517  9.4877  0.0795  accepted
## 1       SETAR(1,1,0)         TRUE -2.1356  7.6873   0.797  accepted
## 2   SETAR(2,1,0.205)         TRUE   0.613  7.6873   0.354  accepted
## 8   SETAR(4,2,0.103)        FALSE 16.2147 11.0705  0.0063  rejected
## 7   SETAR(4,1,0.301)        FALSE 12.6877 11.0705  0.0265  rejected
## 10  SETAR(4,4,0.205)        FALSE 10.1737 11.0705  0.0705  accepted
## 6  SETAR(3,3,-0.197)         TRUE  8.1902  7.6873  0.0449  rejected
## 15  SETAR(5,5,0.103)        FALSE 15.6153 12.5916   0.016  rejected
## 14  SETAR(5,4,0.205)        FALSE 10.9191 12.5916  0.0909  accepted
## 12  SETAR(5,2,0.103)        FALSE  17.916 12.5916  0.0064  rejected
```

For significance level `alpha = 0.05` the linearity hypothesis is not rejected only for the following models:

```
## [1] "SETAR(3,2,0.103)"  "SETAR(4,2,0.103)"  "SETAR(4,1,0.301)"
## [4] "SETAR(3,3,-0.197)" "SETAR(5,5,0.103)"  "SETAR(5,2,0.103)"
```

3.3 Modified LR Test Via Boostrapping

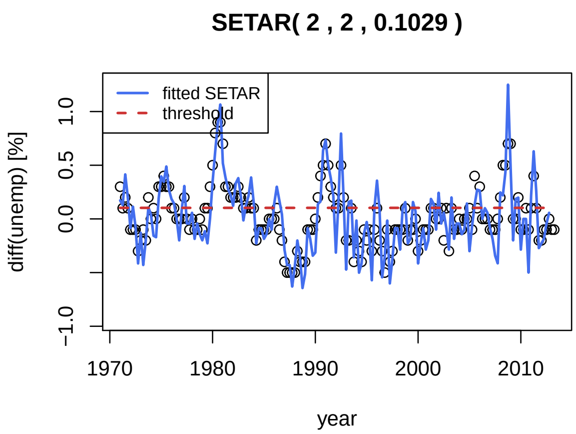
The proposed LR test has a significant drawback in the fact that it can only be done when Hansen's conditions are satisfied. This is due to the fact that we do not know the distribution of the resulting F-statistic. According to Hansen (1996), however, the distribution of a bootstrapped statistic F^* converges weakly in probability to the distribution of F, so that repeated bootstrap draws from F^* can be used to approximate the asymptotic distribution of F.

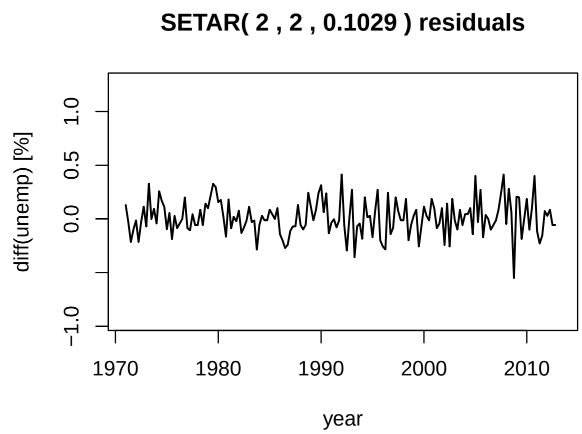
```
##                model      TV      CV  p-value        time
## 1   SETAR(2,2,0.103) 24.9116 12.8744        0 *** 6.87 s
## 2   SETAR(3,2,0.103) 25.4032 13.6361        0 *** 6.37 s
## 3   SETAR(3,1,0.301) 11.9899 22.8584      0.5     6.15 s
## 4       SETAR(1,1,0) -2.1356  7.6873 0.797037        0 s
## 5   SETAR(2,1,0.205)   0.613  7.6873 0.353984        0 s
## 6   SETAR(4,2,0.103) 25.5851   9.039        0 *** 6.63 s
## 7   SETAR(4,1,0.301) 25.5795 13.4306        0 *** 6.71 s
## 8   SETAR(4,4,0.205) 25.5851 17.2471        0 *** 6.69 s
## 9  SETAR(3,3,-0.197)  8.1902  7.6873 0.044858   *    0 s
## 10  SETAR(5,5,0.103)  30.371 27.2061        0 *** 7.22 s
## 11  SETAR(5,4,0.205)  30.371 36.1615      0.5     7.18 s
## 12  SETAR(5,2,0.103) 27.6078 22.2637        0 *** 7.15 s
```

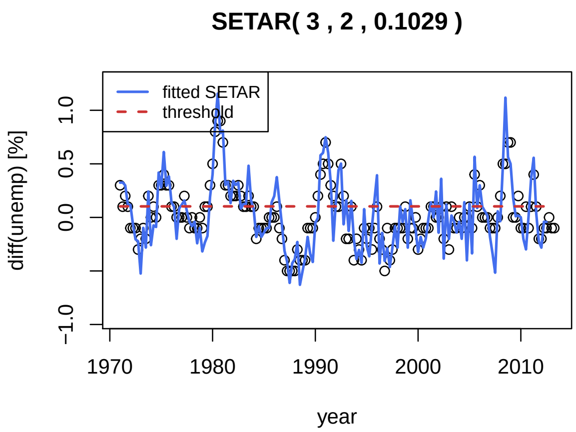
In the above table, we list the results of the modified LR test with significance codes `.` (`p-value` $\in [0.05, 0.1)$), `*` (`p-value` $\in [0.01, 0.05)$), `**` (`p-value` $\in [0.001, 0.01)$), and `***` (`p-value` \$ < 0.001\$), and with computation time denoted in the `time` column.

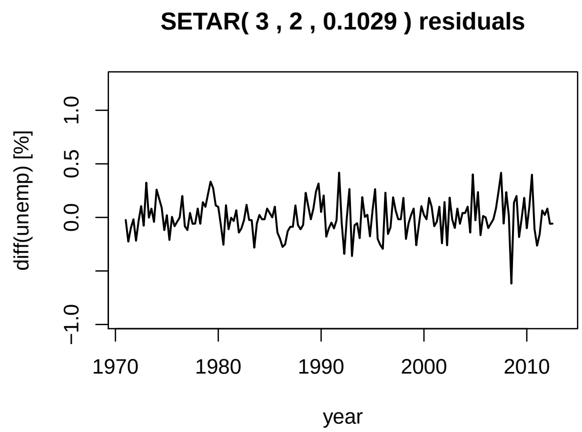
3.4 Visualisation of Non-Linear Models

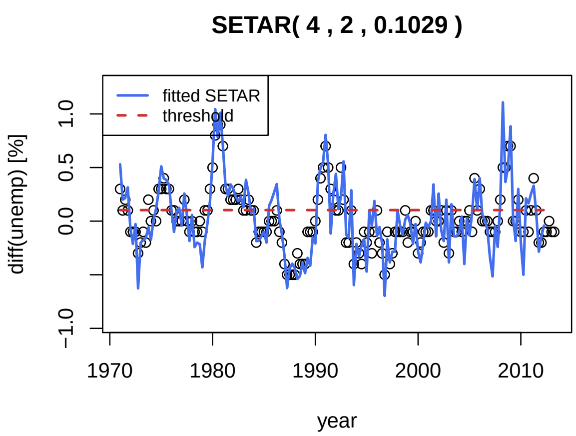
From the results of the previous procedure, we will visualize the models for which the linearity null-hypothesis was rejected based on the LR and LM tests:

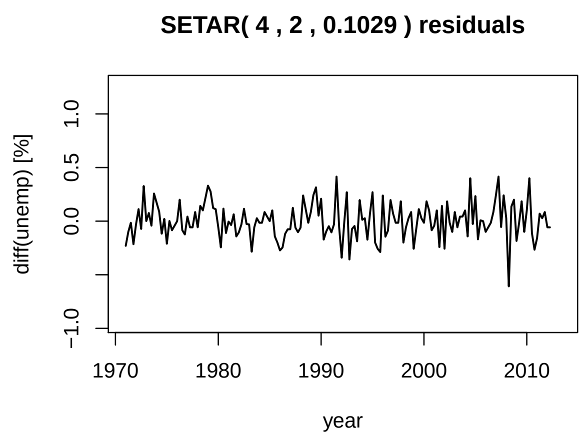

SETAR(2 , 2 , 0.1029)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

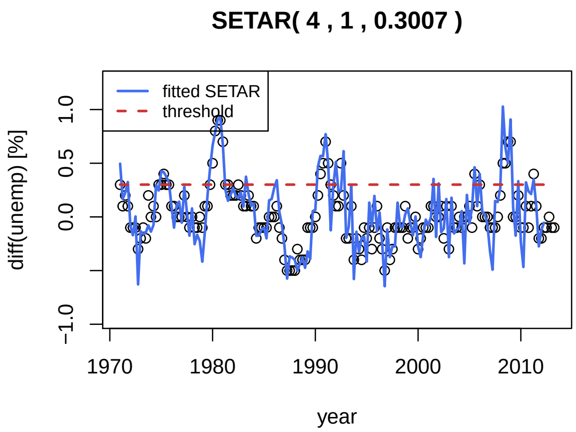

SETAR(2 , 2 , 0.1029) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

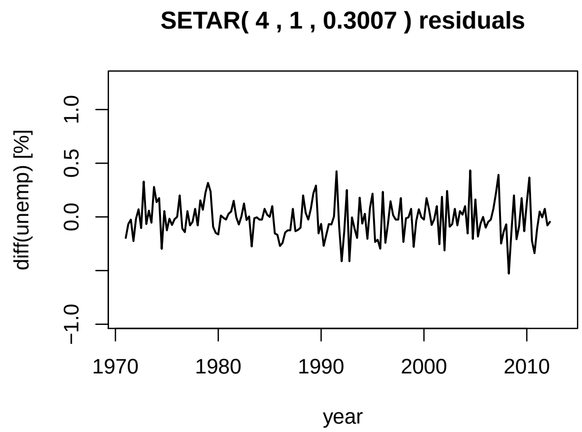

SETAR(3 , 2 , 0.1029)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

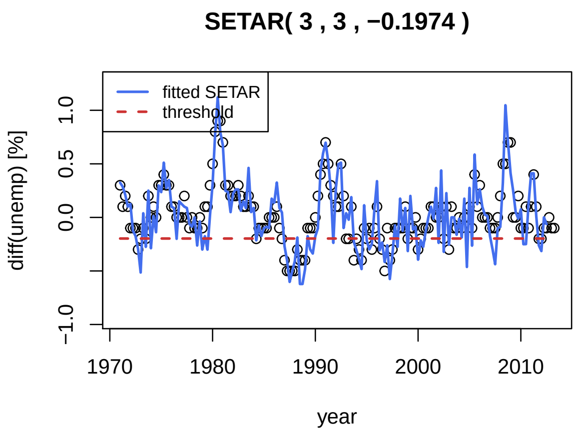

SETAR(3 , 2 , 0.1029) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

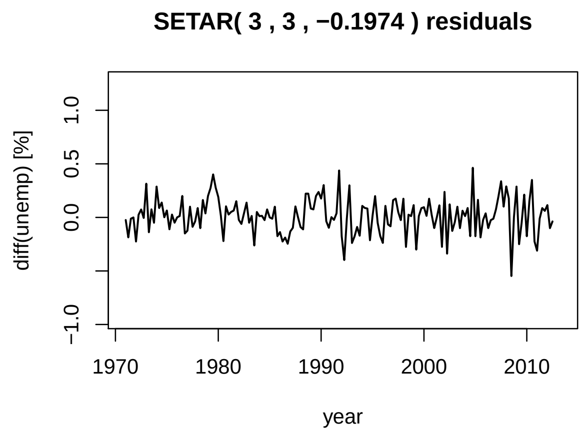

SETAR(4 , 2 , 0.1029)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

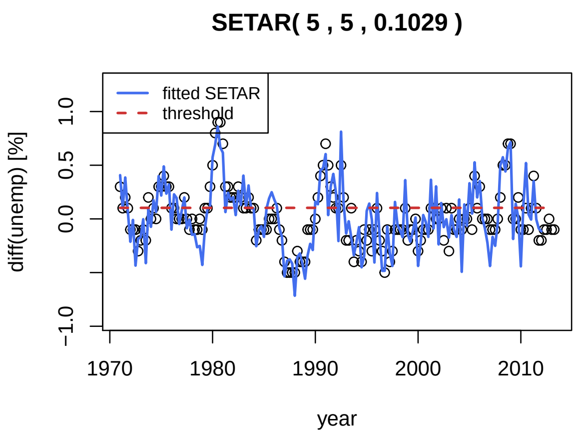

SETAR(4 , 2 , 0.1029) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

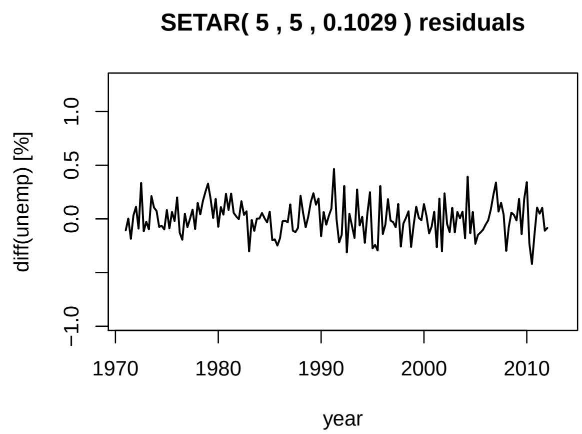

SETAR(4 , 1 , 0.3007)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

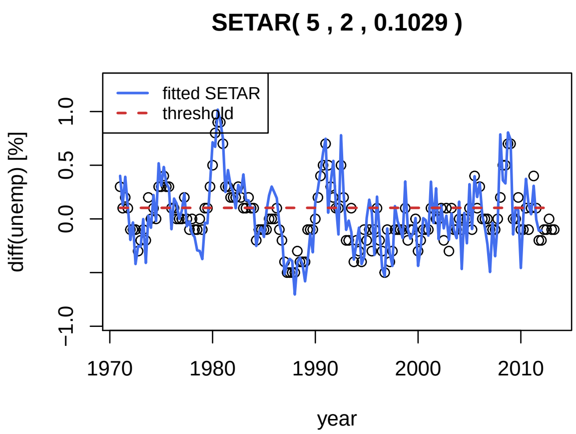

SETAR(4 , 1 , 0.3007) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year

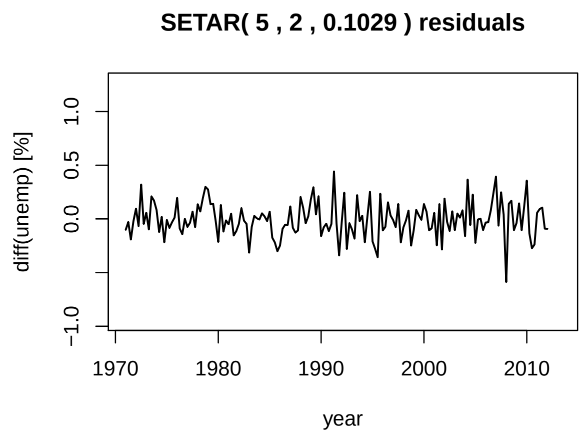

SETAR(3 , 3 , −0.1974)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year


SETAR(3 , 3 , −0.1974) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year


SETAR(5 , 5 , 0.1029)
fitted SETAR
threshold
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year


SETAR(5 , 5 , 0.1029) residuals
diff(unemp) [%]
−1.0
0.0
0.5
1.0
1970
1980
1990
2000
2010
year





3.5 Conclusion

In the above tests, linearity was rejected for the following models: SETAR(3,2,0.103), SETAR(4,2,0.103), SETAR(4,1,0.301), SETAR(3,3,-0.197), SETAR(5,5,0.103), SETAR(5,2,0.103), and thus the validity of their SETAR2 estimation is accepted. It is not yet clear whether another regime should be present in the stochastic process. This will be assessed in the following chapter.

4. 3-Regime SETARs and Diagnostic Tests of SETAR Models

The next step in the analysis using SETAR models is verifying whether 2 regimes suffice. If they do not, we will have to consider the possibility that a third regime needs to be added. In that case, we need to write methods for such model

4.1 Useful Functions

An indicator of a 3-regime SETAR model can take a vector form:

$$I(z_t, c_1, c_2) = \begin{cases} (1,0,0) & \text{if } z_t \leq c_1 \\ (0,1,0) & \text{if } c_1 < z_t \leq c_2 \\ (0,0,1) & \text{if } z_t > c_2 \end{cases}$$

and autoregression combination given by a column vector Y_t can be obtained as a direct product $I(z_t, c_1, c_2) \otimes Y_t$. Using this approach, one can then easily determine the model's covariance matrix

```
## SETAR(1, 1, -0.1, 0.2) cov. matrix:
```

```
##             [,1]      [,2]        [,3]        [,4]       [,5]       [,6]
## [1,] 0.09004739 0.2843602  0.00000000  0.00000000  0.0000000  0.0000000
## [2,] 0.28436019 1.1611374  0.00000000  0.00000000  0.0000000  0.0000000
## [3,] 0.00000000 0.0000000  0.01209941 -0.01635056  0.0000000  0.0000000
## [4,] 0.00000000 0.0000000 -0.01635056  1.37344670  0.0000000  0.0000000
## [5,] 0.00000000 0.0000000  0.00000000  0.00000000  0.1417483 -0.2753650
## [6,] 0.00000000 0.0000000  0.00000000  0.00000000 -0.2753650  0.6653114
```

of parameters $\phi_{i,j}$, $j = 1, 2$, $i = 0, 1, ..., p$.

4.2 The Brock-Dechert-Scheinkman (BDS) Test

Regarded as the most successful tests for nonlinearity due to its universality, the BDS test relies on evaluating a correlation integral $C(q, r)$ as a measure of repeated occurrence of patterns in the time series. It is the estimate of the probability of two arbitrary q-dimensional points in $\mathbb{R}^q$ being no further than $r\hat{\sigma}_\varepsilon$ apart ($0.5 \leq r \leq 1.5$). If the data is generated by an iid process, the correlation integral should approach $C(q, r) \to C(1, r)^q$.

```
## remaining nonlinearity rejected  SETAR(2,2,0.103)  with mean p-val =  0.875
## remaining nonlinearity rejected  SETAR(3,2,0.103)  with mean p-val =  1
## possible remaining nonlinearity for SETAR(3,1,0.301)  with mean p-val =  0
## possible remaining nonlinearity for SETAR(1,1,0)  with mean p-val =  0.125
## possible remaining nonlinearity for SETAR(2,1,0.205)  with mean p-val =  0.125
## remaining nonlinearity rejected  SETAR(4,2,0.103)  with mean p-val =  0.875
## possible remaining nonlinearity for SETAR(4,1,0.301)  with mean p-val =  0.25
## remaining nonlinearity rejected  SETAR(4,4,0.205)  with mean p-val =  0.75
## possible remaining nonlinearity for SETAR(3,3,-0.197)  with mean p-val =  0.25
## remaining nonlinearity rejected  SETAR(5,5,0.103)  with mean p-val =  0.625
## remaining nonlinearity rejected  SETAR(5,4,0.205)  with mean p-val =  0.625
## remaining nonlinearity rejected  SETAR(5,2,0.103)  with mean p-val =  0.75

##
## ==== Remaining nonlinearities detected for: ======

## [1] "SETAR(3,1,0.301)"  "SETAR(1,1,0)"      "SETAR(2,1,0.205)"
## [4] "SETAR(4,1,0.301)"  "SETAR(3,3,-0.197)"
```

4.3 SETAR3 Parameter Estimation

Similarly to section 2.3, we construct an estimation procedure with two distinct threshold parameters c_1 and c_2. Our helper functions are ready from section 4.1. Since the number of regimes m appears as a parameter in the implementation above, we can compose a generalized method for estimation and postprocessing for any number of regimes with output:

```
## SETAR(2, 1, -0.1), m=2

## List of 17
##  $ nReg       : int 2
##  $ p          : num 2
##  $ d          : num 1
##  $ c          : num -0.1
##  $ data       : num [1:170] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 170
##  $ PhiParams  : num [1:6] 0.0711 0.7998 0.1755 0 0.6911 ...
##  $ PhiStErrors: num [1:6] 0.023 0.0946 0.0666 0 0.0444 ...
##  $ skel       : num [1:168, 1] 0.1162 0.1539 0.1005 -0.0534 -0.0264 ...
##  $ residuals  : num [1:168, 1] 0.0838 -0.0539 -0.2005 -0.0466 -0.0736 ...
##  $ resSigmaSq : num 0.0293
##  $ nRegCounts : num [1:2] 50 120
##  $ pOrders    : num [1:2] 2 2
##  $ regSigmaSq : num [1:2] 0.0357 0.0272
##  $ AIC        : num -588
##  $ BIC        : num -574
##  $ name       : chr "SETAR(2,2,-0.1)"

## SETAR(2, 1, -0.1, 0.2), m=3

## List of 17
##  $ nReg       : int 3
##  $ p          : num 2
##  $ d          : num 1
##  $ c          : num [1:2] -0.1 0.2
```

```
##  $ data       : num [1:170] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 170
##  $ PhiParams  : num [1:9] 0.0711 0.7998 0.1755 -0.0171 0.3404 ...
##  $ PhiStErrors: num [1:9] 0.02302 0.09461 0.06663 0.00844 0.09906 ...
##  $ skel       : num [1:168, 1] 0.0908 0.1688 0.0662 -0.0265 -0.0264 ...
##  $ residuals  : num [1:168, 1] 0.1092 -0.0688 -0.1662 -0.0735 -0.0736 ...
##  $ resSigmaSq : num 0.0285
##  $ nRegCounts : num [1:3] 50 84 36
##  $ pOrders    : num [1:3] 2 2 1
##  $ regSigmaSq : num [1:3] 0.0302 0.0178 0.0538
##  $ AIC        : num -589
##  $ BIC        : num -573
##  $ name       : chr "SETAR(2,2,-0.1,0.2)"
```

```
## SETAR(2, 1, -0.1, 0.1, 0.2), m=4
```

```
## List of 17
##  $ nReg       : int 4
##  $ p          : num 2
##  $ d          : num 1
##  $ c          : num [1:3] -0.1 0.1 0.2
##  $ data       : num [1:170] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 170
##  $ PhiParams  : num [1:12] 0.0711 0.7998 0.1755 0 0.4321 ...
##  $ PhiStErrors: num [1:12] 0.023 0.0946 0.0666 0 0.1364 ...
##  $ skel       : num [1:168, 1] 0.1028 0.1688 0.083 -0.0233 -0.0264 ...
##  $ residuals  : num [1:168, 1] 0.0972 -0.0688 -0.183 -0.0767 -0.0736 ...
##  $ resSigmaSq : num 0.0285
##  $ nRegCounts : num [1:4] 50 65 19 36
##  $ pOrders    : num [1:4] 2 2 1 1
##  $ regSigmaSq : num [1:4] 0.0305 0.0172 0.0206 0.0536
##  $ AIC        : num -585
##  $ BIC        : num -567
##  $ name       : chr "SETAR(2,2,-0.1,0.1,0.2)"
```

4.4 SETAR Estimation procedure

Now that we prepared all necessary functions we may proceed to search for 3-regime SETAR's in a suitable search space. This time we will construct our outer loop through delays d which will be reduced to only the delays that are contained within the models with detected remaining nonlinearity:

```
##  unique delays:
```

```
## [1] 1 3
```

Still, even after this alleviation, the search might be computationally demanding. To obtain our results within reasonable time we use `foreach` and `doParallel` packages to compute search through c_1 and c_2 thresholds, and then process the results. Similraly to section 2.3, we iterate through

$$p = 1, 2, 3, 4, 5, 6, 7\ ,\ d = 1, ...,\ p\ ,\quad c_1, c_2 = -0.3, -0.2853, -0.2707...0.3739, 0.3886, 0.4032, 0.4179$$

```
##    p p1 p2 p3 d      c1     c2  n1 n2 n3       AIC       BIC resSigmaSq
## 6  6  2  4  6 1  0.0077 0.4032 102 55 13 -606.0682 -584.2020 0.02371665
## 1  1  1  1  1 1  0.0077 0.4032 102 55 13 -588.3777 -577.9832 0.02925708
## 10 5  4  3  5 3  0.0077 0.4032 102 55 13 -600.4441 -575.9002 0.02451439
## 9  4  4  3  1 3  0.0077 0.2128 102 40 28 -597.8376 -575.2928 0.02609259
## 3  3  2  2  3 1  0.0077 0.3007 102 51 17 -592.0732 -575.0699 0.02731196
## 8  3  2  3  2 3 -0.1974 0.1102  34 95 41 -594.6764 -574.7411 0.02689692
## 4  4  2  2  3 1  0.0077 0.3007 102 51 17 -590.4034 -573.4001 0.02758155
## 12 7  4  5  1 3  0.0077 0.2128 102 40 28 -599.1070 -573.1844 0.02529623
## 11 6  4  4  2 3  0.0077 0.3007 102 51 17 -597.9831 -572.6995 0.02546401
```

```
## 2   2   2   2   1 1   0.0077 0.2128 102 40 28 -587.4819 -571.8759 0.02872771
## 7   7   2   4   6 1   0.0077 0.3007 102 51 17 -594.8218 -571.4552 0.02533871
## 5   5   3   4   5 1   0.0077 0.2128 102 40 28 -585.8058 -558.8683 0.02671882
```

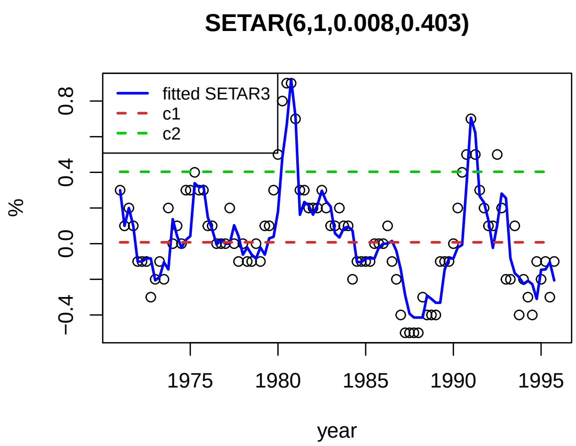
These are the SETAR3 models that can replace the SETAR2's with remaining nonlinearity, ordered by *BIC* with estimated coefficients:

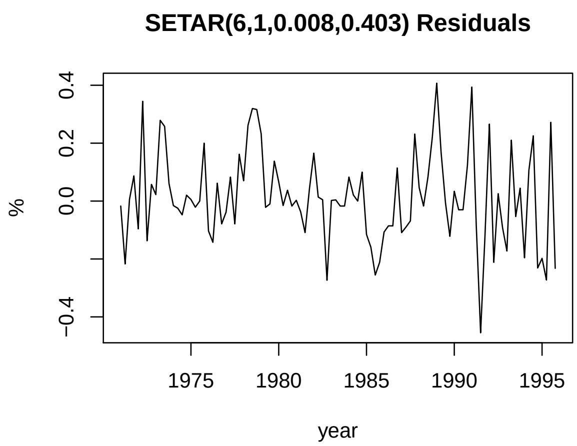
```
## [1] "SETAR(1,1,0.008,0.403)"
```

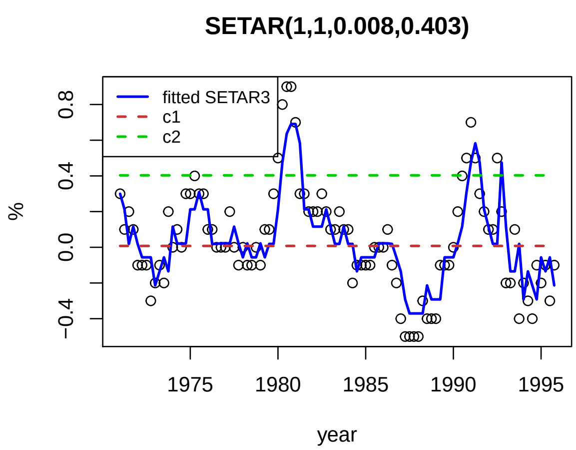
```
##                  [,1]       [,2]        [,3]      [,4]      [,5]      [,6]
## Phi      0.02224840 0.78551645 -0.07770132 0.9679972 0.2052632 0.5394736
## stdError 0.01108097 0.05520027  0.02213275 0.1055138 0.0940983 0.1418585
```

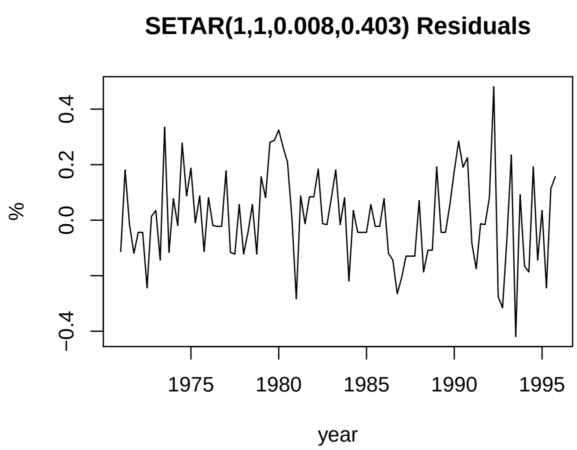
As we might notice, some models differ only in their information criteria and coefficients, since they were estimated from a different maximum order *p*.

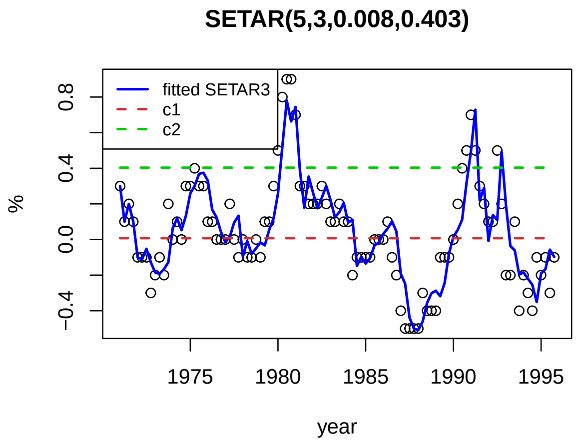
4.5 SETAR3 Visualisation

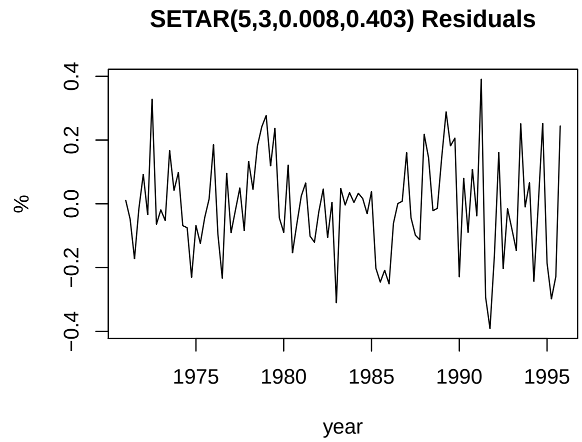












4.6 Conclusion

Due to our limited sampling space of `delays` we obtained 12 possible $SETAR3(p, d, c_1, c_2)$ models, and since remaining SETAR3 nonlinearity was detected in 7 of the original SETAR2 models, we replace them by the top 7 newly found SETAR3's:

```
##                      model      BIC
## 1         SETAR(2,2,0.103) -602.841
## 2         SETAR(3,2,0.103) -588.235
## 3   SETAR(6,1,0.008,0.403) -584.202
## 4   SETAR(1,1,0.008,0.403) -577.983
## 5         SETAR(4,2,0.103) -577.642
## 6   SETAR(5,3,0.008,0.403)   -575.9
## 7         SETAR(4,4,0.205) -575.688
## 8   SETAR(4,3,0.008,0.213) -575.293
## 9   SETAR(3,1,0.008,0.301)  -575.07
## 10        SETAR(5,5,0.103) -573.138
## 11        SETAR(5,4,0.205) -571.548
## 12        SETAR(5,2,0.103) -571.394
```

To illustrate the inner dynamics of at least two highest ranked SETAR3 models, we use their respective coefficients:

$$X_t = \begin{cases} (0.62 \pm 0.07)X_{t-1} + (0.21 \pm 0.05)X_{t-2} + \varepsilon_t \\ (-0.08 \pm 0.02) + (0.9 \pm 0.12)X_{t-1} + (0.23 \pm 0.07)X_{t-2} + (-0.39 \pm 0.1)X_{t-5} + (0.38 \pm 0.07)X_{t-6} + \varepsilon_t \\ (0.6 \pm 0.22)X_{t-1} + (0.58 \pm 0.27)X_{t-2} + (0.88 \pm 0.34)X_{t-3} + (-1.7 \pm 0.36)X_{t-4} + (-0.96 \pm 0.4)X_{t-5} + (0.93 \pm 0.27)X_{t-6} + \varepsilon_t \end{cases}$$

$\hat{\sigma}_{\varepsilon,1} = 0.0171$, $\hat{\sigma}_{\varepsilon,2} = 0.0312$

$$X_t = \begin{cases} (0.02 \pm 0.02) + (0.79 \pm 0.79)X_{t-1} + \varepsilon_t & \text{if } X_{t-1} \leq 0.0077 \\ (-0.08 \pm -0.08) + (0.97 \pm 0.97)X_{t-1} + \varepsilon_t & \text{if } 0.0077 < X_{t-1} \leq 0.4032 \qquad \hat{\sigma}_\varepsilon = 0.0293 \\ (0.21 \pm 0.21) + (0.54 \pm 0.54)X_{t-1} + \varepsilon_t & \text{if } X_{t-1} > 0.4032 \end{cases}$$

$\hat{\sigma}_{\varepsilon,1} = 0.0228$, $\hat{\sigma}_{\varepsilon,2} = 0.0333$

5. Predictions via SETAR Models and Their Evaluation

5.1 Helper functions

Since we have not yet defined a skeleton function, i.e. one that would continue plugging in the time series values even after the end of testing data. For that purpose we implement a step-forward function for an m-regime SETAR equivalent to an m-regime skeleton:

$$F(z_t, \theta) = \begin{cases} \phi_{0,1} + \phi_{1,1} X_{t-1} + ... + \phi_{p_1,1} & \text{if } \; z_t \leq c_1 \\ \phi_{0,2} + \phi_{1,2} X_{t-1} + ... + \phi_{p_2,2} & \text{if } \; c_1 < z_t \leq c_2 \\ \quad \vdots & \quad \vdots \\ \phi_{0,m} + \phi_{1,m} X_{t-1} + ... + \phi_{p_m,m} & \text{if } \; z_t > c_{m-1} \end{cases}$$

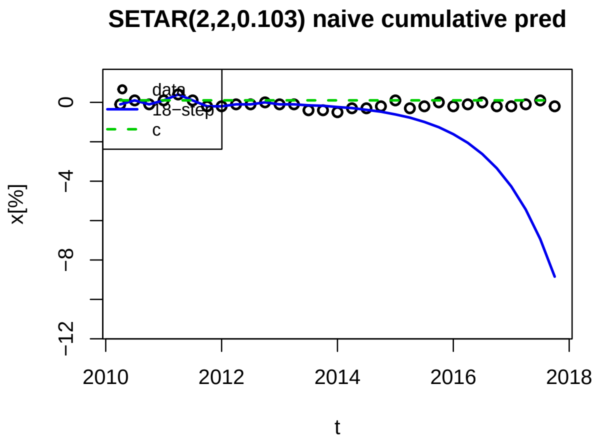
We can test it on a particular SETAR model:

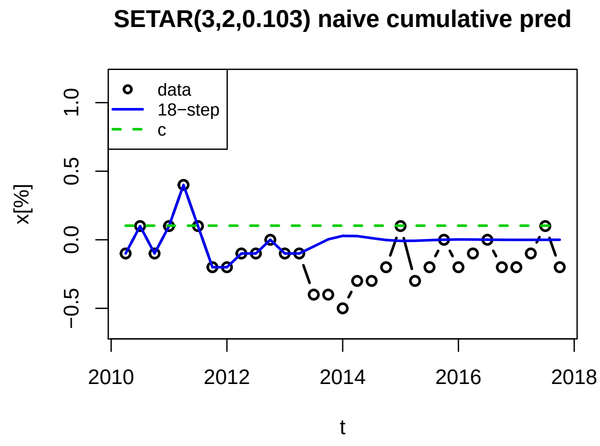
```
## SETAR(6,1,0.008,0.403) 1-step:

## x_out:

## [1] -0.10  0.32

## data:

## [1] -0.1 -0.4
```

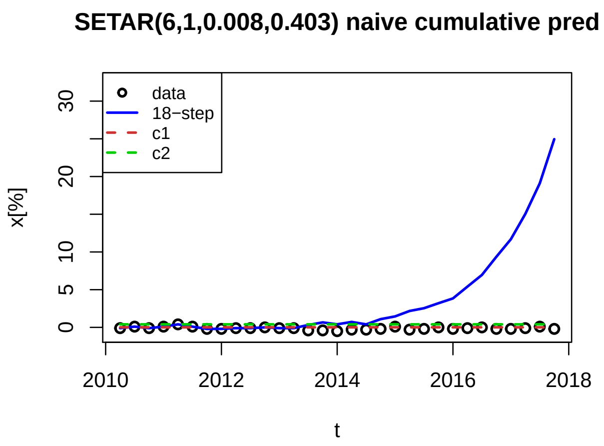
This is a single-step prediction of the data using the first model. When we set `n_ahead > 1`, the step function builds up upon previous predicted values and the skeleton converges to a model's particular equilibrium:

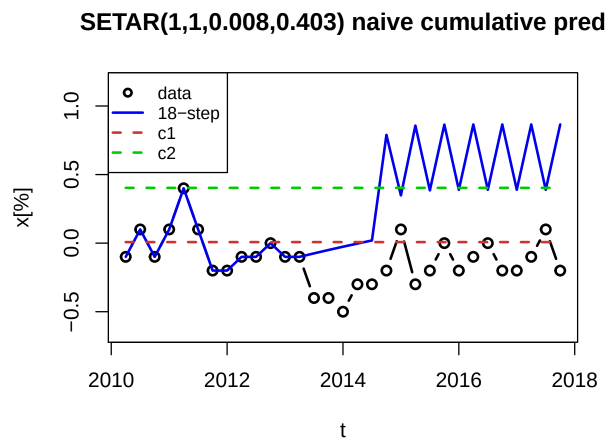
```
## SETAR(3,2,0.103) predict

##           x_out data
## 1  -0.100000000 -0.1
## 2  -0.048566862 -0.4
## 3   0.003344905 -0.4
## 4   0.028518439 -0.5
## 5   0.026915124 -0.3
## 6   0.012161582 -0.3
## 7  -0.001767309 -0.2
## 8  -0.008100326  0.1
## 9  -0.007203219 -0.3
## 10 -0.003021448 -0.2

##
##  equilibria:

## [1] 0.0000 0.3459
```

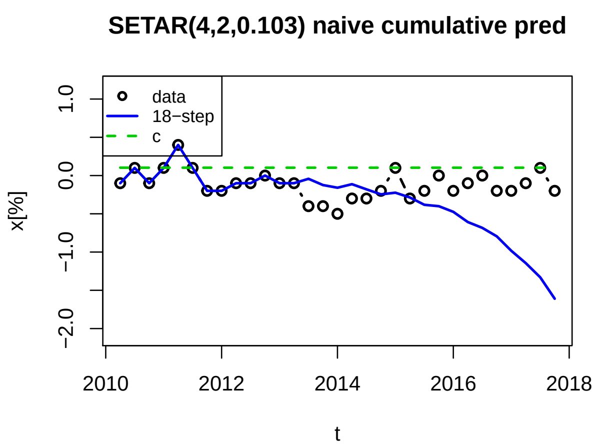
If the model is explosive, the predictions will diverge:

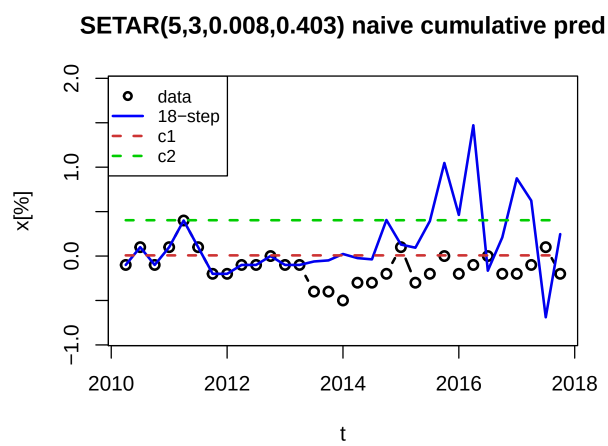










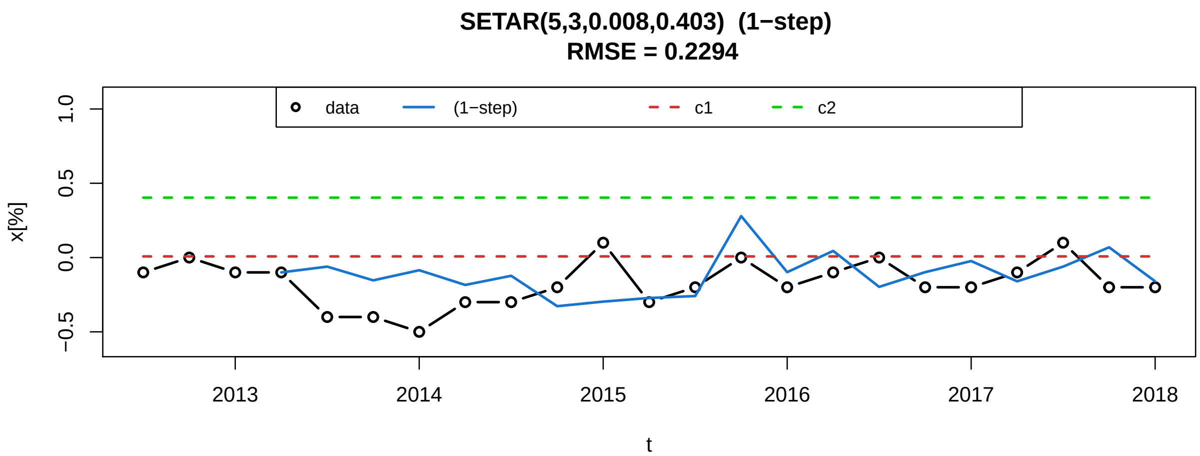


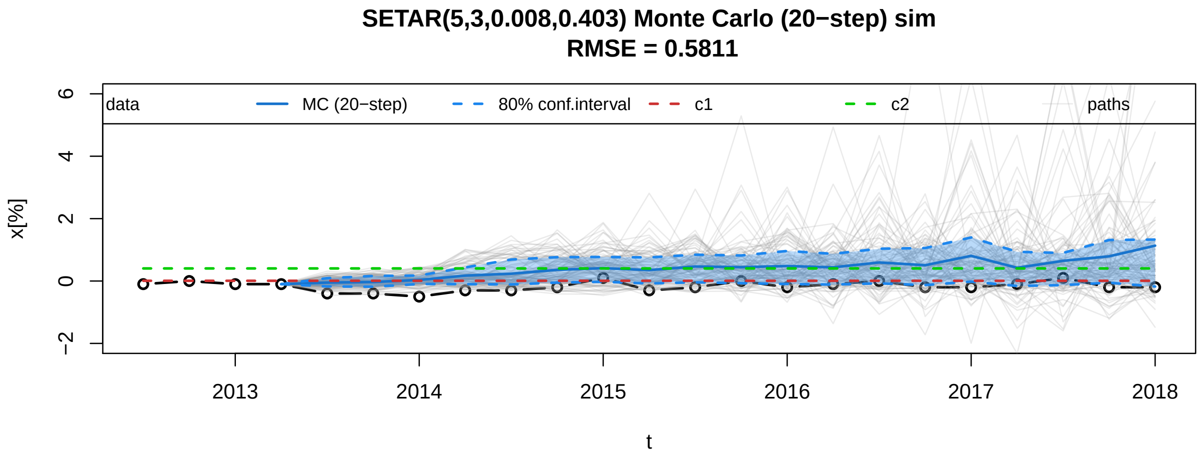
The examples above tested 18 steps of a naiive approach to prediction, by assuming the process evolves via its skeleton. More convenient approaches are `Monte Carlo` (`"MC"`) and `Bootstrap`. Both rely on adding noise to series predictions. Monte Carlo approach simulates normally distributed noise $\epsilon \sim N(0, \hat{\sigma}_\varepsilon)$ from residual standard error $\hat{\sigma}_\varepsilon$, and Bootstrap, on the other hand, does not assume the normality of model residuals and instead randomly samples the residuals themselves.

Now we test our prediction method on the data:

```
## naive (10-step):
##  [1] -0.100 -0.100 -0.061 -0.049  0.023 -0.023 -0.037  0.405  0.130  0.094
## [11]  0.394
## Monte Carlo (10-step):
##  [1] -0.100 -0.100 -0.075 -0.063  0.023  0.084  0.151  0.336  0.334  0.329
## [11]  0.334
## Bootstrap (10-step):
##  [1] -0.100 -0.100 -0.068 -0.092  0.009  0.074  0.153  0.274  0.283  0.351
## [11]  0.417
## data:
##  [1] -0.1 -0.4 -0.4 -0.5 -0.3 -0.3 -0.2  0.1 -0.3 -0.2  0.0
## naive (1-step):
##  [1] -0.100 -0.100 -0.061 -0.154 -0.085 -0.184 -0.123 -0.327 -0.297 -0.272
## [11] -0.259
## Monte Carlo (1-step):
##  [1] -0.100 -0.100 -0.061 -0.154 -0.085 -0.184 -0.123 -0.327 -0.297 -0.272
## [11] -0.259
## Bootstrap (1-step):
##  [1] -0.100 -0.100 -0.061 -0.154 -0.085 -0.184 -0.123 -0.327 -0.297 -0.272
## [11] -0.259
## data:
##  [1] -0.1 -0.4 -0.4 -0.5 -0.3 -0.3 -0.2  0.1 -0.3 -0.2  0.0
```

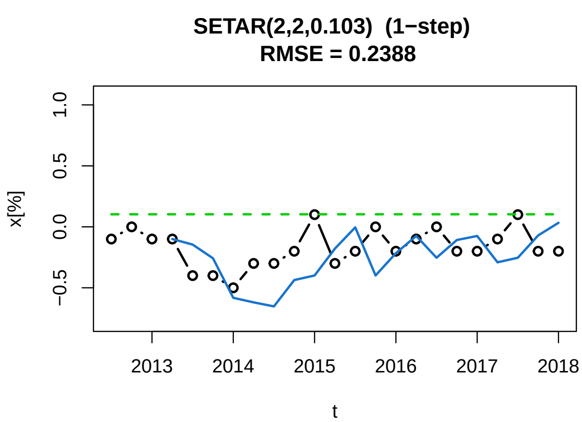
To test other prediction results, such as confidence intervals and simulation paths, we implement a plot procedure:

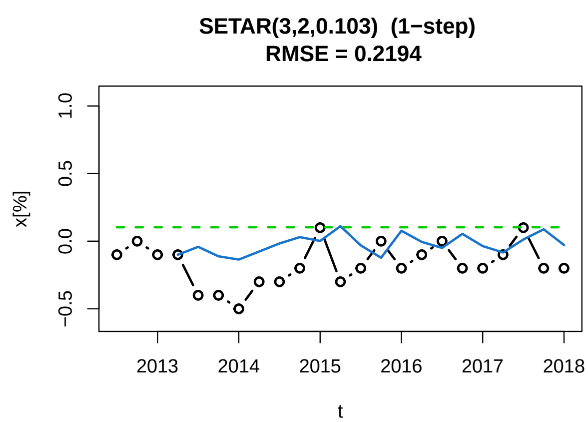


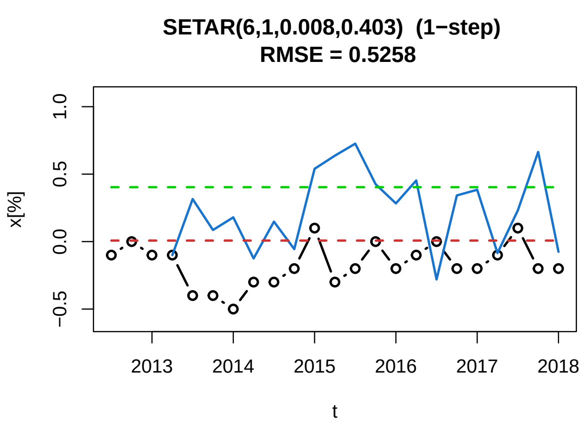


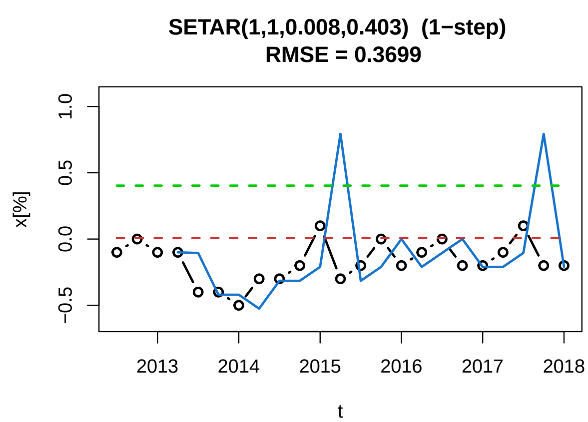
Now we possess all necessary tools to proceed evaluating our SETAR models according to their predictive abilities.

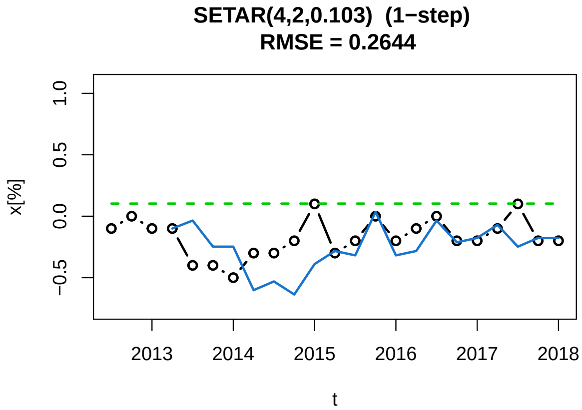
5.2 Single-Step Predictions of SETAR Models

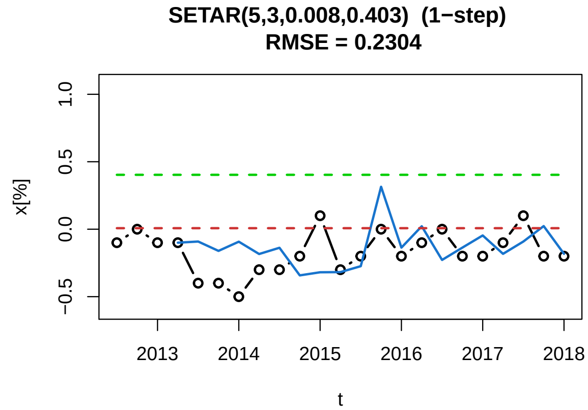


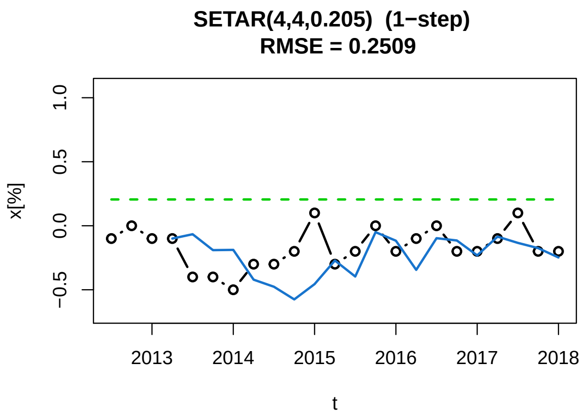


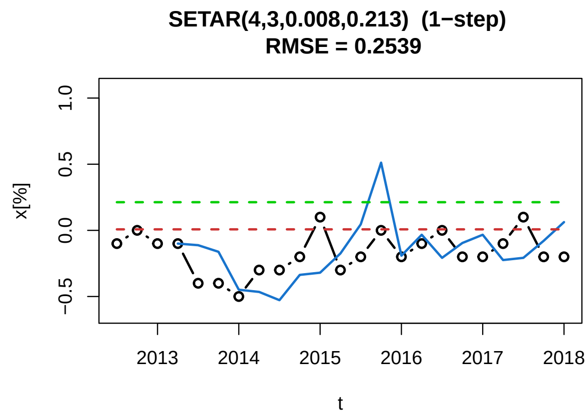


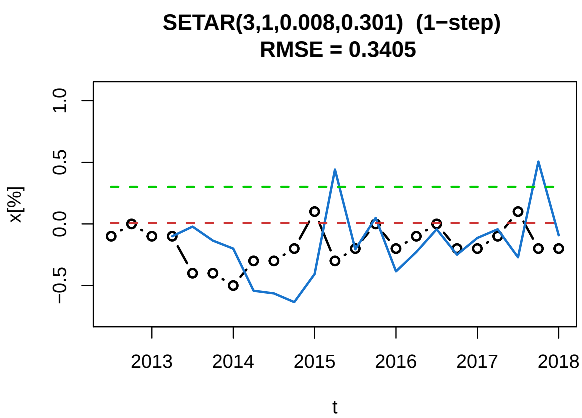


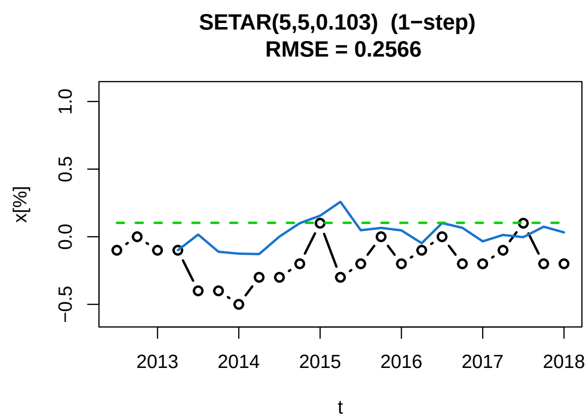

SETAR(4,2,0.103) (1−step)
RMSE = 0.2644
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t

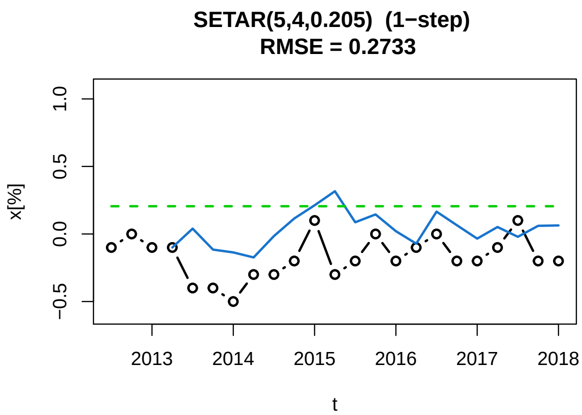

SETAR(5,3,0.008,0.403) (1−step)
RMSE = 0.2304
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t

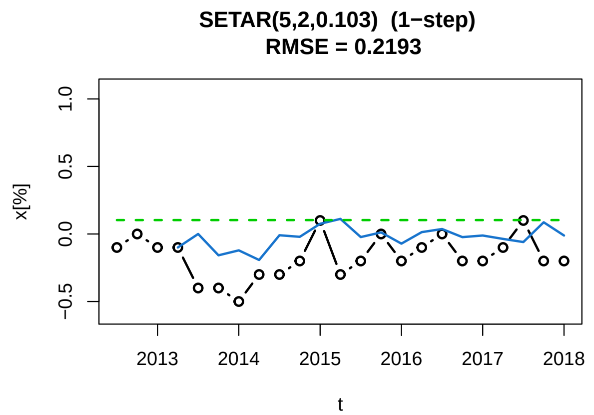

SETAR(4,4,0.205) (1−step)
RMSE = 0.2509
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t


SETAR(4,3,0.008,0.213) (1−step)
RMSE = 0.2539
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t


SETAR(3,1,0.008,0.301) (1−step)
RMSE = 0.3405
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t

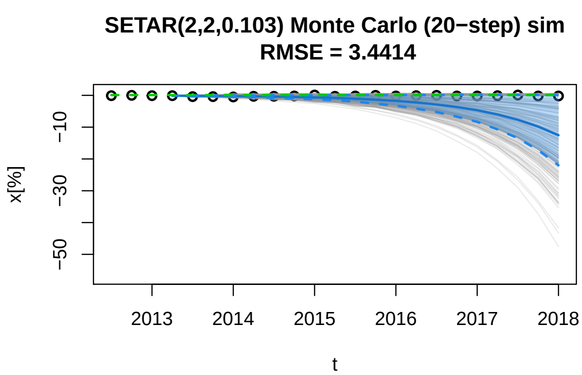

SETAR(5,5,0.103) (1−step)
RMSE = 0.2566
x[%]
1.0
0.5
0.0
−0.5
2013
2014
2015
2016
2017
2018
t

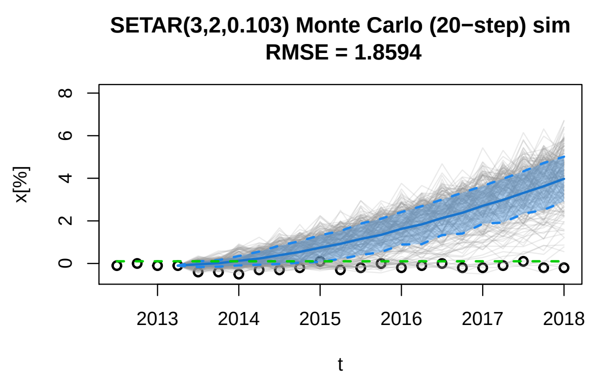


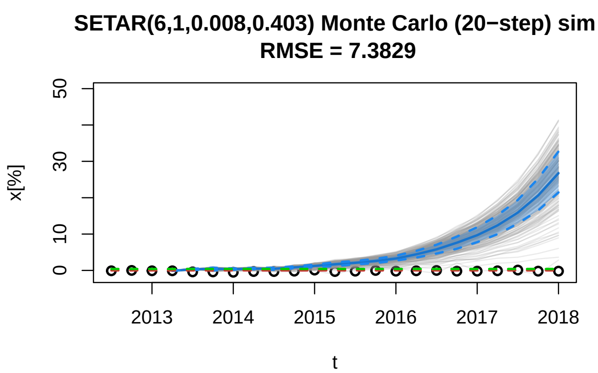


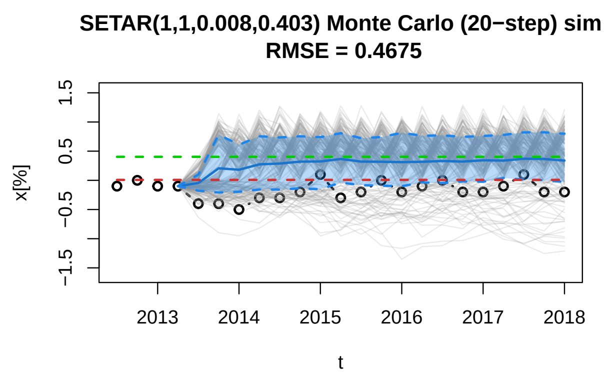
5.3 Multi-Step Predictions of SETAR Models

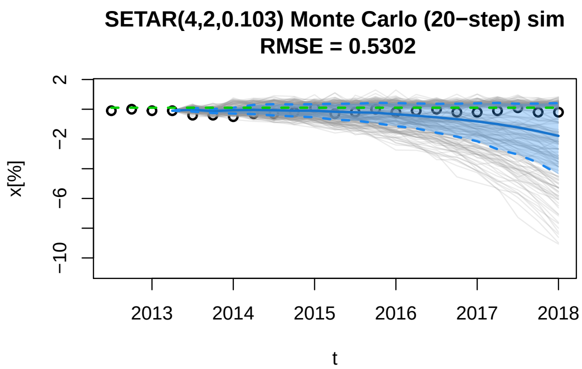
Monte Carlo:

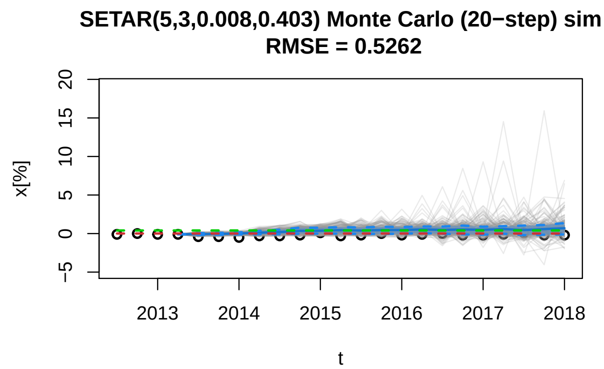


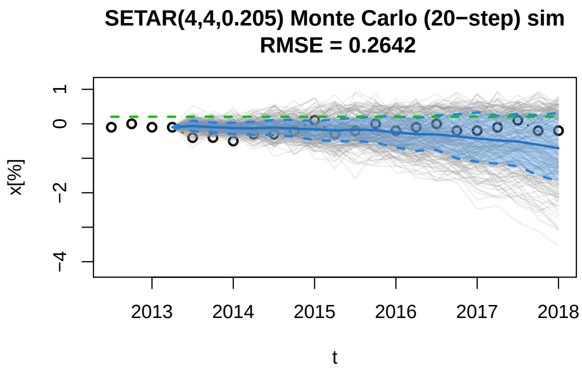


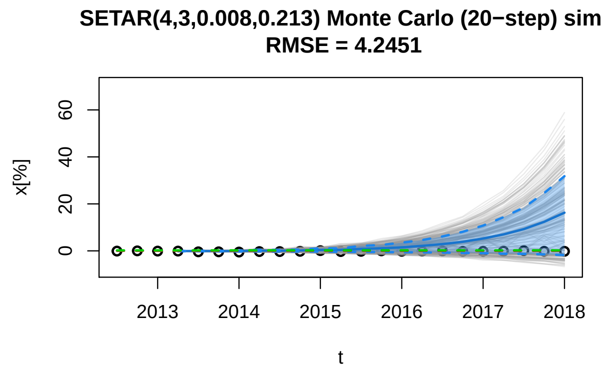


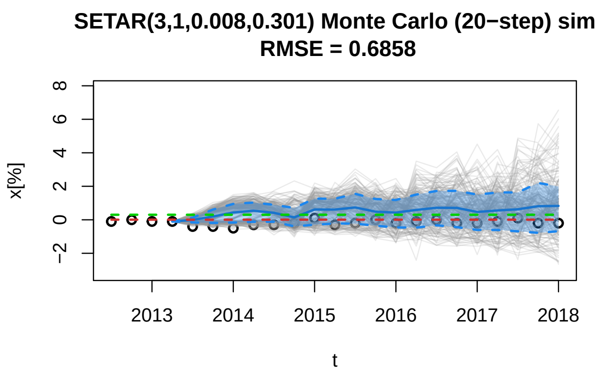


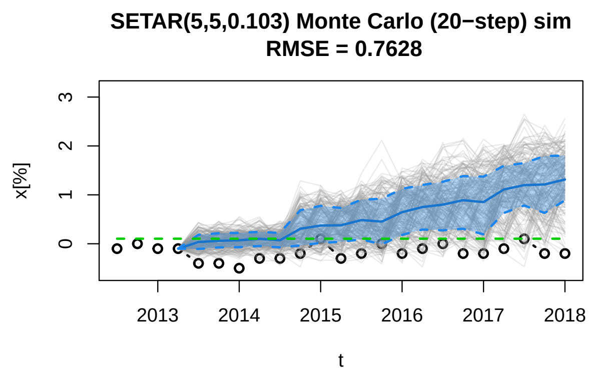

SETAR(4,2,0.103) Monte Carlo (20−step) sim
RMSE = 0.5302
x[%]
t

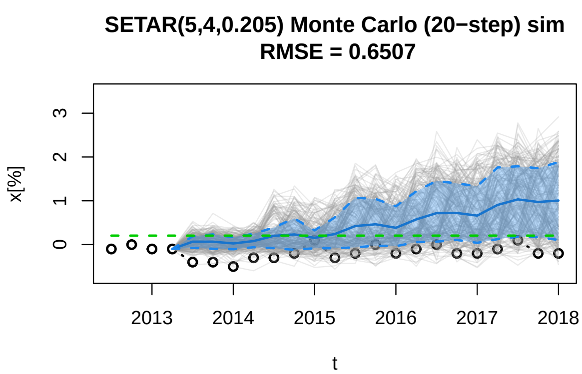

SETAR(5,3,0.008,0.403) Monte Carlo (20−step) sim
RMSE = 0.5262
x[%]
t

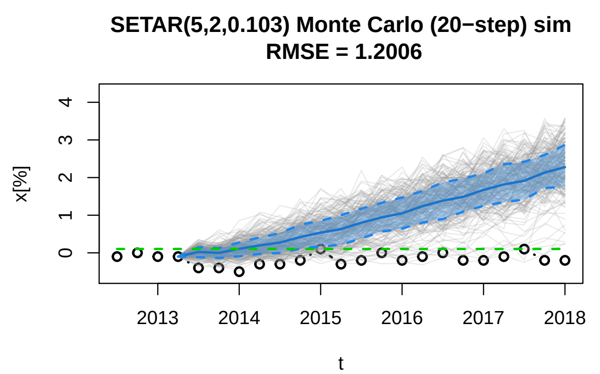

SETAR(4,4,0.205) Monte Carlo (20−step) sim
RMSE = 0.2642
x[%]
t


SETAR(4,3,0.008,0.213) Monte Carlo (20−step) sim
RMSE = 4.2451
x[%]
t


SETAR(3,1,0.008,0.301) Monte Carlo (20−step) sim
RMSE = 0.6858
x[%]
t

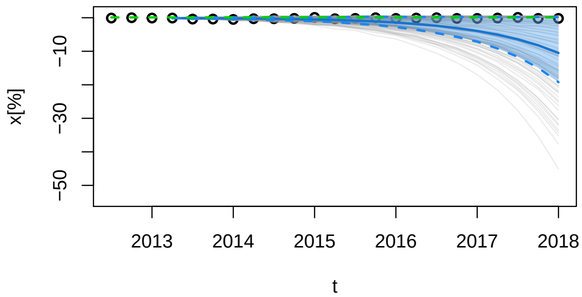

SETAR(5,5,0.103) Monte Carlo (20−step) sim
RMSE = 0.7628
x[%]
t


SETAR(5,4,0.205) Monte Carlo (20−step) sim
RMSE = 0.6507
x[%]
t

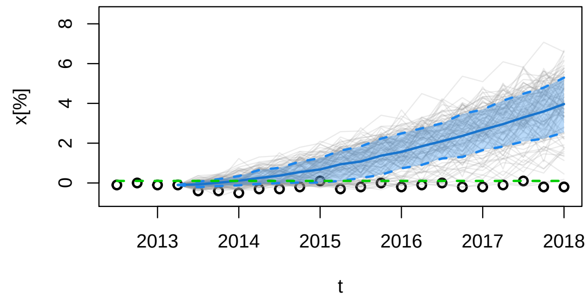

SETAR(5,2,0.103) Monte Carlo (20−step) sim
RMSE = 1.2006
x[%]
t

Bootstrap:

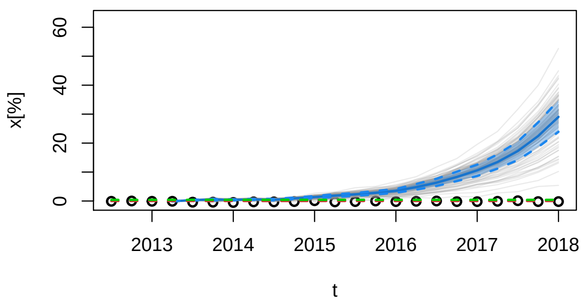
SETAR(2,2,0.103) Bootstrap (20−step) sim
RMSE = 2.8746



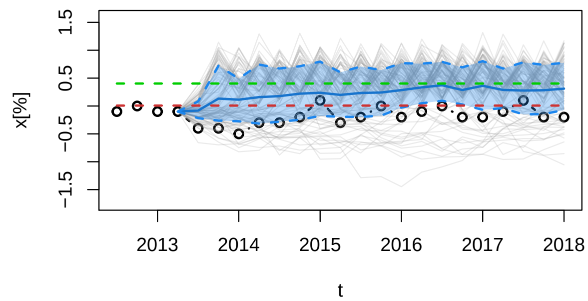
SETAR(3,2,0.103) Bootstrap (20−step) sim
RMSE = 1.8423



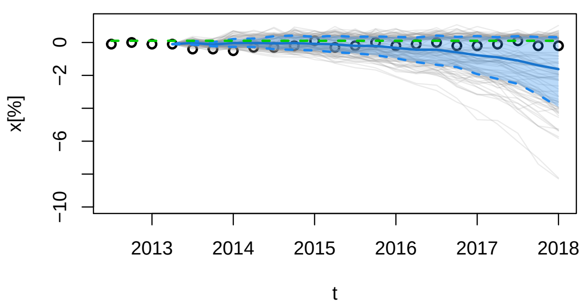
SETAR(6,1,0.008,0.403) Bootstrap (20−step) sim
RMSE = 8.021



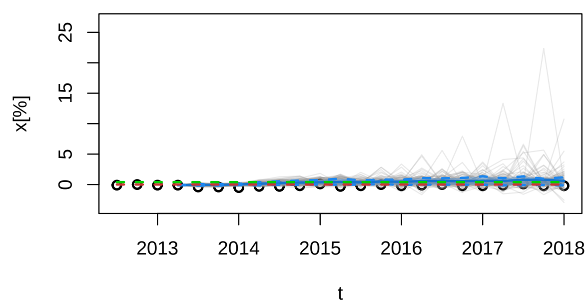
SETAR(1,1,0.008,0.403) Bootstrap (20−step) sim
RMSE = 0.41



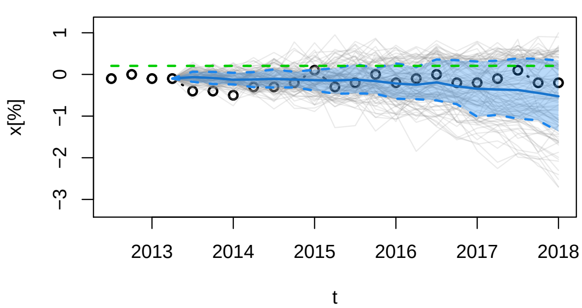
SETAR(4,2,0.103) Bootstrap (20−step) sim
RMSE = 0.4893



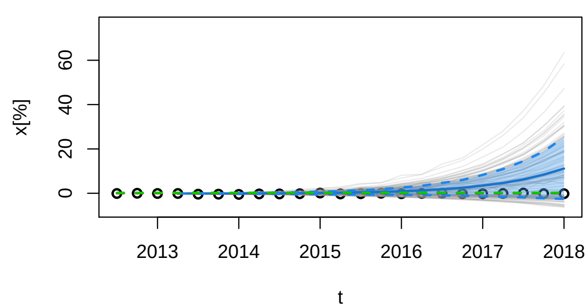
SETAR(5,3,0.008,0.403) Bootstrap (20−step) sim
RMSE = 0.5871

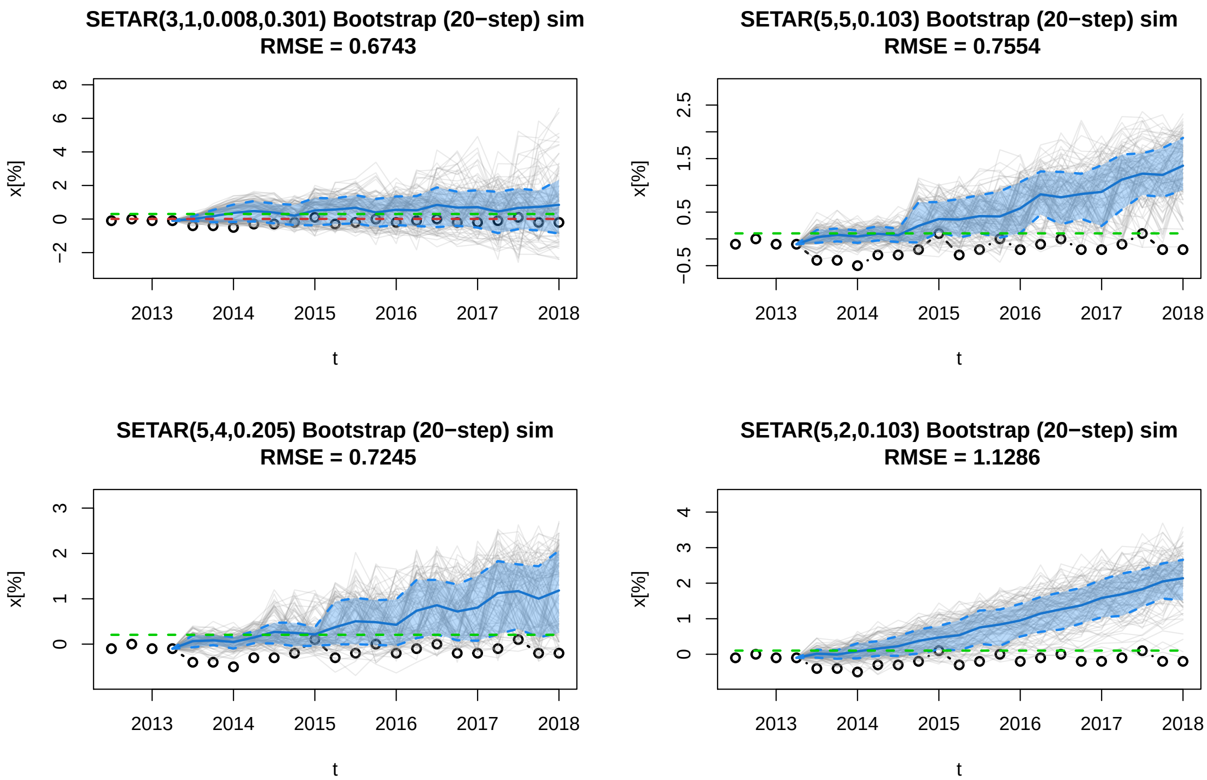


SETAR(4,4,0.205) Bootstrap (20−step) sim
RMSE = 0.222



SETAR(4,3,0.008,0.213) Bootstrap (20−step) sim
RMSE = 2.872





5.4 Evaluating Models

From what we observe, some models exhibit explosive properties, and have equilibria significantly distant from the evaluated data set, which, of course, impacts the resulting prediction error. Now we evaluate models according to their predictive properties, and filter out those that produce distant and/or explosive predictions.

```
##                      model      BIC sigmaSq MSE(1-step)    MSE(MC)    MSE(boot)
## 1         SETAR(2,2,0.103) -602.841  0.0262      0.0605     7.8224 ~     8.5872 ~
## 2         SETAR(3,2,0.103) -588.235  0.0267      0.0455     3.5444 ~     3.7417 ~
## 3   SETAR(6,1,0.008,0.403) -584.202  0.0237      0.2804 ~ 54.7335 ~    64.3895 ~
## 4   SETAR(1,1,0.008,0.403) -577.983  0.0293      0.1351 ~  0.2083       0.2058
## 5         SETAR(4,2,0.103) -577.642  0.0268      0.0735     0.3375       0.2148
## 6   SETAR(5,3,0.008,0.403)   -575.9  0.0245      0.0526     0.3314       0.2917
## 7         SETAR(4,4,0.205) -575.688  0.0270      0.0643     0.0833       0.0446
## 8   SETAR(4,3,0.008,0.213) -575.293  0.0261      0.0702    20.9411 ~     7.2916 ~
## 9   SETAR(3,1,0.008,0.301)  -575.07  0.0273      0.1140 ~   0.4626       0.5084
## 10        SETAR(5,5,0.103) -573.138  0.0254      0.0704     0.5804 ~     0.6291 ~
## 11        SETAR(5,4,0.205) -571.548  0.0262      0.0729     0.4194       0.5350
## 12        SETAR(5,2,0.103) -571.394  0.0259      0.0504     1.3318 ~     1.2221 ~
```

We notice that models that quickly diverge have a significantly higher MSE than their $\hat{\sigma}^2_\varepsilon$ (`RSS / (n - k)`). One could then formulate a condition that model is "too divergent" if its training data MSE (`RSS / (n - k)`) is significantly lower than prediction MSE. The significance factor could be taken, for example, as 4, i.e.: if the prediction MSE exceeds 4-multiple of training data `RSS / (n - k)`, the prediction diverges. In the above table, we mark divergent predictions with `"~"`. When examining multi-step predictions we take a factor of 8 of 1-step MSE.

5.5 Conclusion and SETAR Evaluation

After considering all possible configurations of SETAR models, testing them for remaining nonlinearity, and evaluating their predictive properties, we conclude the following:

Some models placed within the top 3 best in their fit onto the training data (according to their BIC), contained SETAR3-type remaining nonlinearity, and thus had their thresholds estimated, turned out to have radically different equilibria than their 2-regime predecessors. Some were even explosive when we examined their predictive properties. Hence their training data fit quality cannot justify their mismatch between data and their properties as stochastic dynamical systems.

With regards to their predictive properties, we pick the best models according to their given MSE depending on the prediction method (`"naive"`, `"mc"`, `"boot"`):

```
## Models sorted by 1-step naive MSE:

##                       model      BIC sigmaSq MSE(1-step)
## 2         SETAR(3,2,0.103) -588.235  0.0267      0.0455
## 12        SETAR(5,2,0.103) -571.394  0.0259      0.0504
## 6  SETAR(5,3,0.008,0.403)   -575.9  0.0245      0.0526
## 1         SETAR(2,2,0.103) -602.841  0.0262      0.0605
## 7         SETAR(4,4,0.205) -575.688  0.0270      0.0643
## 8  SETAR(4,3,0.008,0.213) -575.293  0.0261      0.0702
## 10        SETAR(5,5,0.103) -573.138  0.0254      0.0704
## 11        SETAR(5,4,0.205) -571.548  0.0262      0.0729
## 5         SETAR(4,2,0.103) -577.642  0.0268      0.0735

## Models sorted by Monte Carlo MSE:

##                       model      BIC sigmaSq MSE_mc
## 7         SETAR(4,4,0.205) -575.688  0.0270 0.0833
## 4  SETAR(1,1,0.008,0.403) -577.983  0.0293 0.2083
## 6  SETAR(5,3,0.008,0.403)   -575.9  0.0245 0.3314
## 5         SETAR(4,2,0.103) -577.642  0.0268 0.3375
## 11        SETAR(5,4,0.205) -571.548  0.0262 0.4194
## 9  SETAR(3,1,0.008,0.301)  -575.07  0.0273 0.4626

## Models sorted by Bootstrap MSE:

##                       model      BIC sigmaSq MSE_boot
## 7         SETAR(4,4,0.205) -575.688  0.0270   0.0446
## 4  SETAR(1,1,0.008,0.403) -577.983  0.0293   0.2058
## 6  SETAR(5,3,0.008,0.403)   -575.9  0.0245   0.2917
## 5         SETAR(4,2,0.103) -577.642  0.0268   0.2148
## 11        SETAR(5,4,0.205) -571.548  0.0262   0.5350
## 9  SETAR(3,1,0.008,0.301)  -575.07  0.0273   0.5084
```

and as we observe, the order of the first 4 models remains the same regardless of the prediction method.

6. STAR Model Estimation

While SETAR-type models are governed by a discontinuous transition function (`Indicator`), we have not yet tried a continuous alternative of transitioning between regimes. STAR (Smooth Threshold AutoRegressive) models utilize the possibility of a smooth regime transition. As will the context of the following sections suggest, we can choose a particular transition function between exponential (ESTAR) and logistic (LSTAR) and a scalar parameter γ which describes the "smoothness" of a regime transition. For $\gamma \to \infty$ we obtain a discontinuous step-like transition function, as in SETAR's indicator.

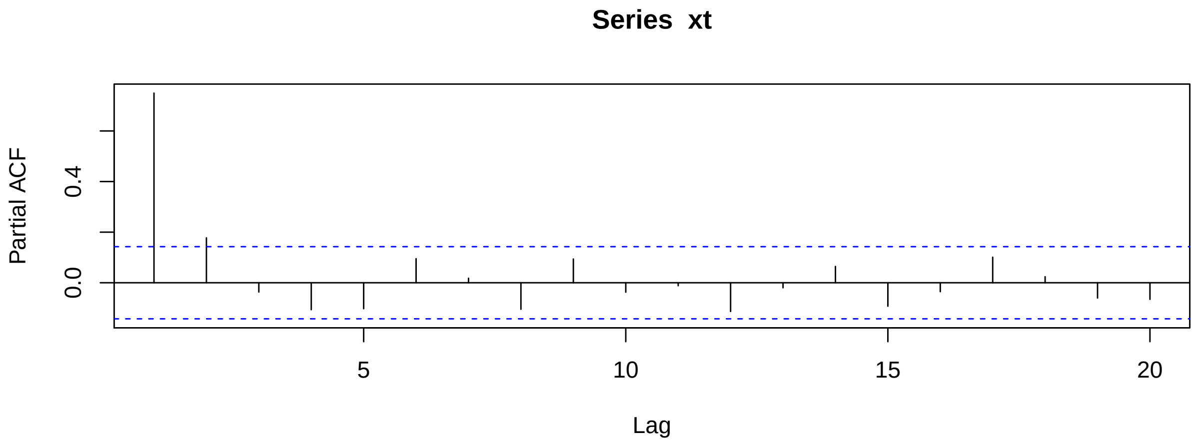
6.1 Tests For STAR-type nonlinearity

To test for STAR-type nonlinearity we use a specialized form of an LM test.

To decide whether an exponential or a logistic transition function provides a better fit, we test for the last parameter vector of a regression test from `LMtest_LINvsSTAR` is zero. By accepting the hypothesis, we conclude that exponential transition fits our data better.

The statistic of an LM test is assumed to be of $\chi^2(p)$ distribution. This means that the p-value of the test has to be compared with a significance factor normalized by the parameter vector's dimension, i.e.: p-val $< \alpha/p$ (Bonferoni correction). Hence the `alpha_corrected` value.

Naturally, we need to test for STAR-type non-linearity for an array of discrete parameters p and d. The delay parameter d is, of course, bounded by the lag parameter p, the upper bound of which can be assumed from the partial autocorrelation of the series:



Since partial autocorrelation dies out after 2 lag steps, we should limit the sample space by $p_{max} = 2$, however to show a reasonably large sample, we follow an assumption that $p_{max} = 12$:

```
## LSTAR non-linearity test results (linearity rejected):

##    p d      p_value        crit
## 3   2 2 0.0063931099 0.008333333
## 8   4 2 0.0015530134 0.004166667
## 12  5 2 0.0025031538 0.003333333
## 22  7 1 0.0018492820 0.002380952
## 23  7 2 0.0004380275 0.002380952
## 30  8 2 0.0010607821 0.002083333
## 38  9 2 0.0010282912 0.001851852

## ESTAR non-linearity test results (linearity rejected):

##   p d     p_value        crit
## 3 2 2 0.003337111 0.008333333
```

Tests show a clear LSTAR preference, with an overlap with ESTAR nonlinearity for $p = 2$.

```
## $LSTARs
##    p d
## 3  2 2
## 8  4 2
## 12 5 2
## 22 7 1
## 23 7 2
## 30 8 2
## 38 9 2
##
## $ESTARs
##   p d
```

```
## 3 2 2
```

The parameter intersection between the above samples should also be examined for preference, using our ESTAR vs LSTAR test:

```
##      p d     LM crit_val p_value alpha_corrected.p
## [1,] 2 2 2.3587   5.9915  0.3075             0.025

##
##  ESTAR preference:

## [1] NA
```

Which suggests that we ought to consider only LSTAR models.

6.2 Estimating STAR Model Parameters

First, we implement and test all necessary functions (for m regimes).

Regime Transition

We distinguish logistic G_L and exponential G_E transition functions:

$$G_L(z_t, c, \gamma) = \frac{1}{1 + \mathrm{e}^{-\gamma(z_t - c)}} \qquad G_E(z_t, c, \gamma) = 1 - \mathrm{e}^{-\gamma(z_t - c)^2}$$

parametrized by a smoothing parameter $\gamma \in \mathbb{R}^{m-1}$ (for $m > 1$ all products are direct vector products).

Single Regime Basis

$$Y_t = (1, X_{t-1}, ..., X_{t-p})^\top$$

m-Regime Basis

$$X_t^{(m)} = \Big((1 - G(z_t, c_1, \gamma_1))Y_t \ , \ \ (G(z_t, c_1, \gamma_1) - G(z_t, c_2, \gamma_2))Y_t \ , \ \dots \ , G(z_t, c_{m-1}, \gamma_{m-1})Y_t\Big)^\top$$

Skeleton

$$F(z_t, \theta) = (\phi^{(+)} - \phi^{(-)})X_t^{(m)} \ , \qquad \phi^{(+)} = (\phi_1, ..., \phi_m)^\top, \phi^{(-)} = (\mathbf{0}_{(p+1)\times 1}, ..., \phi_{m-1})^\top$$

where $\theta = (\phi_1, ..., \phi_m, c, \gamma)^\top$, and $\dim\theta = m(p+1) + 2(m-1)$.

Information Criteria

$$AIC_{STAR} = -2\ l(\theta) + 2\dim\theta \ , \quad BIC_{STAR} = -2\ l(\theta) + \dim\theta\log n$$

where $l(\theta) = -\frac{n}{2}\log\frac{2\pi}{\sigma_\varepsilon^2} - \frac{1}{2}(n-p)$ is the log-likelihood function.

Parameter Estimation

The initial estimation of a STAR model candidate follows the same pattern as estimation methods in sections 2.2 and 4.3:

Estimating Standard Errors of STAR Parameters

Notice, that we can no longer find a consitent estimate of parameter variances from the AR covariance matrix alone. Individual regimes are interdependent thanks to smooth transition functions. We will need to find an estimate of a covariance matrix $\mathbf{\Sigma}_{\hat{\theta}}$ of all parameters $\theta = (\phi_1, \phi_2, ..., \phi_m, \gamma_1, ..., \gamma_{m-1}, c_1, ..., c_{m-1})^\top$ (where m is the number of regimes). Under our given conditions, the least square estimate $\hat{\theta}$ of θ is asymptotically normally distributed, i.e.: $\sqrt{n}(\hat{\theta} - \theta) \sim N(\mathbf{0}, n\mathbf{\Sigma}_{\hat{\theta}})$ for $n \to \infty$. Where a consitent estimate of the covariance matrix is given by $\hat{\mathbf{\Sigma}}_{\hat{\theta}} = \frac{1}{n}\hat{\mathbf{H}}_n^{-1}\hat{\mathbf{M}}_n\hat{\mathbf{H}}_n^{-1}$, where $\hat{\mathbf{H}}_n = \frac{1}{n-p}\sum_{t=p+1}^{n} \nabla^2 r_t(\hat{\theta})$ is the mean Hessian, $\hat{\mathbf{M}}_n = \frac{1}{n-p}\sum_{t=p+1}^{n} \nabla r_t(\hat{\theta})\nabla r_t(\hat{\theta})^\top$ the mean information matrix, and $r_t(\theta) = \big(x_t - F(x_{t-d}, \theta)\big)^2$ (F is the model skeleton).

Given the dimensions of our model (m - number of regimes, p - AR parameter lag), we obtain symmetrical square matrices of dimension $\dim\theta \times \dim\theta$, where $\dim\theta = m(p+1) + 2(m-1)$. Both $\big(r_t(\hat{\theta})\nabla r_t(\hat{\theta})^\top\big)_{\dim\theta\times\dim\theta}$ and $\big(\nabla^2 r_t(\hat{\theta})\big)_{\dim\theta\times\dim\theta}$ have to be computed for each $t = p+1, ..., n$. Differentiating r_t with respect to its parameters we get $\frac{\partial r_t}{\partial\theta_i} = 2(F_t - x_t)\frac{\partial F_t}{\partial\theta_i} = (\nabla r_t)_i$ and $\frac{\partial^2 r_t}{\partial\theta_i\partial\theta_j} = 2\left(\frac{\partial F_t}{\partial\theta_i}\frac{\partial F_t}{\partial\theta_j} + (F_t - x_t)\frac{\partial^2 F_t}{\partial\theta_i\partial\theta_j}\right) = (\nabla^2 r_t)_{i,j}$. The respective derivatives of skeleton F_t are elements of a gradient vector $(\nabla F_t)_{\dim\theta}$ and Hessian $(\nabla^2 F_t)_{\dim\theta\times\dim\theta}$. It suffices to compute the elements of ∇F_t and $\nabla^2 F_t$ to estimate the mean residual square (`rs`) Hessian and information matrix, and use the above formulas.

The skeleton gradient and Hessian can be divided into sub-matrices according to the differentiated variable:

$$\nabla F_t = \begin{pmatrix} \nabla_{\boldsymbol{\phi}} F_t \\ \nabla_{\boldsymbol{\gamma}} F_t \\ \nabla_{\boldsymbol{c}} F_t \end{pmatrix} \quad , \quad \boldsymbol{\phi} = (\phi_1, ..., \phi_m)^\top \ , \ \boldsymbol{\gamma} = (\gamma_1, ..., \gamma_{m-1})^\top \ , \ \boldsymbol{c} = (c_1, ..., c_{m-1})^\top$$

$$\nabla^2 F_t = \begin{pmatrix} \nabla^2_{\boldsymbol{\phi},\boldsymbol{\phi}} F_t & \nabla^2_{\boldsymbol{\phi},\boldsymbol{\gamma}} F_t & \nabla^2_{\boldsymbol{\phi},\boldsymbol{c}} F_t \\ (\nabla^2_{\boldsymbol{\phi},\boldsymbol{\gamma}} F_t)^\top & \nabla^2_{\boldsymbol{\gamma},\boldsymbol{\gamma}} F_t & \nabla^2_{\boldsymbol{\gamma},\boldsymbol{c}} F_t \\ (\nabla^2_{\boldsymbol{\phi},\boldsymbol{c}} F_t)^\top & (\nabla^2_{\boldsymbol{\gamma},\boldsymbol{c}} F_t)^\top & \nabla^2_{\boldsymbol{c},\boldsymbol{c}} F_t \end{pmatrix}$$

Now take $\boldsymbol{\phi}^{(+)} = \boldsymbol{\phi}$ and $\boldsymbol{\phi}^{(-)} = (\mathbf{0}_{(p+1)\times 1}, \phi_1, ..., \phi_{m-1})^\top$, and also create a transition vector $\boldsymbol{G}_t = (1, G(x_{t-d}, \gamma_1, c_1), ..., G(x_{t-d}, \gamma_{m-1}, c_{m-1}))^\top$. Then the skeleton can be expressed as:

$$F_t = \boldsymbol{G}_t^\top(\boldsymbol{\phi}^{(+)} - \boldsymbol{\phi}^{(-)})\boldsymbol{Y}_t$$

We notice that since parameters $\phi_{j,i}$ are linear, we get $\nabla^2_{\boldsymbol{\phi},\boldsymbol{\phi}} F_t = \mathbf{0}_{m(p+1)\times m(p+1)}$. Similarly, we can then fill in the remaining derivatives:

$$\nabla_{\boldsymbol{\phi}} F_t = \begin{pmatrix} (1 - G(x_{t-d}, \gamma_1, c_1))\boldsymbol{Y}_t \\ (G(x_{t-d}, \gamma_1, c_1) - G(x_{t-d}, \gamma_2, c_2))\boldsymbol{Y}_t \\ \vdots \\ (G(x_{t-d}, \gamma_{m-1}, c_{m-1}) - 0)\boldsymbol{Y}_t \end{pmatrix} \ , \ \nabla_{\boldsymbol{\gamma}} F_t = \begin{pmatrix} \frac{\partial G}{\partial\gamma_1}(x_{t-d}, \gamma_1, c_1)(\phi_2 - \phi_1)^\top\boldsymbol{Y}_t \\ \frac{\partial G}{\partial\gamma_2}(x_{t-d}, \gamma_2, c_2)(\phi_3 - \phi_2)^\top\boldsymbol{Y}_t \\ \vdots \\ \frac{\partial G}{\partial\gamma_{m-1}}(x_{t-d}, \gamma_{m-1}, c_{m-1})(\phi_m - \phi_{m-1})^\top\boldsymbol{Y}_t \end{pmatrix}$$

$$\nabla_{\boldsymbol{c}} F_t = \begin{pmatrix} \vdots \\ \text{analogously to } \nabla_{\boldsymbol{\gamma}} \\ \vdots \end{pmatrix}$$

$$\nabla^2_{\boldsymbol{\phi},\boldsymbol{\gamma}} F_t = \begin{pmatrix} -\frac{\partial(\boldsymbol{G}_t)_2}{\partial\gamma_1}\boldsymbol{Y}_t & \mathbf{0}_{(p+1)\times 1} & \cdots & \mathbf{0}_{(p+1)\times 1} \\ \frac{\partial(\boldsymbol{G}_t)_2}{\partial\gamma_1}\boldsymbol{Y}_t & -\frac{\partial(\boldsymbol{G}_t)_3}{\partial\gamma_2}\boldsymbol{Y}_t & \cdots & \mathbf{0}_{(p+1)\times 1} \\ \mathbf{0}_{(p+1)\times 1} & \frac{\partial(\boldsymbol{G}_t)_3}{\partial\gamma_2}\boldsymbol{Y}_t & \cdots & \mathbf{0}_{(p+1)\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(p+1)\times 1} & \mathbf{0}_{(p+1)\times 1} & \cdots & -\frac{\partial(\boldsymbol{G}_t)_m}{\partial\gamma_{m-1}}\boldsymbol{Y}_t \\ \mathbf{0}_{(p+1)\times 1} & \mathbf{0}_{(p+1)\times 1} & \cdots & \frac{\partial(\boldsymbol{G}_t)_m}{\partial\gamma_{m-1}}\boldsymbol{Y}_t \end{pmatrix} \ , \ \nabla^2_{\boldsymbol{\phi},\boldsymbol{c}} F_t = \begin{pmatrix} \vdots \\ \text{... analogously to } \nabla^2_{\boldsymbol{\phi},\boldsymbol{\gamma}} \text{ ...} \\ \vdots \end{pmatrix}$$

$$\nabla^2_{\boldsymbol{\gamma},\boldsymbol{\gamma}} F_t = \mathrm{diag}\bigg(\frac{\partial G}{\partial \gamma_1}(x_{t-d}, \gamma_1, c_1)(\phi_2 - \phi_1)^\top \boldsymbol{Y}_t, ..., \frac{\partial G}{\partial \gamma_{m-1}}(x_{t-d}, \gamma_{m-1}, c_{m-1})(\phi_m - \phi_{m-1})^\top \boldsymbol{Y}_t \bigg)$$

$$\nabla^2_{\boldsymbol{c},\boldsymbol{c}} F_t \;\text{ and }\; \nabla^2_{\boldsymbol{\gamma},\boldsymbol{c}} F_t = \mathrm{diag}\big(\text{ ... analogously to } \nabla^2_{\boldsymbol{\gamma},\boldsymbol{\gamma}} \text{ ... } \big)$$

Note that all matrices marked with "analogously to..." contain their respective derivatives of the transition function G. These will be filled in according to its respective type (logistic, exponential):

$$G_E(x_{t-d}, \gamma, c) = 1 - \mathrm{e}^{-\gamma(x_{t-d}-c)^2}$$

$$\frac{\partial G_E}{\partial \gamma} = \mathrm{e}^{-\gamma(c-x_{t-d})^2}(c - x_{t-d})^2$$

$$\frac{\partial G_E}{\partial c} = 2\mathrm{e}^{-\gamma(c-x_{t-d})^2}\gamma(c - x_{t-d})$$

$$\frac{\partial^2 G_E}{\partial \gamma \partial c} = -2\mathrm{e}^{-\gamma(c-x_{t-d})^2}(c - x_{t-d})(c^2\gamma - 2cx_{t-d} + \gamma x_{t-d}^2\gamma - 1)$$

$$\frac{\partial^2 G_E}{\partial c^2} = -2\mathrm{e}^{-\gamma(c-x_{t-d})^2}\gamma(2c^2\gamma - 4cx_{t-d}\gamma + 2x_{t-d}^2\gamma - 1)$$

$$\frac{\partial^2 G_E}{\partial \gamma^2} = -\mathrm{e}^{-\gamma(c-x_{t-d})^2}(c - x_{t-d})^4$$

$$G_L(x_{t-d}, \gamma, c) = \frac{1}{1+\mathrm{e}^{-\gamma(x_{t-d}-c)}}$$

$$\frac{\partial G_L}{\partial \gamma} = \frac{\mathrm{e}^{-\gamma(x_{t-d}-c)}(x_{t-d}-c)}{\left(1+\mathrm{e}^{-\gamma(x_{t-d}-c)}\right)^2}$$

$$\frac{\partial G_L}{\partial c} = -\frac{\gamma\mathrm{e}^{-\gamma(x_{t-d}-c)}}{\left(\mathrm{e}^{c\gamma}+\mathrm{e}^{\gamma x_{t-d}}\right)^2}$$

$$\frac{\partial^2 G_L}{\partial \gamma \partial c} = -\frac{\mathrm{e}^{\gamma(x_{t-d}-c)}(1-c\gamma+x_{t-d}\gamma+\mathrm{e}^{\gamma(x_{t-d}-c)}(1+c\gamma-x_{t-d}\gamma))}{(1+\mathrm{e}^{\gamma(x_{t-d}-c)})^3}$$

$$\frac{\partial^2 G_L}{\partial c^2} = \frac{\mathrm{e}^{\gamma(x_{t-d}+c)}\gamma^2(\mathrm{e}^{c\gamma}-\mathrm{e}^{x_{t-d}\gamma})}{(\mathrm{e}^{c\gamma}+\mathrm{e}^{x_{t-d}\gamma})^3}$$

$$\frac{\mathrm{e}^{\gamma(c+x_{t-d})}(c-x_{t-d})^2(\mathrm{e}^{c\gamma}-\mathrm{e}^{x_{t-d}\gamma}))}{(\mathrm{e}^{c\gamma}+\mathrm{e}^{x_{t-d}\gamma})^3}$$

```
##
## LSTAR(2,1,-0.1,10) PhiParams & standard errors:
##  Phi:

## [1]  0.14844846  0.95684360  0.19873972 -0.07726319  0.83045225  0.13664019

## Phi_se:

## [1] 0.45350968 1.03250814 0.10470951 0.09760295 0.22380264 0.10832803

##
## LSTAR(2,1,-0.1,0.1,10,11) PhiParams & standard errors:
##  Phi:

## [1] -0.21249835  0.32557665  0.04959413  0.03538595 -2.06519468  0.70066290
## [7]  0.09524044  0.75428752 -0.07661200

## Phi_se:

## [1] 0.5193461 0.8390685 0.5202534 0.2855643 6.3644170 3.1718359 1.4882409
## [8] 1.9094895 0.2314645
```

This procedure can then be added to the postprocessing method which produces a STAR model with the following attributes:

```
## LSTAR(2, 2, 0, 10), m=2 :

## List of 16
##  $ name       : chr "LSTAR(2,2,0,10)"
##  $ nReg       : int 2
##  $ type       : chr "logistic"
##  $ p          : num 2
##  $ d          : num 2
##  $ c          : num 0
##  $ gamma      : num 10
##  $ data       : num [1:189] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 189
```

```
##  $ skel       : num [1:187, 1] 0.0947 0.1933 0.118 -0.0343 -0.0808 ...
##  $ residuals  : num [1:187, 1] 0.1053 -0.0933 -0.218 -0.0657 -0.0192 ...
##  $ resSigmaSq : num 0.027
##  $ PhiParams  : num [1:6] -0.0684 0.2047 0.3672 0.1028 0.9621 ...
##  $ PhiStErrors: num [1:6] 0.0717 0.2488 0.1165 0.078 0.229 ...
##  $ AIC        : num 1233
##  $ BIC        : num 1259
```

```
## LSTAR(2, 2, 0, 0.2, 10, 12), m=3 :
```

```
## List of 16
##  $ name       : chr "LSTAR(2,2,0,0.2,10,12)"
##  $ nReg       : int 3
##  $ type       : chr "logistic"
##  $ p          : num 2
##  $ d          : num 2
##  $ c          : num [1:2] 0 0.2
##  $ gamma      : num [1:2] 10 12
##  $ data       : num [1:189] 0.3 0.1 0.2 0.1 -0.1 ...
##  $ n          : int 189
##  $ skel       : num [1:187, 1] 0.1003 0.1632 0.1187 -0.0264 -0.0782 ...
##  $ residuals  : num [1:187, 1] 0.0997 -0.0632 -0.2187 -0.0736 -0.0218 ...
##  $ resSigmaSq : num 0.0267
##  $ PhiParams  : num [1:9] -0.0347 0.3063 0.3804 0.0273 0.5822 ...
##  $ PhiStErrors: num [1:9] 0.0421 0.2574 0.1733 0.0707 0.1857 ...
##  $ AIC        : num 1245
##  $ BIC        : num 1287
```

6.3 2-Regime STAR Estimation Procedure

Now that we've prepared all necessary methods, we proceed to implementing an estimation procedure, similar to the search for SETAR models in sections 2.3 and 4.4:

```
## hypars=
```

```
##    p d
## 3  2 2
## 8  4 2
## 12 5 2
## 22 7 1
## 23 7 2
## 30 8 2
## 38 9 2
```

We search through the above hyperparameter array, and for

$$c = -0.39, -0.3742, -0.3584...NA, NA, NA, NA$$

and

$$\gamma = 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10$$

```
##   transition p d      c gamma       AIC       BIC resSigmaSq
## 1   logistic 2 2 0.1314    10 1235.1634 1261.0974     0.0267
## 2   logistic 4 2  0.084    10 1244.4277 1283.3287     0.0262
## 3   logistic 5 2 0.0682    10 1251.7345  1297.119     0.0256
## 4   logistic 7 1    0.4    10 1262.9404 1321.2918     0.0249
## 5   logistic 7 2 0.0524    10   1262.99 1321.3414     0.0249
## 6   logistic 8 2    0.4    10 1270.2884 1335.1233     0.0244
## 7   logistic 9 2    0.4    10 1275.5283 1346.8468     0.0241
```

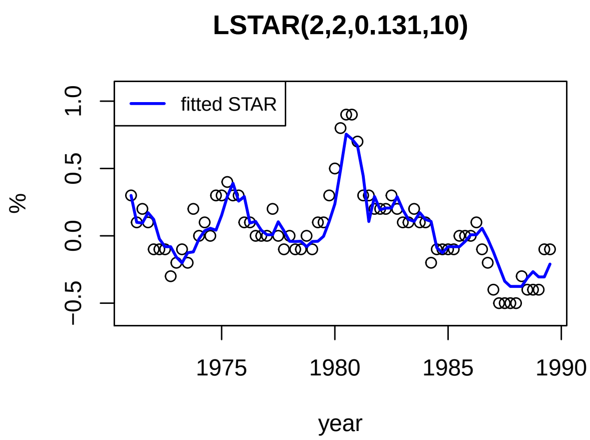
Again, we sorted the models by their BIC. We seem to have a rather small sample, but that is most likely due to only having 2 models with detected $STAR$-type nonlinearity. Estimated parameters were obtained during the postprocessing phase on a model with lowest $\hat{\sigma}^2_\varepsilon$.

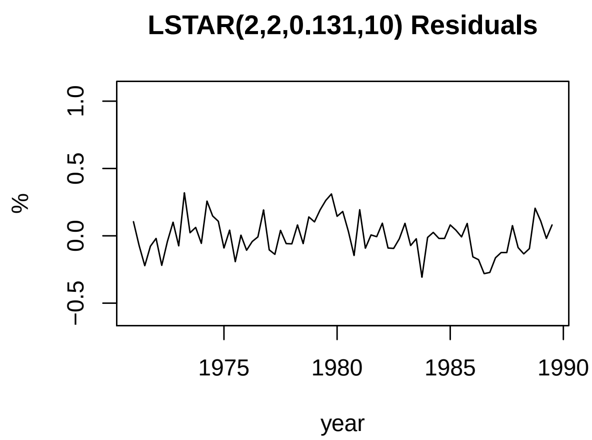
On top of that the postprocessing phase also contains an estimation of standard errors of parameters:

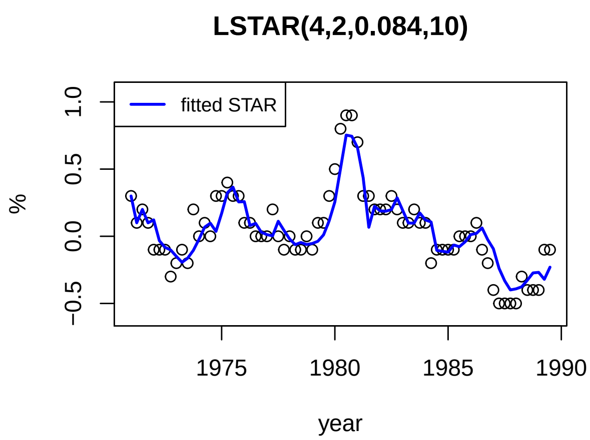
```
##                 [,1]      [,2]      [,3]      [,4]      [,5]       [,6]
## Phi      -0.02606206 0.3141173 0.3861456 0.1340813 1.1096505 -0.5167062
## stdError  0.06736699 0.1824448 0.1283139 0.1104117 0.1799038  0.2961448
```

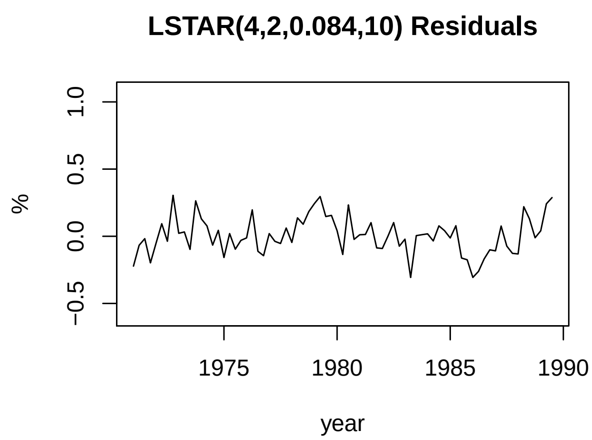
6.4 Visualisation

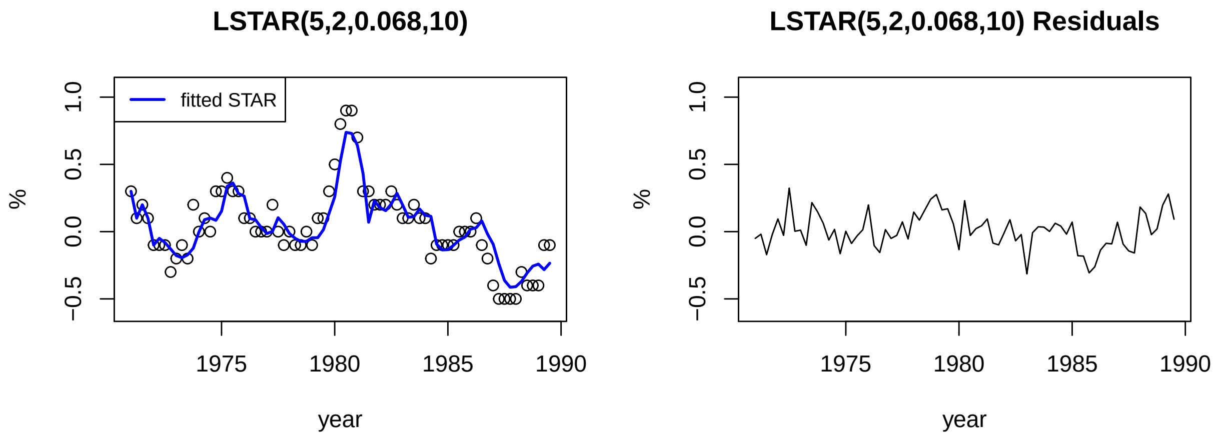
We can now visualize the results of the top 3 models:





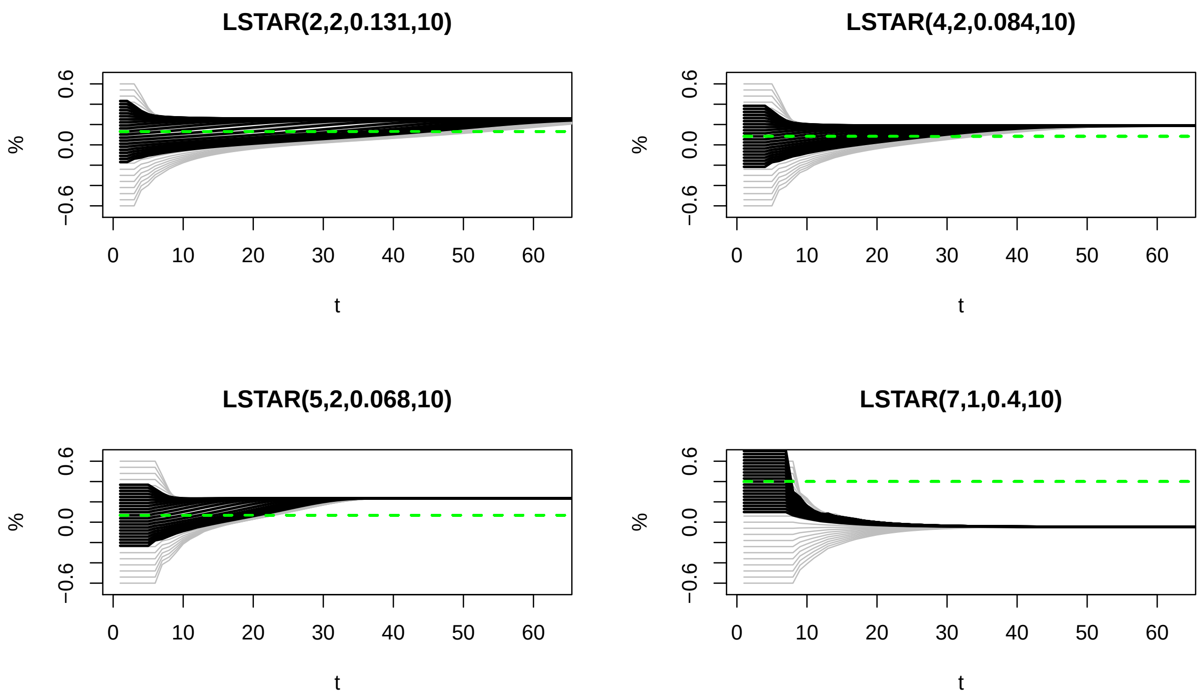


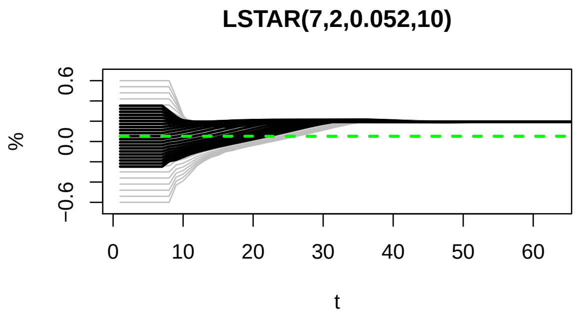


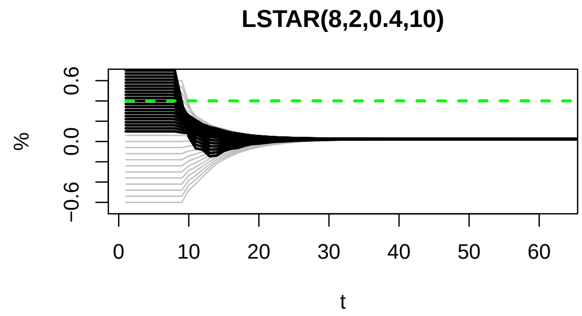


6.5 Deterministic Properties

Naturally, it is just as useful to understand the deterministic evolution of the model, as it is in the case of SETAR models from section 2.4.








```
## model equilibria:

## LSTAR(2,2,0.131,10) LSTAR(4,2,0.084,10) LSTAR(5,2,0.068,10)   LSTAR(7,1,0.4,10)
##              0.2575              0.1883              0.2329             -0.0456
## LSTAR(7,2,0.052,10)    LSTAR(8,2,0.4,10)   LSTAR(9,2,0.4,10)
##              0.1951              0.0243              0.0069
```

6.6 Conclusion

The STAR models, selected in this chapter were sampled by the results of nonlinearity tests. Here we see them ordered by their BIC:

```
##   STAR2 search results:
## 1   LSTAR(2,2,0.131,10)
## 2   LSTAR(4,2,0.084,10)
## 3   LSTAR(5,2,0.068,10)
## 4     LSTAR(7,1,0.4,10)
## 5   LSTAR(7,2,0.052,10)
## 6     LSTAR(8,2,0.4,10)
## 7     LSTAR(9,2,0.4,10)
```

And in order to describe its inner dynamics, we can write the first model explicitly as:

$$\begin{aligned} X_t = &((-0.0261 \pm 0.0674) + (0.3141 \pm 0.1824)X_{t-1} + (0.3861 \pm 0.1283)X_{t-2})\ [1 - G_L(X_{t-2}, 0.1314, 10)] + \qquad \hat{\sigma}_\varepsilon^2 = 0.0267 \\ &((0.1341 \pm 0.1104) + (1.1097 \pm 0.1799)X_{t-1} + (-0.5167 \pm 0.2961)X_{t-2})\ G_L(X_{t-2}, 0.1314, 10) + \varepsilon_t \end{aligned}$$

Besides an $LSTAR$ exclusivity in the selected sample, we notice that the upper bound of the γ parameter (`gamma = 10`) seems to be selected, which might suggest a preference for a steep transition between regimes.

7 STAR Model Diagnostics, STAR3, and Predictions

In this chapter we conclude the $STAR$ evaluation of our data by testing for remaining $STAR$-3 nonlinearity, and follow by performing forecasts on the resulting models. Fortunately, we have prepared a set of general methods for m-regimes, which means that we only need to change the default regime count to `m = 3` when estimating additional models.

7.1 Autocorrelation of Residuals

The null hypothesis of this test claims that the residuals of a nonlinear model are uncorrelated.

Under the assuption of the null hypothesis, the test statistic LM is asymptotically $\chi^2(q)$-distributed, which means that we reject the null hypothesis on $\alpha \cdot 100\%$ significance level if `p-value` $< \alpha/q$.

7.2 Remaining STAR Nonlinearity

The lack of correlation in the residuals serves only as a primary filter, and does not imply a remaining nonlinearity. This requires a specialized test with a null hypothesis that a 2-regime $STAR$ is sufficient.

Again, the LM statistic is asymptotically $\chi^2(3p)$-distributed if the null hypothesis is true for an LSTAR nonlinearity ($\chi^2(2p)$-distributed for ESTAR), so we compare the p-value with a corrected α.

7.3 SETAR2 Diagnostics and Evaluation

```
##                      p-val(res autocorr) alpha_corr   p-val(nlin3) alpha_corr
## LSTAR(2,2,0.131,10)               0.2729       0.05         0.0629     0.0083
## LSTAR(4,2,0.084,10)               0.0668       0.05 .        0.003     0.0042 *
## LSTAR(5,2,0.068,10)               0.1444       0.05         0.0082     0.0033
## LSTAR(7,1,0.4,10)                 0.0827       0.05 .       0.0301     0.0024
## LSTAR(7,2,0.052,10)               0.1905       0.05         0.0337     0.0024
## LSTAR(8,2,0.4,10)                 0.4197       0.05         0.0721     0.0021
## LSTAR(9,2,0.4,10)                 0.5714       0.05         0.0366     0.0019
```

As it appears, only one model contains possible $STAR$-3 nonlinearity. Hence we sample the delay space

```
##  unique delays:
## [1] 2
```

7.4 Estimating 3-regime STAR Models

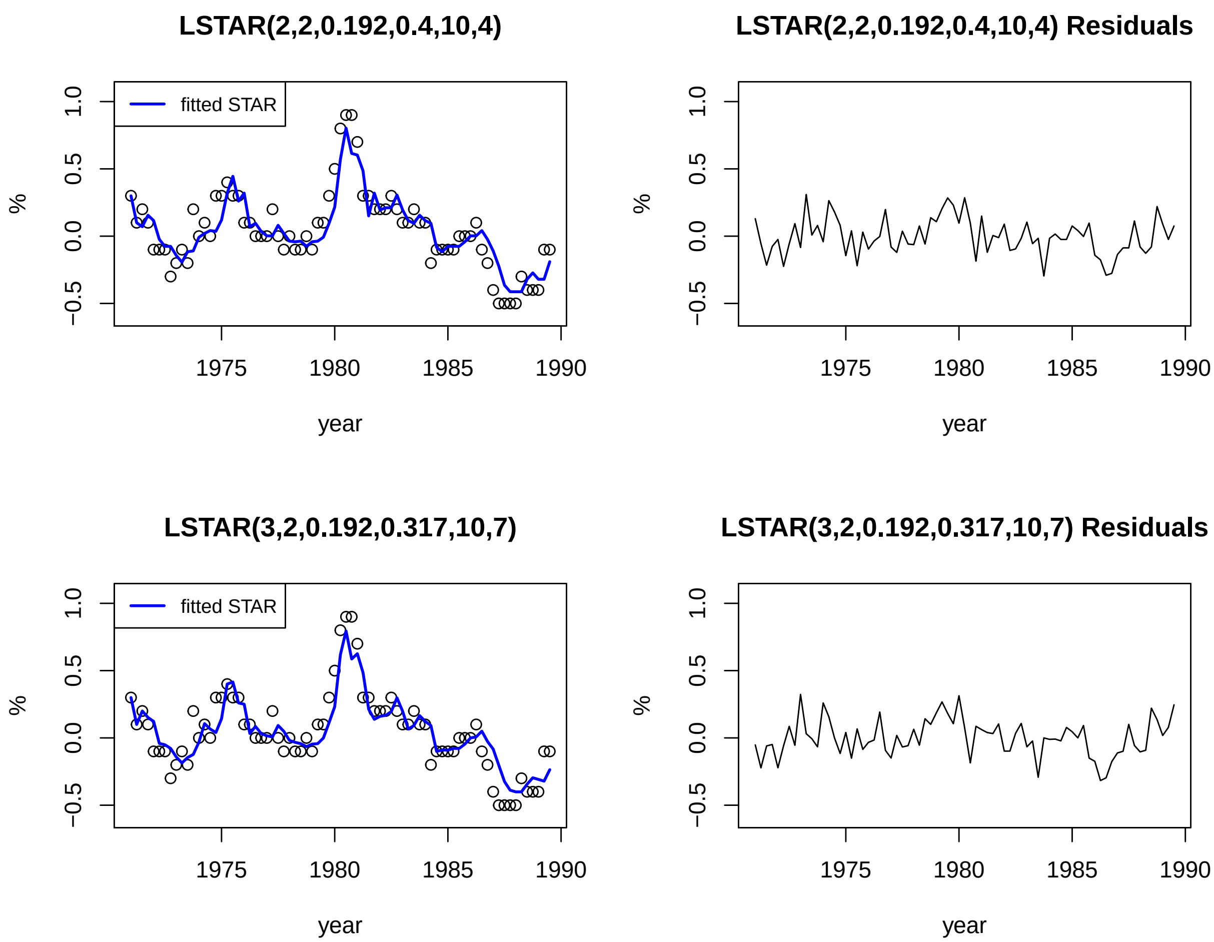
Similarly to section 4.4, we compose an overall estimation procedure to search for STAR-3 candidates:

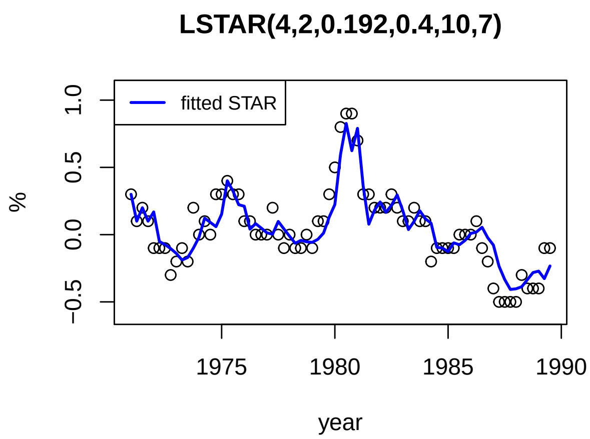
```
##
## (p,d)=(2,2):==>sigmaSq(2,2,0.1921,0.4,10,4) = 0.0262169373233735
## (p,d)=(3,2):==>sigmaSq(3,2,0.1921,0.3168,10,7) = 0.0253911000849095
## (p,d)=(4,2):==>sigmaSq(4,2,0.1921,0.4,10,7) = 0.0240654930912324
## (p,d)=(5,2):==>sigmaSq(5,2,0.1921,0.4,10,7) = 0.0232941687863673
## (p,d)=(6,2):==>sigmaSq(6,2,0.1505,0.4,10,7) = 0.0233832307276683
## (p,d)=(7,2):==>sigmaSq(7,2,0.1505,0.4,10,10) = 0.0228685482505765
## (p,d)=(8,2):==>sigmaSq(8,2,0.1089,0.4,10,10) = 0.021804943890685
## (p,d)=(9,2):==>sigmaSq(9,2,0.0674,0.4,10,10) = 0.021457673718699
## (p,d)=(10,2):==>sigmaSq(10,2,0.0674,0.4,10,10) = 0.0213223829658001
## (p,d)=(11,2):==>sigmaSq(11,2,0.0258,0.4,10,10) = 0.021409269277794
## (p,d)=(12,2):==>sigmaSq(12,2,-0.0574,0.4,10,10) = 0.0206992656742827

##
##  elapsed time: 7.1 min

##    transition  p d      c1     c2 gamma1 gamma2       AIC       BIC resSigmaSq
## 1    logistic  2 2  0.1921    0.4     10      4 1248.5738 1290.7166     0.0262
## 2    logistic  3 2  0.1921 0.3168     10      7 1259.6232 1311.4911     0.0254
## 3    logistic  4 2  0.1921    0.4     10      7 1274.7573 1336.3505     0.0241
## 4    logistic  5 2  0.1921    0.4     10      7 1285.9141 1357.2326     0.0233
## 5    logistic  6 2  0.1505    0.4     10      7 1290.1929 1371.2366     0.0234
## 6    logistic  7 2  0.1505    0.4     10     10 1299.3994 1390.1683     0.0229
## 7    logistic  8 2  0.1089    0.4     10     10 1313.4007 1413.8948     0.0218
## 8    logistic  9 2  0.0674    0.4     10     10  1321.435 1431.6544     0.0215
## 9    logistic 10 2  0.0674    0.4     10     10 1327.6304  1447.575     0.0213
## 10   logistic 11 2  0.0258    0.4     10     10 1331.8618 1461.5317     0.0214
## 11   logistic 12 2 -0.0574    0.4     10     10 1343.2359 1482.6311     0.0207
```

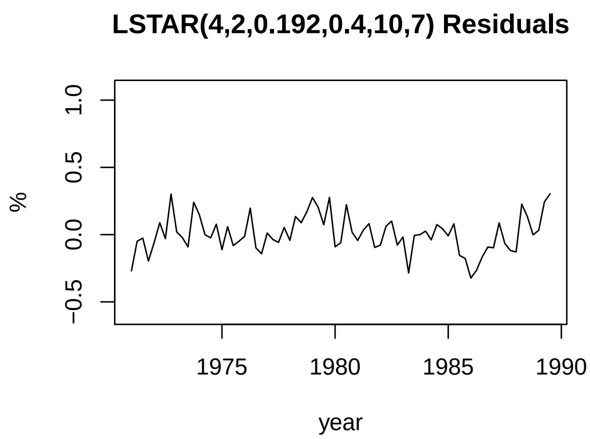
with standard errors:

```
##                 [,1]      [,2]      [,3]     [,4]      [,5]      [,6]       [,7]
## Phi       0.1357175 0.5318766 0.5163462 0.458175  2.911078 -1.537255 -0.5815558
## stdError 4.2600965 6.4185601 0.8612561 8.752651 35.393510 27.254081  8.7554784
##                  [,8]      [,9]
## Phi        0.2732723  0.879196
## stdError 12.4947358 17.490344
```

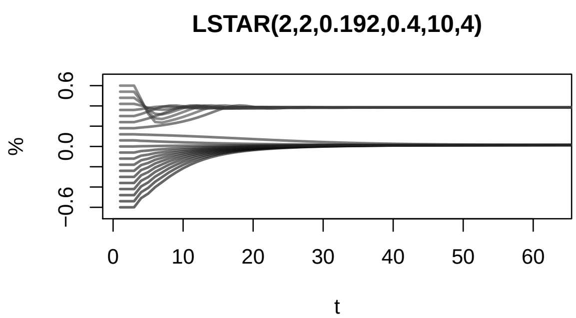
Visualisation

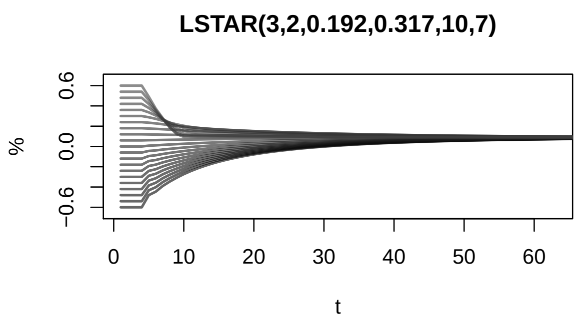


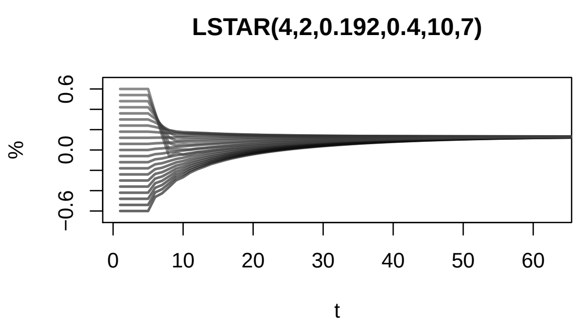

LSTAR(4,2,0.192,0.4,10,7)
fitted STAR
%
−0.5
0.0
0.5
1.0
1975
1980
1985
1990
year

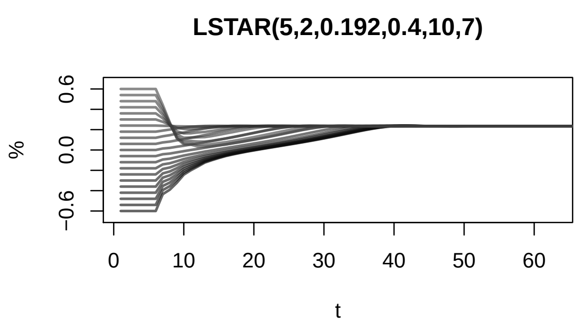

LSTAR(4,2,0.192,0.4,10,7) Residuals
%
−0.5
0.0
0.5
1.0
1975
1980
1985
1990
year

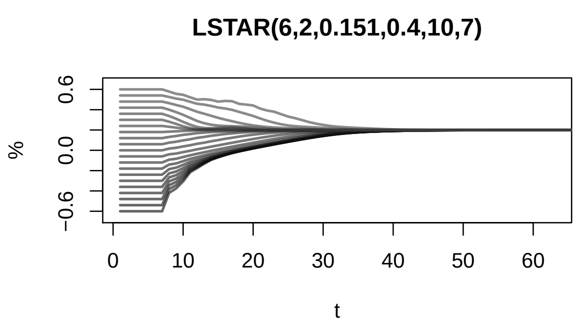
Equilibria

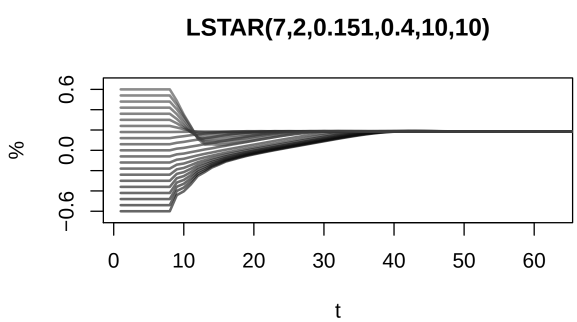

LSTAR(2,2,0.192,0.4,10,4)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t


LSTAR(3,2,0.192,0.317,10,7)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t


LSTAR(4,2,0.192,0.4,10,7)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t


LSTAR(5,2,0.192,0.4,10,7)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t


LSTAR(6,2,0.151,0.4,10,7)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t


LSTAR(7,2,0.151,0.4,10,10)
%
−0.6
0.0
0.6
0
10
20
30
40
50
60
t

```
## model equilibria:

## $`LSTAR(2,2,0.192,0.4,10,4)`
## [1] 0.0143 0.3842
##
## $`LSTAR(3,2,0.192,0.317,10,7)`
## [1] 0.0894
##
## $`LSTAR(4,2,0.192,0.4,10,7)`
## [1] 0.1321
##
## $`LSTAR(5,2,0.192,0.4,10,7)`
## [1] 0.2327
##
## $`LSTAR(6,2,0.151,0.4,10,7)`
## [1] 0.1993
##
## $`LSTAR(7,2,0.151,0.4,10,10)`
## [1] 0.1849
##
## $`LSTAR(8,2,0.109,0.4,10,10)`
## [1] 0.1554
##
## $`LSTAR(9,2,0.067,0.4,10,10)`
## [1] 0.1456
##
## $`LSTAR(10,2,0.067,0.4,10,10)`
## [1] 0.1469
##
## $`LSTAR(11,2,0.026,0.4,10,10)`
## [1] 0.1437
##
## $`LSTAR(12,2,-0.057,0.4,10,10)`
## [1] 0.1324
```

Results

For the purpose of demonstration we write out the explicit equation of the highest ranked STAR3 model:

$$X_t = \begin{array}{c} ((0.1357 \pm 4.2601) + (0.5319 \pm 6.4186)X_{t-1} + (0.5163 \pm 0.8613)X_{t-2})\ [1 - G_L(X_{t-2}, 0.1921, 10)] + \\ ((0.4582 \pm 8.7527) + (2.9111 \pm 35.3935)X_{t-1} + (-1.5373 \pm 27.2541)X_{t-2})\ [G_L(X_{t-2}, 0.1921, 10) - G_L(X_{t-2}, 0.4, 4)] + \\ ((-0.5816 \pm 8.7555) + (0.2733 \pm 12.4947)X_{t-1} + (0.8792 \pm 17.4903)X_{t-2})\ G_L(X_{t-2}, 0.4, 4) + \varepsilon_t \end{array}$$

$\hat{\sigma}^2_\varepsilon = 0.0262$

7.5 Predictions

Now that we replaced the models with remaining STAR-3 nonlinearity with their 3-regime variants, we can proceed to test the resulting models' predictive abilities. We will be using a predict procedure `PredictSTAR` similar to `PredictSETAR` from section 5.1.

Single-step test:

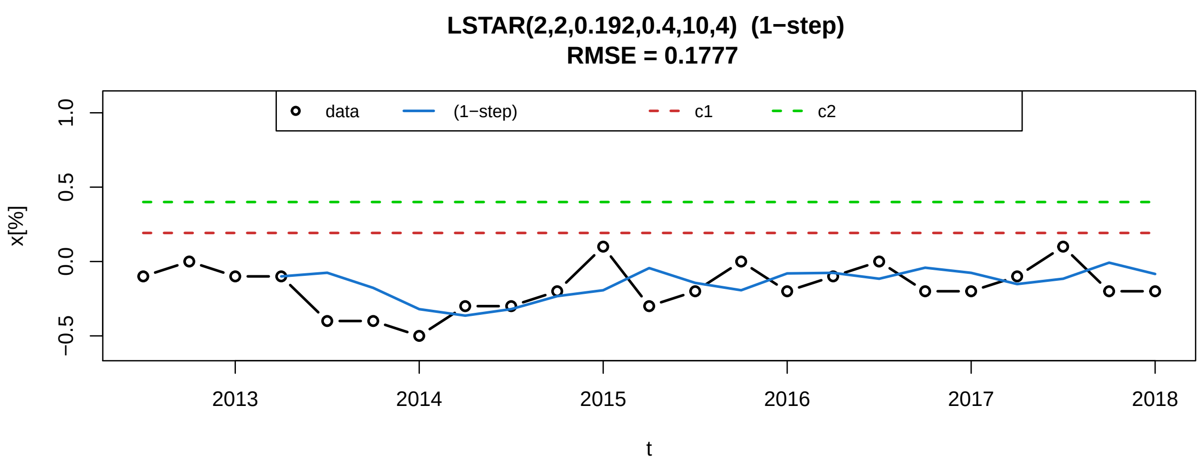
```
## naive (1-step):

##  [1] -0.100 -0.100 -0.076 -0.177 -0.320 -0.364 -0.320 -0.233 -0.193 -0.044
## [11] -0.144
```

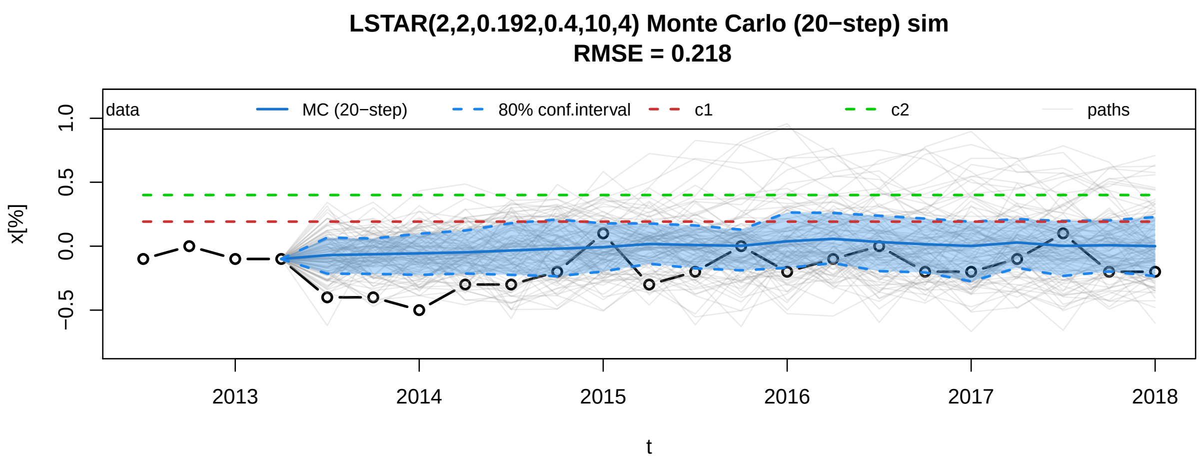
```
## data:
## [1] -0.1 -0.4 -0.4 -0.5 -0.3 -0.3 -0.2  0.1 -0.3 -0.2  0.0
```

Multi-step test:

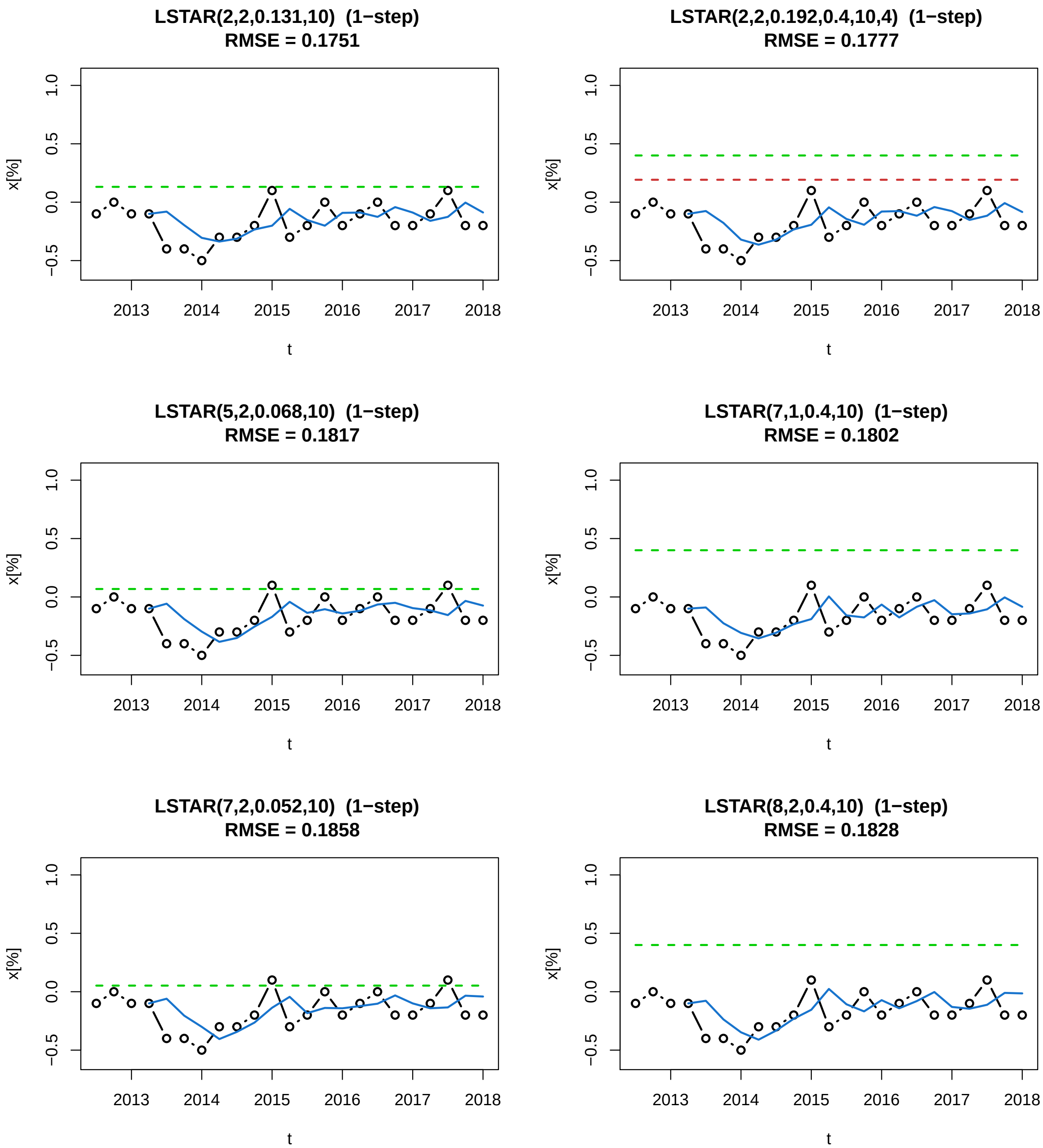
```
## naive (10-step):
##  [1] -0.100 -0.100 -0.076 -0.067 -0.054 -0.046 -0.038 -0.032 -0.026 -0.021
## [11] -0.017
## Monte Carlo (10-step):
##  [1] -0.100 -0.100 -0.075 -0.078 -0.031 -0.075 -0.046 -0.030 -0.049 -0.022
## [11] -0.008
## Bootstrap (10-step):
##  [1] -0.100 -0.100 -0.085 -0.056 -0.079 -0.065 -0.058 -0.018  0.000  0.009
## [11]  0.030
## data:
##  [1] -0.1 -0.4 -0.4 -0.5 -0.3 -0.3 -0.2  0.1 -0.3 -0.2  0.0
```

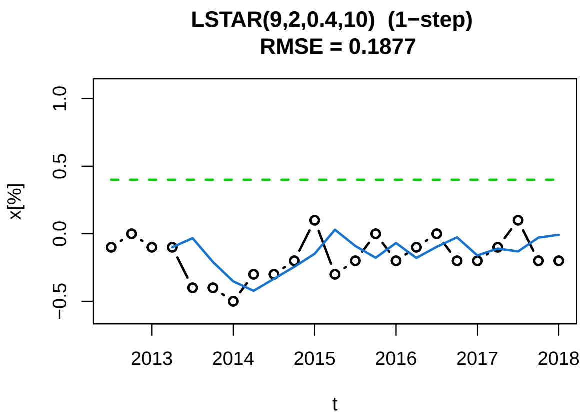
We will also use parts of the previous `predictSETAR_andPlot` method.





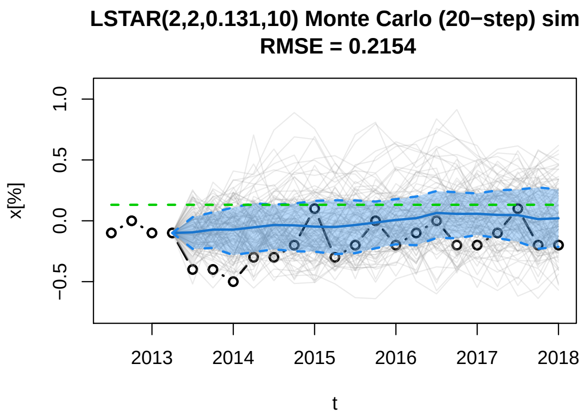
Single-Step Predictions

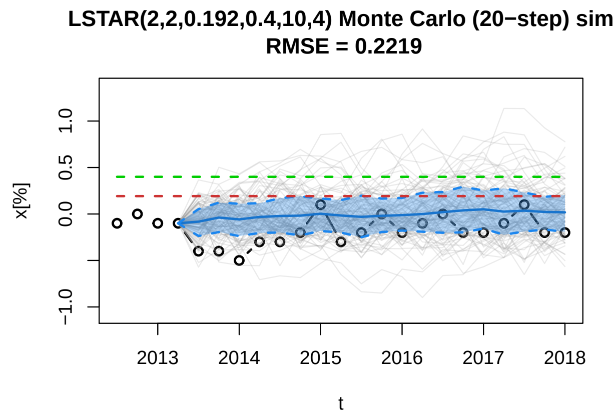


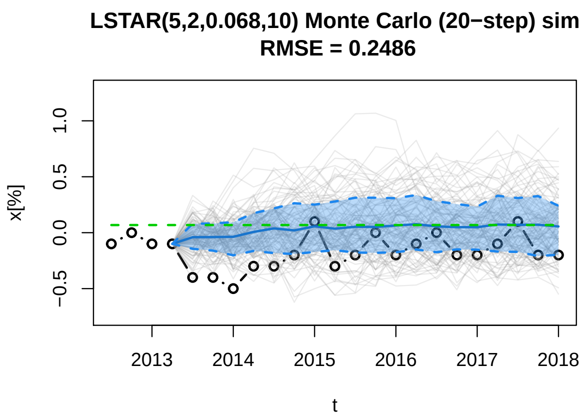


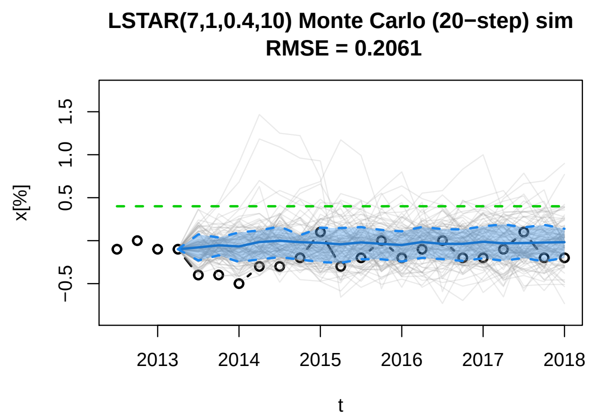
Multi-Step Predictions

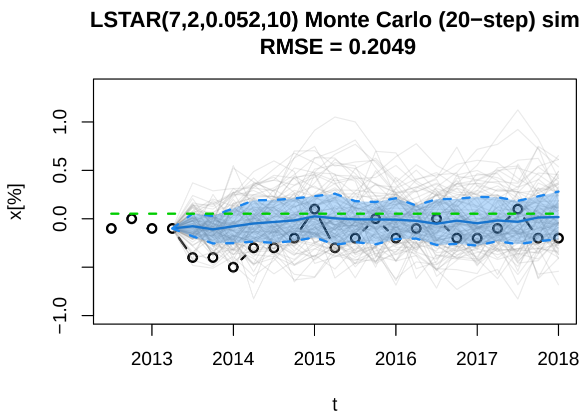
Monte Carlo:

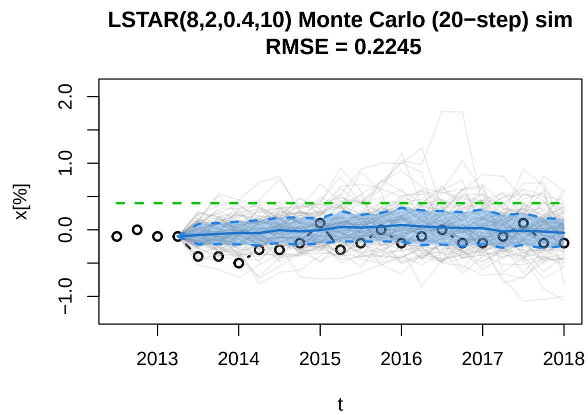


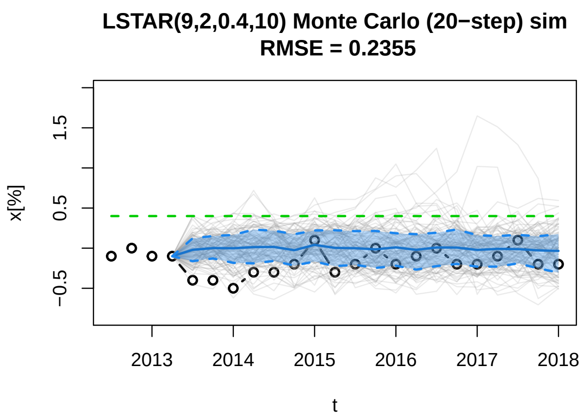




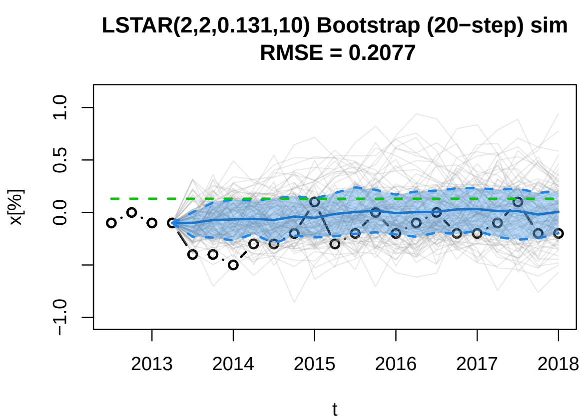


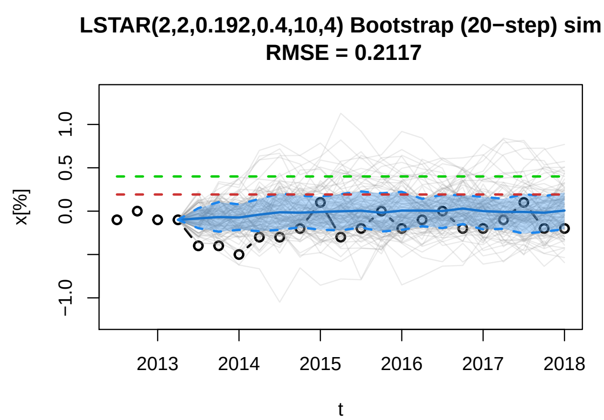

LSTAR(7,2,0.052,10) Monte Carlo (20−step) sim
RMSE = 0.2049
x[%]
−1.0
0.0
0.5
1.0
2013
2014
2015
2016
2017
2018
t

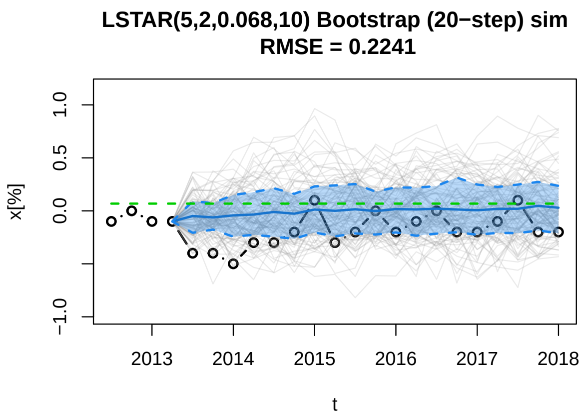

LSTAR(8,2,0.4,10) Monte Carlo (20−step) sim
RMSE = 0.2245
x[%]
−1.0
0.0
1.0
2.0
2013
2014
2015
2016
2017
2018
t

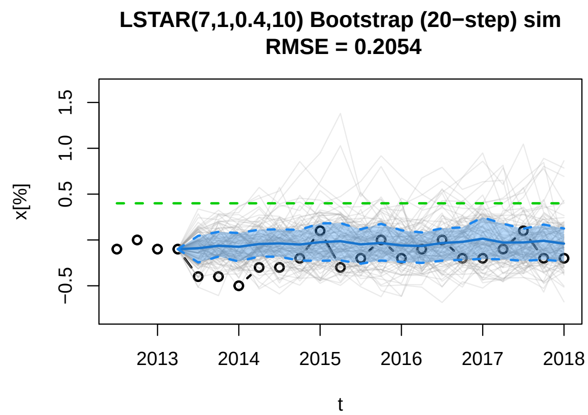

LSTAR(9,2,0.4,10) Monte Carlo (20−step) sim
RMSE = 0.2355
x[%]
−0.5
0.5
1.5
2013
2014
2015
2016
2017
2018
t

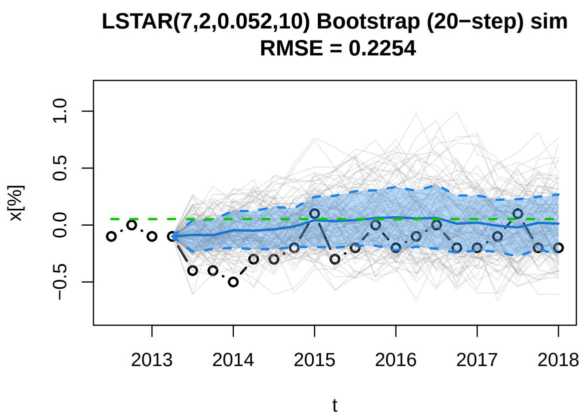
Bootstrap:

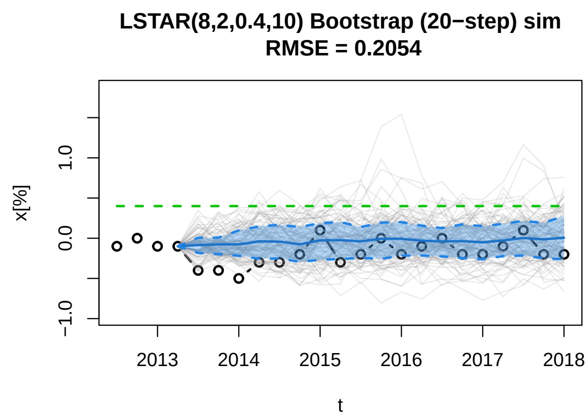

LSTAR(2,2,0.131,10) Bootstrap (20−step) sim
RMSE = 0.2077
x[%]
−1.0
0.0
0.5
1.0
2013
2014
2015
2016
2017
2018
t

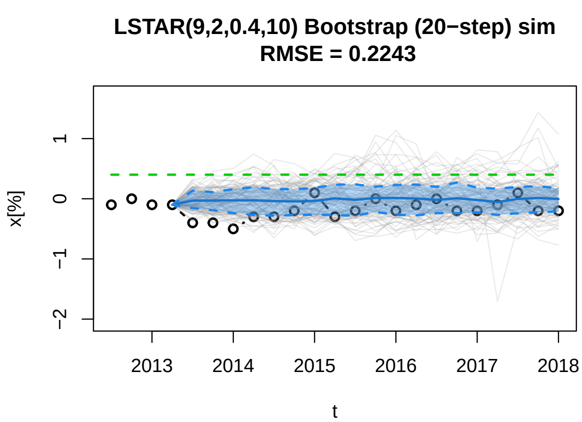

LSTAR(2,2,0.192,0.4,10,4) Bootstrap (20−step) sim
RMSE = 0.2117
x[%]
−1.0
0.0
0.5
1.0
2013
2014
2015
2016
2017
2018
t











7.6 Conclusion and STAR Model Evaluation

Compared to SETAR predictions from chapter 5, we do not have any wildly divergent models. The mean paths of multi-step predictions quickly stabilize, with only slight fluctuations around the local equilibrium.

```
##                        model      BIC sigmaSq MSE(1-step)  MSE(MC)  MSE(boot)
## 1        LSTAR(2,2,0.131,10) 1261.097  0.0267      0.0307   0.0452     0.0483
## 2 LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262      0.0316   0.0483     0.0444
## 3        LSTAR(5,2,0.068,10) 1297.119  0.0256      0.0330   0.0456     0.0510
## 4          LSTAR(7,1,0.4,10) 1321.292  0.0249      0.0325   0.0351     0.0398
## 5        LSTAR(7,2,0.052,10) 1321.341  0.0249      0.0345   0.0415     0.0637
```

```
## 6         LSTAR(8,2,0.4,10) 1335.123  0.0244      0.0334   0.0424     0.0465
## 7         LSTAR(9,2,0.4,10) 1346.847  0.0241      0.0352   0.0559     0.0513
```

In the above table, we observe all examined STAR models, ordered by their ability to fit the data (given by BIC) from best to less accurate.

Unlike in the case of SETAR models in section 5.5, when comparing their MSE with $\hat{\sigma}_{\varepsilon}^2$, we see no divergence from the evaluation data. Instead we observe a rather stable behavior of all $STAR$ predictions.

Now we pick the best models according to their given MSE depending on the prediction method (`"naive"`, `"mc"`, `"boot"`):

```
## Models sorted by 1-step naive MSE:

##                       model      BIC sigmaSq MSE(1-step)
## 1       LSTAR(2,2,0.131,10) 1261.097  0.0267      0.0307
## 2 LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262      0.0316
## 4         LSTAR(7,1,0.4,10) 1321.292  0.0249      0.0325
## 3       LSTAR(5,2,0.068,10) 1297.119  0.0256      0.0330
## 6         LSTAR(8,2,0.4,10) 1335.123  0.0244      0.0334
## 5       LSTAR(7,2,0.052,10) 1321.341  0.0249      0.0345
## 7         LSTAR(9,2,0.4,10) 1346.847  0.0241      0.0352

## Models sorted by Monte Carlo MSE:

##                       model      BIC sigmaSq MSE_mc
## 4         LSTAR(7,1,0.4,10) 1321.292  0.0249 0.0351
## 5       LSTAR(7,2,0.052,10) 1321.341  0.0249 0.0415
## 6         LSTAR(8,2,0.4,10) 1335.123  0.0244 0.0424
## 1       LSTAR(2,2,0.131,10) 1261.097  0.0267 0.0452
## 3       LSTAR(5,2,0.068,10) 1297.119  0.0256 0.0456
## 2 LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262 0.0483
## 7         LSTAR(9,2,0.4,10) 1346.847  0.0241 0.0559

## Models sorted by Bootstrap MSE:

##                       model      BIC sigmaSq MSE_boot
## 4         LSTAR(7,1,0.4,10) 1321.292  0.0249   0.0398
## 5       LSTAR(7,2,0.052,10) 1321.341  0.0249   0.0637
## 6         LSTAR(8,2,0.4,10) 1335.123  0.0244   0.0465
## 1       LSTAR(2,2,0.131,10) 1261.097  0.0267   0.0483
## 3       LSTAR(5,2,0.068,10) 1297.119  0.0256   0.0510
## 2 LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262   0.0444
## 7         LSTAR(9,2,0.4,10) 1346.847  0.0241   0.0513
```

And as we see, the 3-regime model seems to be better at multiple step forecasting, while the order remains almost unchanged for 1-step predictions when compared to ordering by BIC. This implies that test data fit accuracy of chosen STAR models also gives rise to reasonable forecasting precision.

8 Markov-Switching Models

Markov Switching models rely on an exogenous random variable s_t that cannot be directly observed, we can only estimate probabilities of the process switching between individual regimes.

8.1 Methods

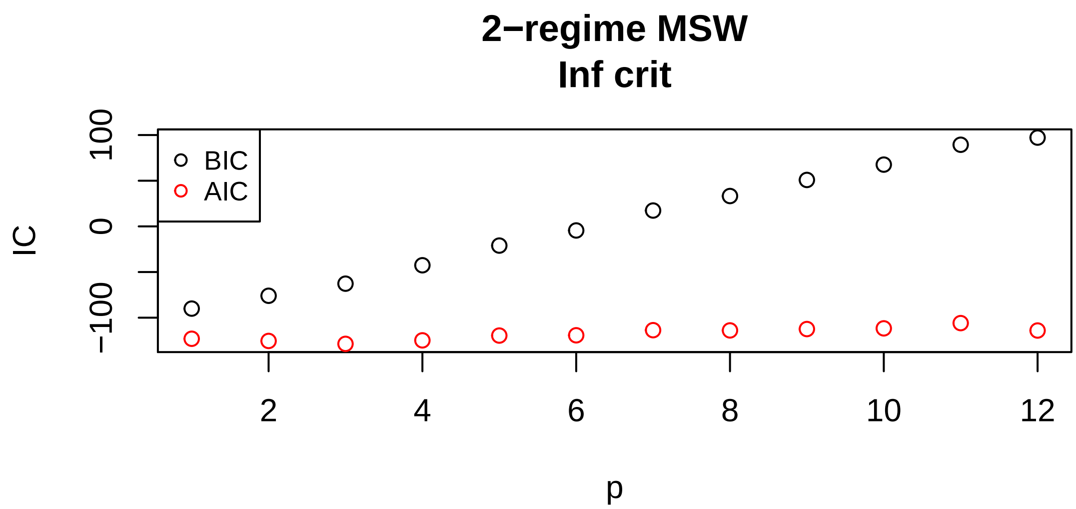
We begin by suplying a dummy fitted linear model to the `msmFit` function.

Then we do this for the following orders p:

```
## 2-reg:   p = 1  2  3  4  5  6  7  8  9  10  11  12  3-reg:   p = 1  2  3  4  5  6  7  8  9  10  11  12
```

8.2 A 2-Regime MSW Model

We will use the above procedure to pool the results in search of parameter p minimizing BIC:



and save the result. We can easily access the fitted `msm` (regime-to-regime) transition probability matrix $\mathbf{P}$ through a property `@transMat`

```
## transProbMatrix =

##        [,1]   [,2]
## [1,] 0.9592 0.0363
## [2,] 0.0408 0.9637
```

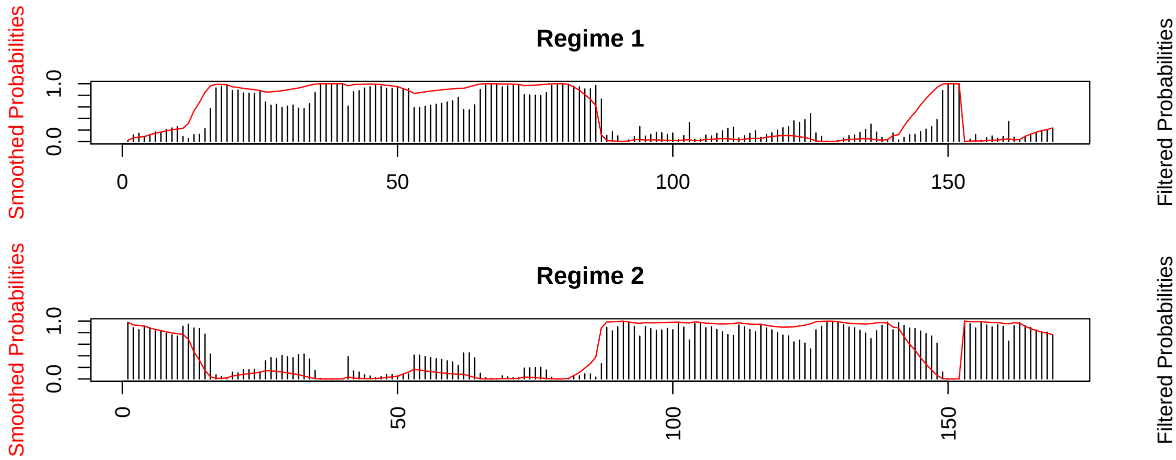
Another important result is the ergodic probability vector $\boldsymbol{\pi} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{e}_{m+1}$ where $\mathbf{A} = (\mathbf{P}^\top - \mathbf{I}_m, \mathbf{1}_m^\top)^\top$ and $\mathbf{1}_m = (1, ..., 1)^\top$ with the $\mathbf{e}_{m+1} = (\mathbf{I}_{m+1})_{.\;,m+1}$ (column):

```
## ergodicProbs =

##           [,1]
## [1,] 0.4705894
## [2,] 0.5294106
```

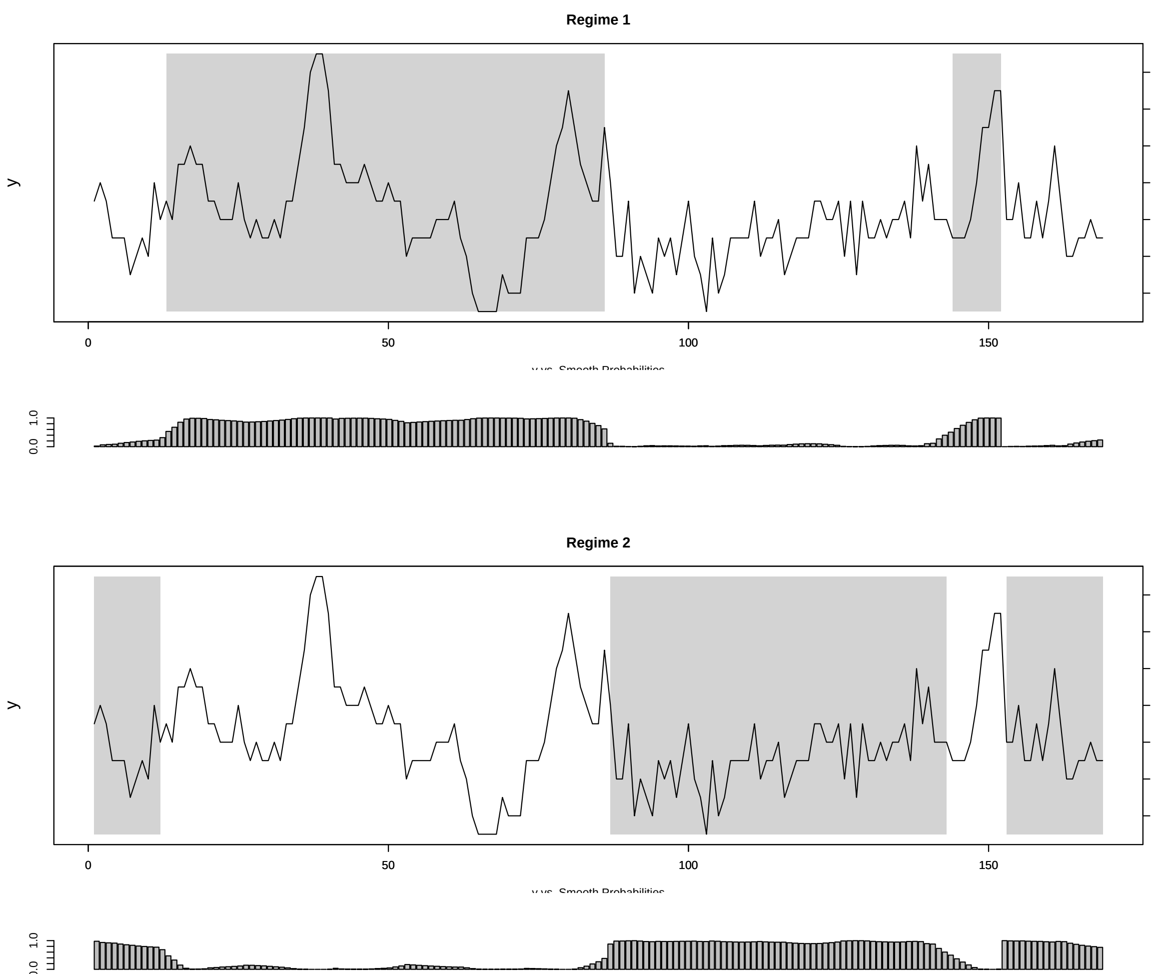
On top of that, we can show the coefficient errors:

```
## 2-regime AR coeffs:

##   (Intercept)       y_1
## 1  0.02095134 0.9166278
## 2 -0.04696155 0.2837363

## sErrors:

##   (Intercept)        y_1
## 1  0.01727896 0.05326628
## 2  0.02079514 0.09867739
```

Moreover, we can plot time components of filtered and smoothed probabilities for each regime:



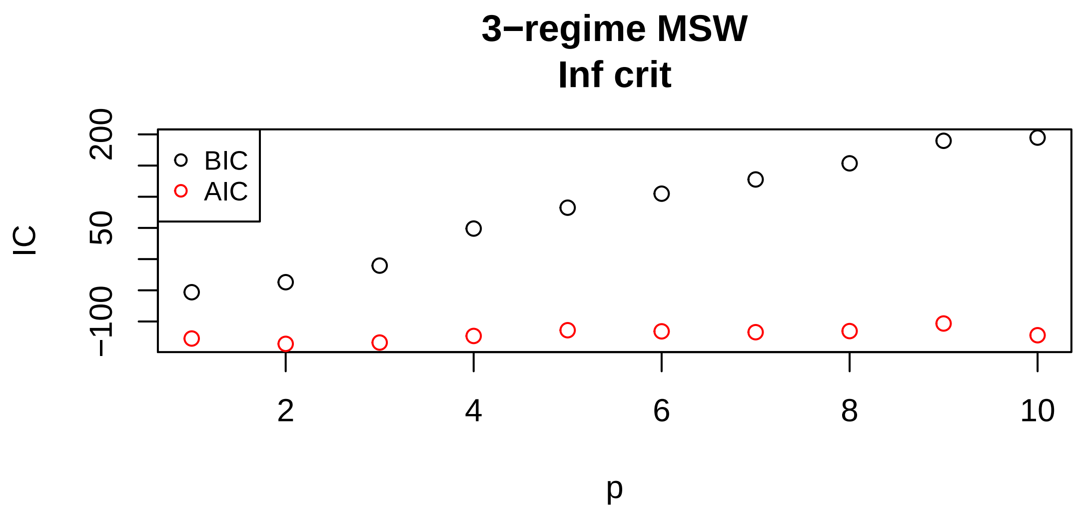
and also mark the portions of the time series, appearing in each regime, from the filtered probabilities



Thus, we can observe that parts of the observed time series, with maximal filtered probability in the first regime, behave with lower volatility, and higher frequency of fluctuation, when compared to parts with maximum filtered probability in the second regime.

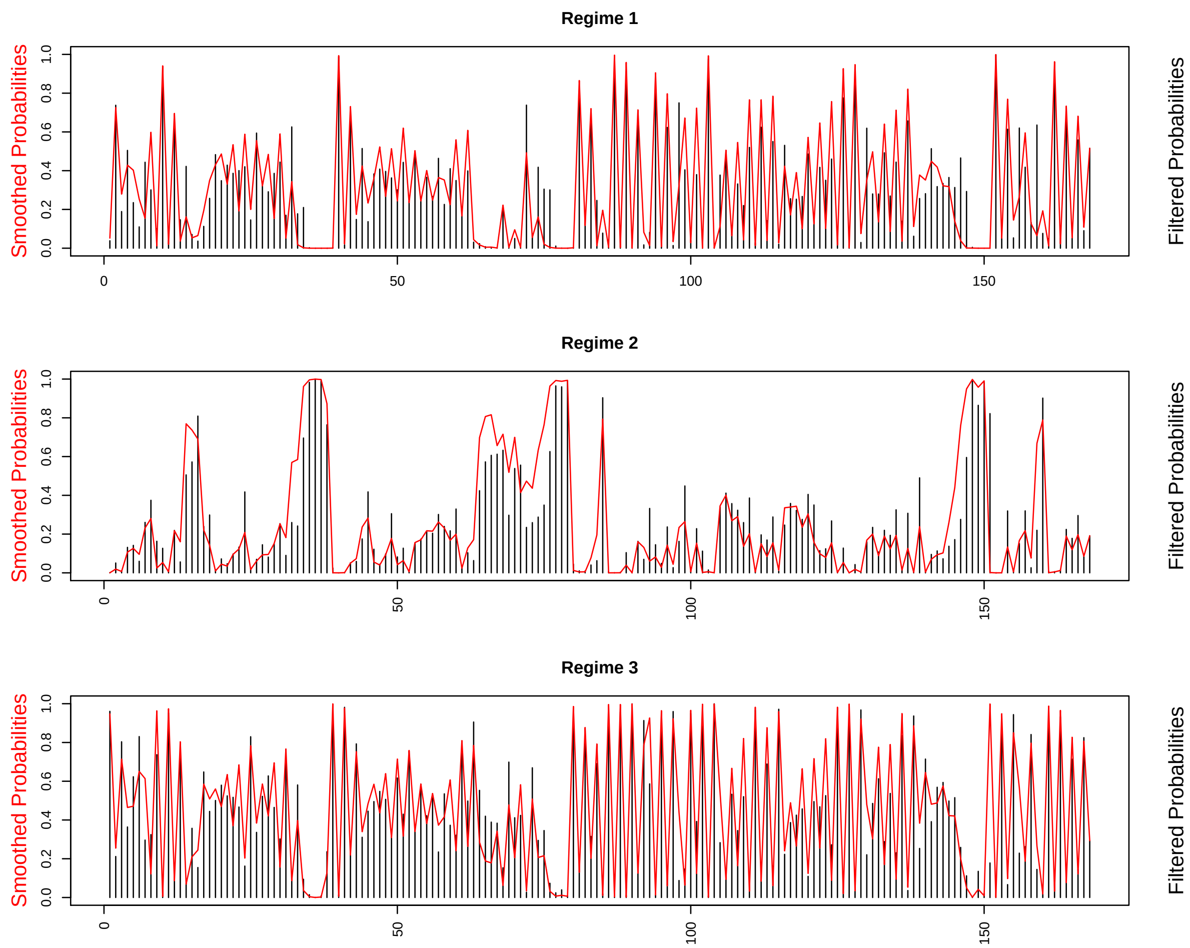
8.3 A 3-regime MSW Model

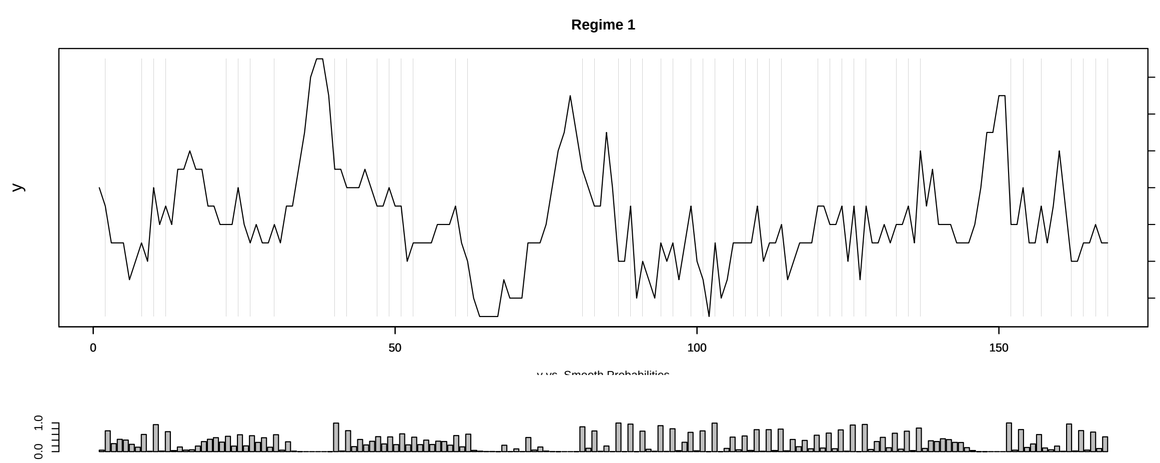
We show the same results as in the previous section:

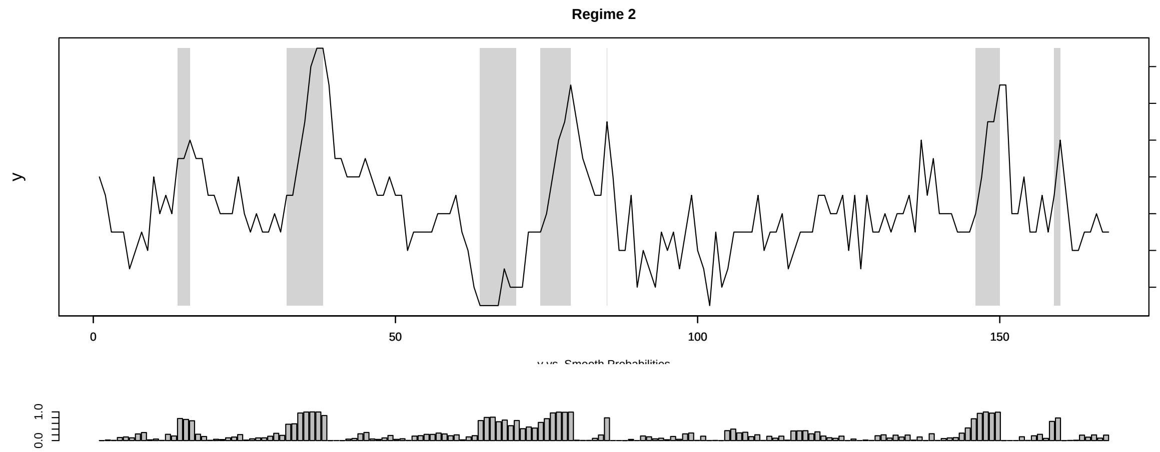


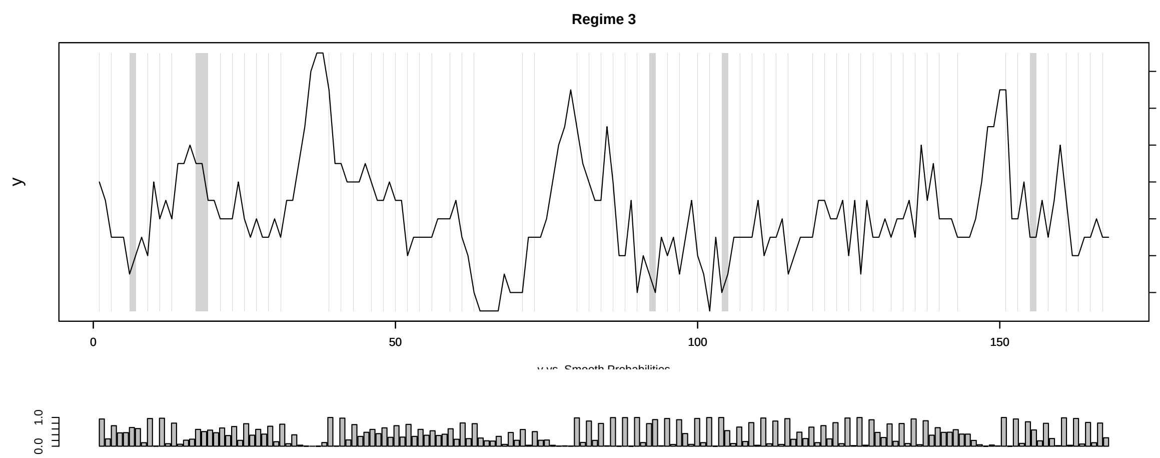
In order to avoid singularity we choose $p = 2$.

```
## transProbMatrix =

##        [,1]   [,2]   [,3]
## [1,] 0.0509 0.0000 0.6579
## [2,] 0.0000 0.6679 0.1862
## [3,] 0.9490 0.3321 0.1559

## ergodicProbs =

##           [,1]
## [1,] 0.3075499
## [2,] 0.2487743
## [3,] 0.4436758

## 3-regime AR coeffs:

##   (Intercept)       y_1         y_2
## 1  0.02837399 0.3503962 -0.05771262
## 2  0.10256305 1.0329795  0.04660607
## 3 -0.07145600 0.5382007  0.34612189

## sErrors:

##   (Intercept)        y_1        y_2
## 1  0.19086563 0.04089129 0.23653172
## 2  0.29064497 0.02100644 0.09344619
## 3  0.08379111 0.19086563 0.04089129
```


Regime 1
Smoothed Probabilities
Filtered Probabilities
Regime 2
Smoothed Probabilities
Filtered Probabilities
Regime 3
Smoothed Probabilities
Filtered Probabilities


Regime 1
y





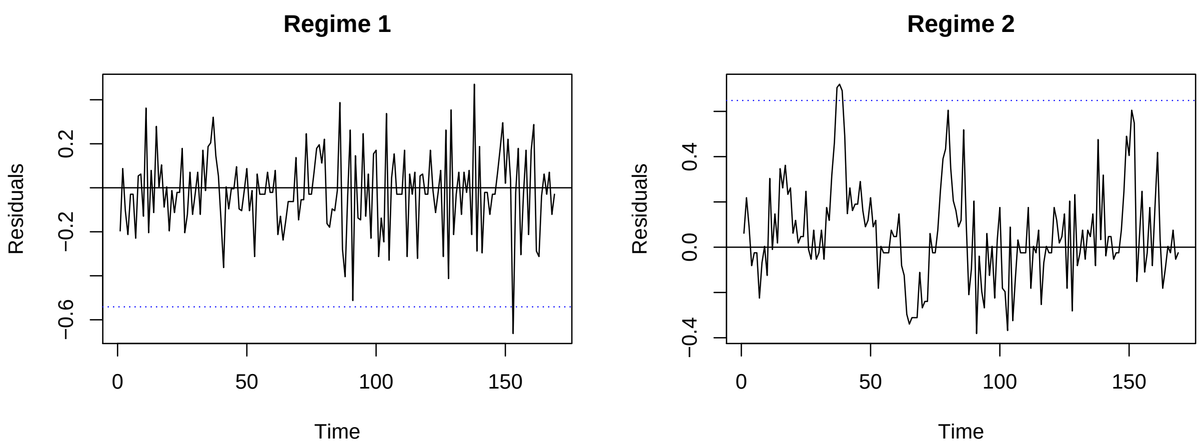
Unlike the previous 2-regime variant, the dominance of each regime in the time series is, according to the filtered probability estimates, quite rare. Further analysis will be carried out in the next section.

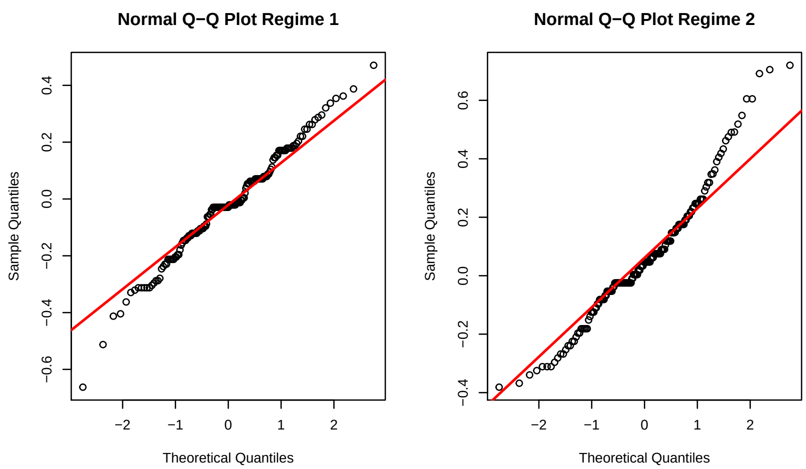
8.4 Diagnostics

The diagnostics of an MSW model rely either on the score function (i.e: gradient of the log-likelihood function: $\nabla_{\theta_{\Phi},\sigma_{\varepsilon}} \log f(X_t|\Omega_{t-1};\theta)$) or take form of a Lagrange multiplier (LM) test.

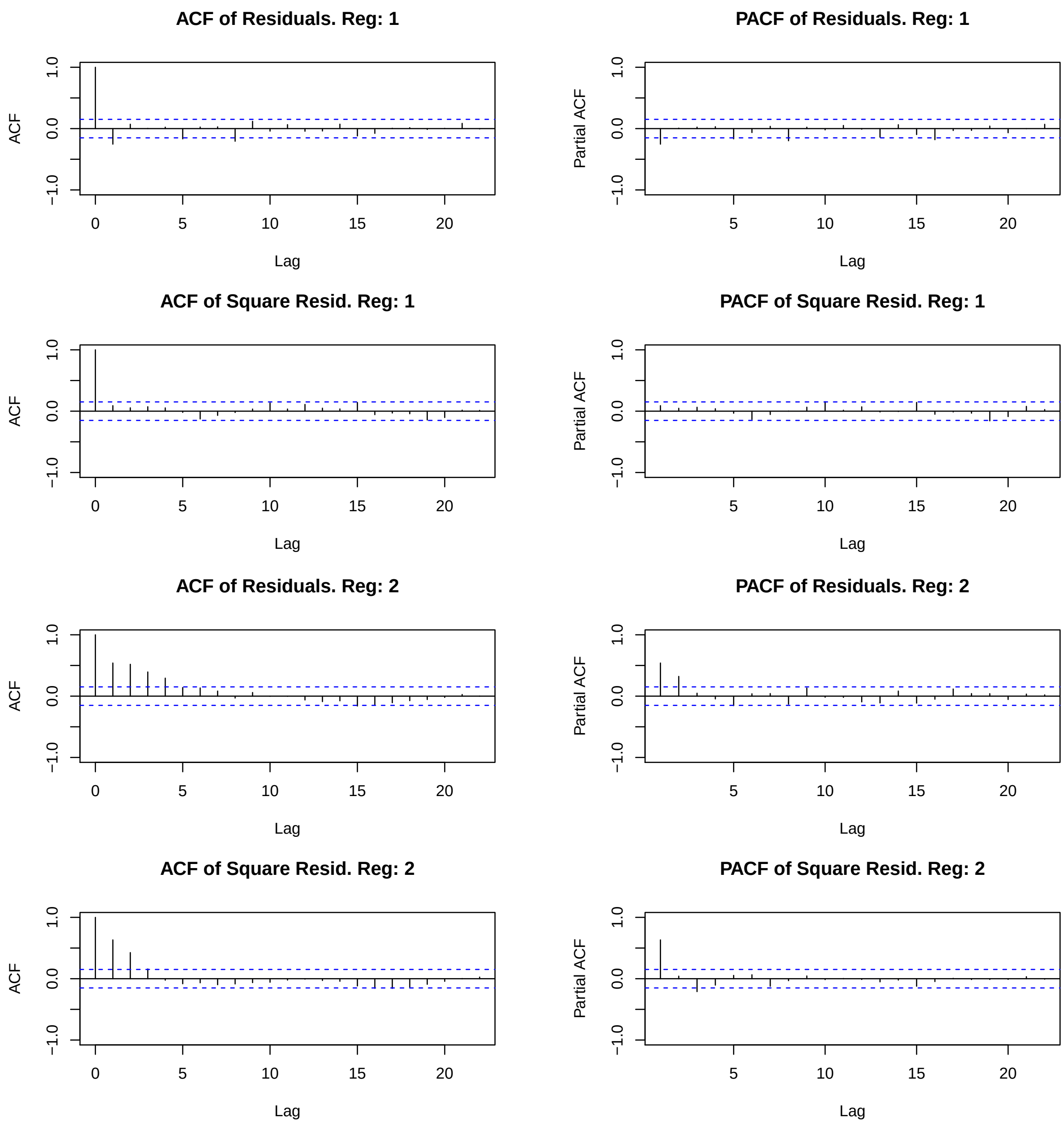
For the sake of simplicity, we chose to analyze the models graphically, by observing their residuals from each regime's fitted values.

2-Regime residuals:

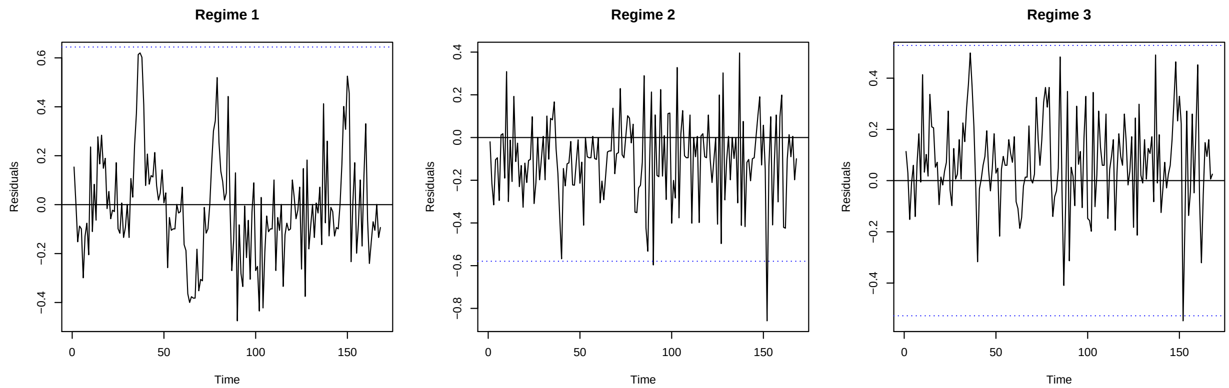




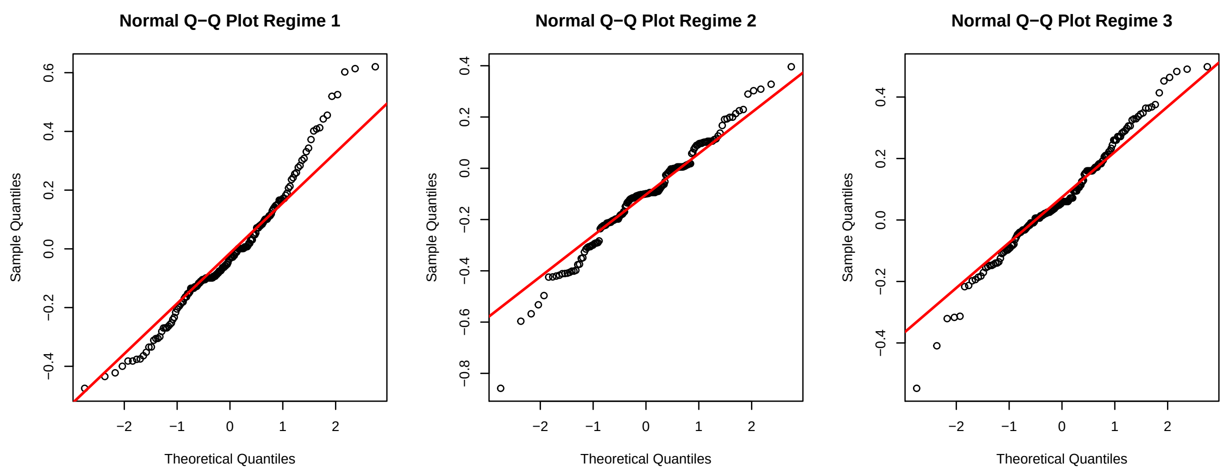
And quantile-quantile plots:
In the first regime the closeness of the data to linearity suggests a uniform distribution of residuals, whereas for the second regime, we observe a slight fluctuation in the quantile plot. This means that the second regime could possibly harbor remaining nonlinear behavior. This can be observed in the residual ACF:



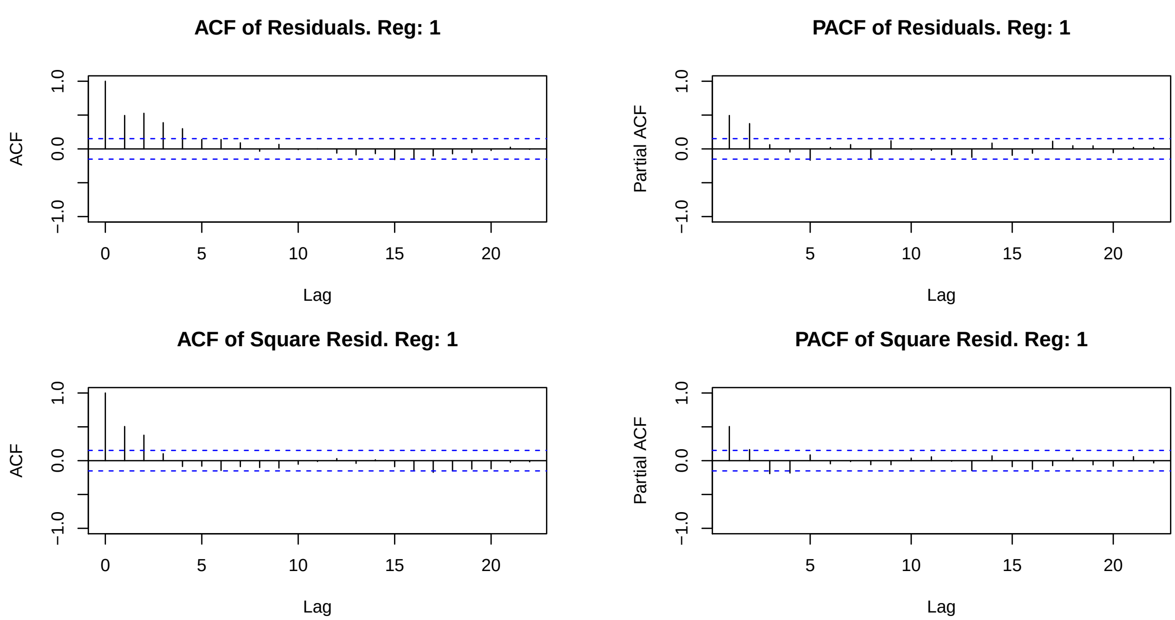
3-regime residuals:

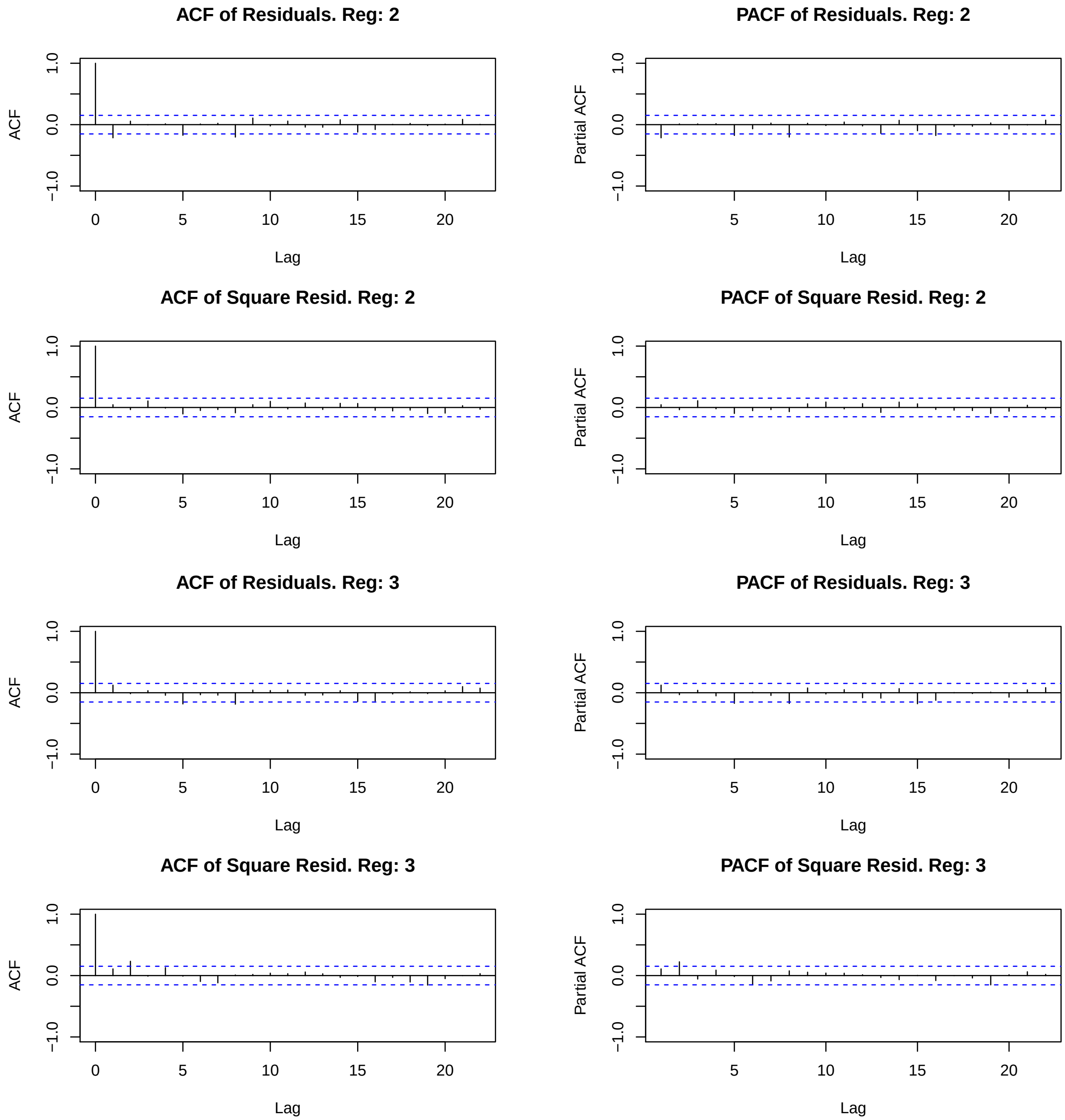


3-regime quantile-quantile plots:



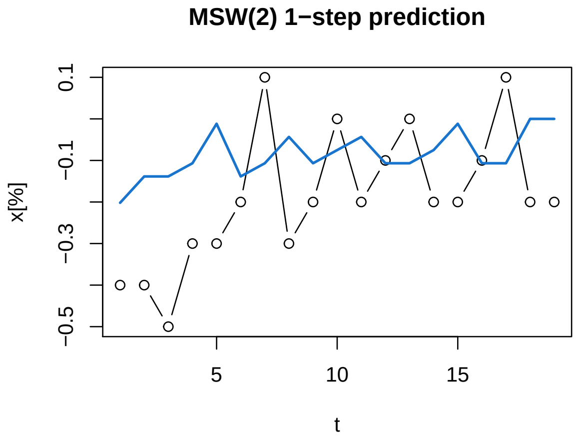
3-regime square resid autocorrelations:

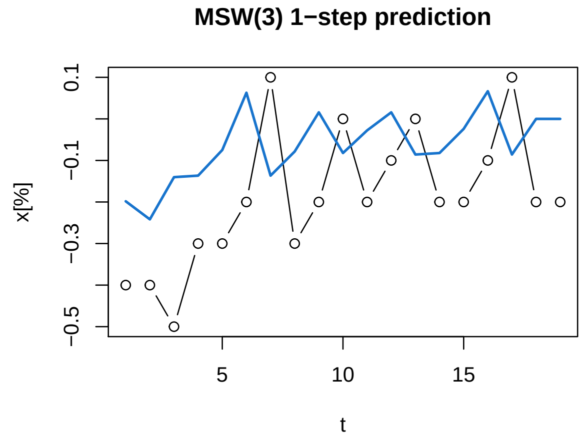




Which suggests data might still contain some nonlinearity not covered by the second regime.

8.5 Predictions





As we can observe, a 3-regime MSW proves to be less efficient in predicting the values of the evaluation part.

8.6 Conclusion

We can conclude that our use of MSW models from the `MSwM` package was purely experimental, mainly due to the fact that they fail to produce reliable 1-step predictions when compared to, for example, STAR models.

$MSW(2,1)$:

$$X_t = \begin{cases} (0.021 \pm 0.0173) + (0.9166 \pm 0.0533)X_{t-1} + \varepsilon_t & \text{if } s_t = 1 \\ (-0.047 \pm 0.0208) + (0.2837 \pm 0.0987)X_{t-1} + \varepsilon_t & \text{if } s_t = 2 \end{cases} \quad \hat{\sigma}^2_{\varepsilon,1} = 0.0328124 \ , \ \hat{\sigma}^2_{\varepsilon,2} = 0.0501344$$

$MSW(2,3)$:

$$X_t = \begin{cases} (0.0284 \pm 0.1909) + (0.3504 \pm 0.0409)X_{t-1} + (-0.0577 \pm 0.2365)X_{t-2} + \varepsilon_t & \text{if } s_t = 1 \\ (0.1026 \pm 0.2906) + (1.033 \pm 0.021)X_{t-1} + (0.0466 \pm 0.0934)X_{t-2} + \varepsilon_t & \text{if } s_t = 2 \\ (-0.0715 \pm 0.0838) + (0.5382 \pm 0.1909)X_{t-1} + (0.3461 \pm 0.0409)X_{t-2} + \varepsilon_t & \text{if } s_t = 3 \end{cases}$$

$\hat{\sigma}^2_{\varepsilon,1} = 0.0460646$, $\hat{\sigma}^2_{\varepsilon,2} = 0.0481927$, $\hat{\sigma}^2_{\varepsilon,3} = 0.0359433$

9 Artificial Neural Networks

Popular in different applications of machine learning, the last type of models we will use are neural networks. Their popularity stems from the fact that they can approximate almost any nonlinear function with arbitrary precision. It is usually the case, that a neural network becomes a black box, trained to perform a certain task, oblivious to any possible internal dynamics that could be described for the data (like in the case of SETAR, STAR, and MSW's).

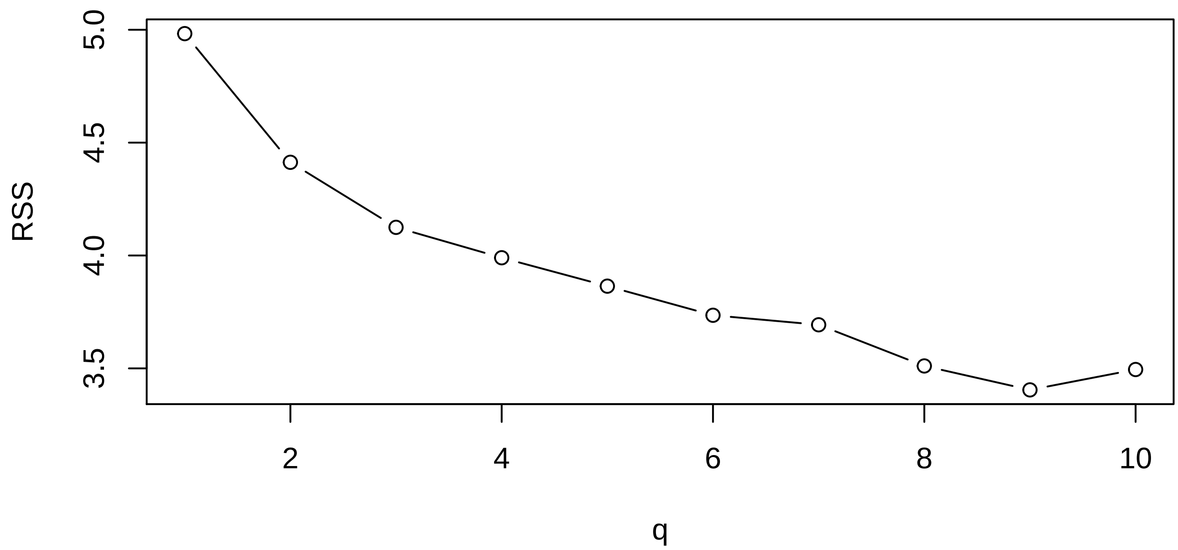
In this chapter, we will construct such models, perform diagnostics and predictions. On top of that we can visualize the nodes of the network.

9.1 Estimation

```
## Series: xt
## Model:  NNAR(2,2)
## Call:   nnetar(y = xt)
##
```

```
## Average of 20 networks, each of which is
## a 2-2-1 network with 9 weights
## options were - linear output units
##
## sigma^2 estimated as 0.02639
```

The automatic detection in the **nnetar** method found the orders $(p, q) =$ `(2,2)`, where q is the number of nodes.

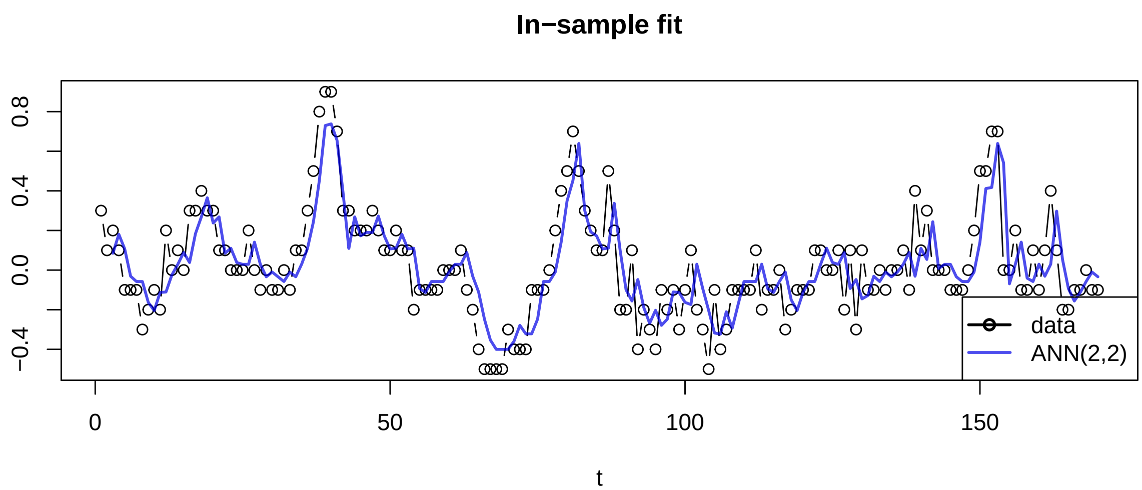


Taking a look at the graph with plotted `RSS` of `nnetar` 's, the steepest drop in the value is observed at $q = 2$ which corresponds to the result of an automatic search.

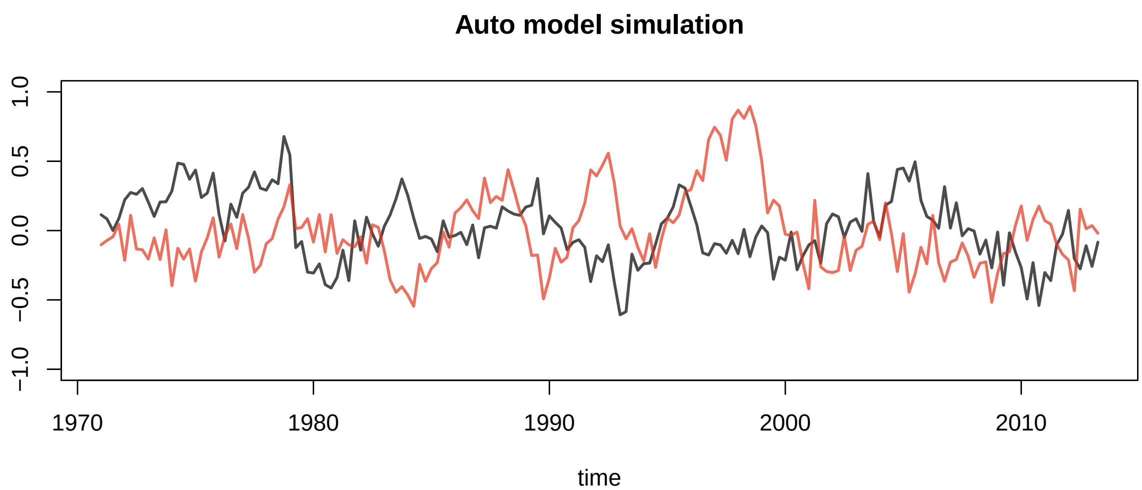
```
## Series: xt
## Model:  NNAR(2,2)
## Call:   nnetar(y = xt, p = 2, size = 2)
##
## Average of 20 networks, each of which is
## a 2-2-1 network with 9 weights
## options were - linear output units
##
## sigma^2 estimated as 0.02601
```

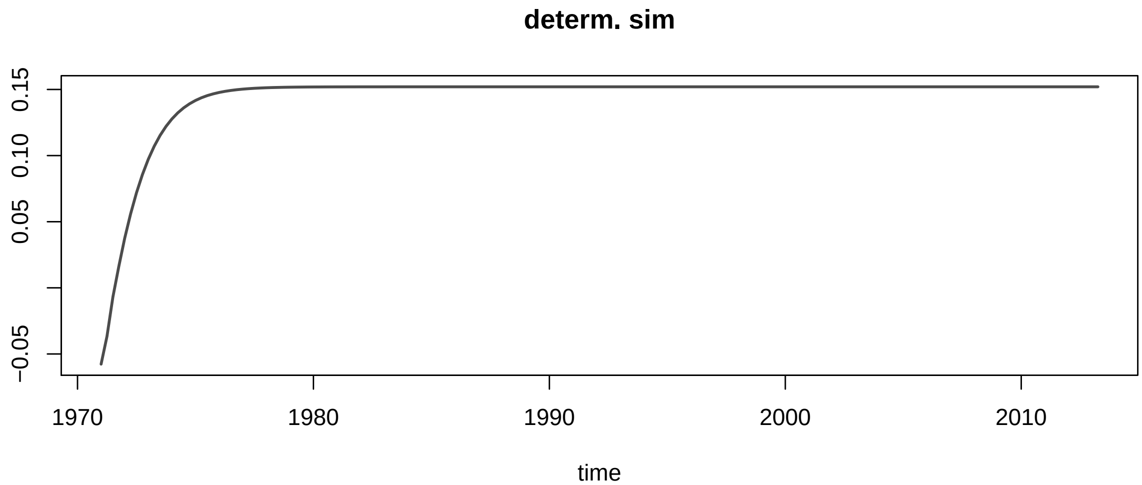
9.2 Diagnostics

First, we plot the test sample fit:



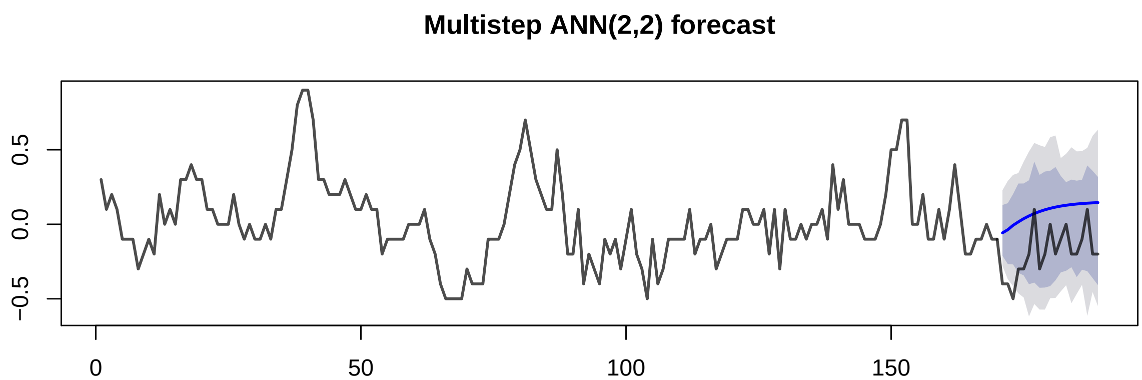
For demonstration purposes, we can plot a stochastic simulation of the model data:

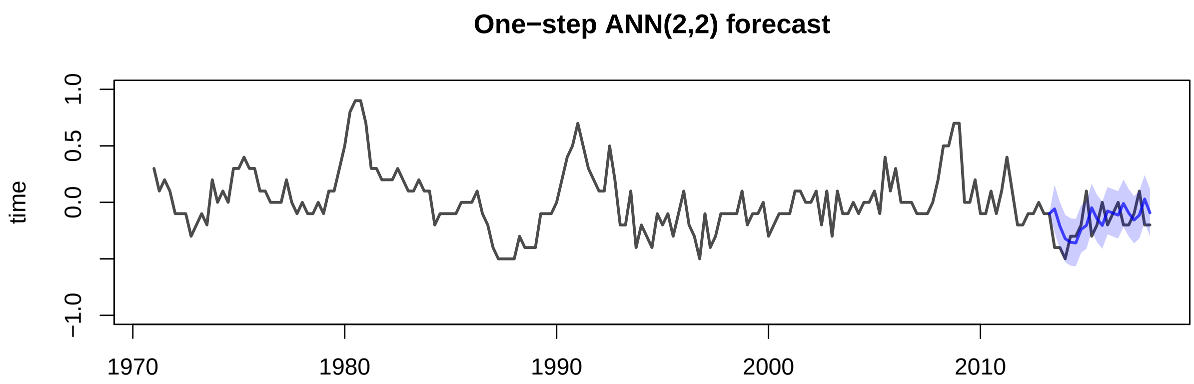




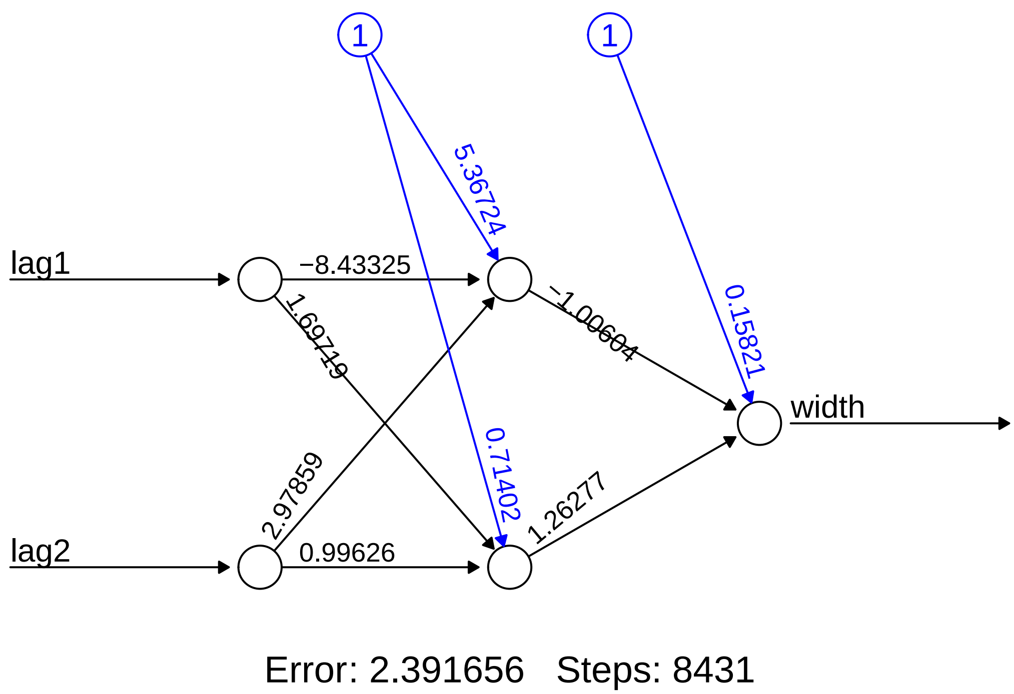
We notice that the deterministic simulation of the chosen model seems to stabilize on an equilibrium of `~0.15`.

9.3 Predictions

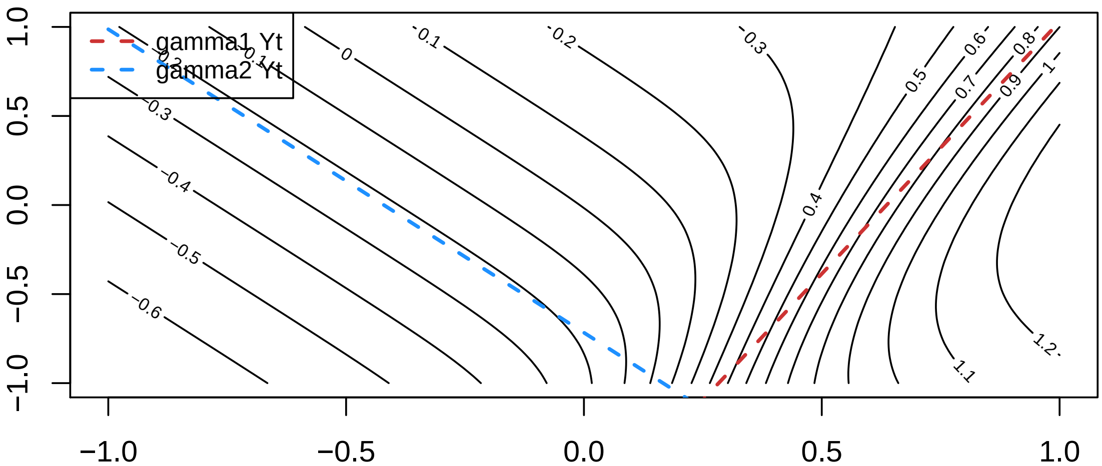




9.4 Visualisation of a Neural Network



Skeleton of a neural network:



The dashed lines on the contour plot above represent the subdivision of the predictor space according to the activations of logistic functions G_1 and G_2, representing $G_j(\gamma_1^\top Y_t) = 1/2$.

9.5 Conclusion

The examined $ANN(2,2)$ model can be formulated as follows:

$$X_t = 0.1582 + -1.006G(\gamma_1^\top Y_t) + 1.2628G(\gamma_2^\top Y_t) + \varepsilon_t$$

$$\gamma_1 = (5.3672, -8.4333, 1.6972)^\top \ , \ \gamma_2 = (5.3672, 2.9786, 0.9963)^\top$$
$$Y_t = (1, X_{t-1}, X_{t-2})^\top \ , \ G(x) = 1/(1 + \mathrm{e}^{-x})$$
$$\hat{\sigma}_\varepsilon^2 = NA$$

10 Evaluation

After covering four different classes of nonlinear models of stochastic processes, we proceed to compare the results by choosing the best candidates from each. There will be two criteria of choice, namely fit quality determined by residual variance σ_ε^2 and 1-step prediction MSE:

```
## Models ordered by sigmaSq:
```

```
##                          model      BIC sigmaSq MSE(1-step)
## 6     SETAR(5,3,0.008,0.403)   -575.9  0.0245      0.0526
## 14                  ANN(2,2)       NA   0.026      0.0306
## 1           SETAR(2,2,0.103) -602.841  0.0262      0.0605
## 2  LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262      0.0316
## 11       LSTAR(2,2,0.131,10) 1261.097  0.0267      0.0307
## 13                  MSW(2,3)       NA  0.0297      0.0388
## 12                  MSW(2,1)       NA  0.0305      0.0366
```

```
## Models ordered by prediction MSE:
```

```
##                        model      BIC sigmaSq MSE(1-step)
## 14                  ANN(2,2)       NA   0.026      0.0306
## 11       LSTAR(2,2,0.131,10) 1261.097  0.0267      0.0307
## 2  LSTAR(2,2,0.192,0.4,10,4) 1290.717  0.0262      0.0316
## 12                  MSW(2,1)       NA  0.0305      0.0366
## 13                  MSW(2,3)       NA  0.0297      0.0388
## 6     SETAR(5,3,0.008,0.403)   -575.9  0.0245      0.0526
## 1           SETAR(2,2,0.103) -602.841  0.0262      0.0605
```

Afterwards we write the mathematical formulas for each chosen model:

$SETAR(2,2,0.103)$:

$$X_t = \begin{cases} (0.4317 \pm 0.0416)X_{t-1} + (0.4283 \pm 0.0504)X_{t-2} + \varepsilon_t & \text{if } X_{t-2} \leq 0.1029 \\ (0.0914 \pm 0.0258) + (1.0803 \pm 0.059)X_{t-1} + (-0.4245 \pm 0.078)X_{t-2} + \varepsilon_t & \text{if } X_{t-2} > 0.1029 \end{cases}$$

$\hat{\sigma}^2_\varepsilon = 0.0262$, $\hat{\sigma}_{\varepsilon,1} = 0.0202$, $\hat{\sigma}_{\varepsilon,2} = 0.0474$

$SETAR(5,3,0.008,0.403)$:

$$X_t = \begin{cases} (0.04 \pm 0.01) + (0.4 \pm 0.05)X_{t-1} + (0.48 \pm 0.06)X_{t-2} + (0.31 \pm 0.08)X_{t-3} + (-0.24 \pm 0.04)X_{t-5} + \varepsilon_t & \text{if } X_{t-3} \leq 0.008 \\ (0.96 \pm 0.06)X_{t-1} + (0.29 \pm 0.07)X_{t-4} + (-0.15 \pm 0.06)X_{t-5} + \varepsilon_t & \text{if } 0.008 < X_{t-3} \\ (1.16 \pm 0.12)X_{t-1} + (-0.56 \pm 0.14)X_{t-2} + (0.68 \pm 0.23)X_{t-3} + (-1.22 \pm 0.3)X_{t-4} + (0.88 \pm 0.3)X_{t-5} + \varepsilon_t & \text{if } X_{t-3} > 0.403 \end{cases}$$

$\hat{\sigma}_{\varepsilon,1} = 0.021$, $\hat{\sigma}_{\varepsilon,2} = 0.024$

$LSTAR(2,2,0.131,10)$:

$$\begin{aligned} X_t = &((-0.0261 \pm 0.0674) + (0.3141 \pm 0.1824)X_{t-1} + (0.3861 \pm 0.1283)X_{t-2})\ [1 - G_L(X_{t-2}, 0.1314, 10)] + \\ &((0.1341 \pm 0.1104) + (1.1097 \pm 0.1799)X_{t-1} + (-0.5167 \pm 0.2961)X_{t-2})\ G_L(X_{t-2}, 0.1314, 10) + \varepsilon_t \end{aligned} \qquad \hat{\sigma}^2_\varepsilon = 0.0267$$

$LSTAR(2,2,0.192,0.4,10,4)$:

$$X_t = \begin{array}{c} ((0.14 \pm 4.26) + (0.53 \pm 6.42)X_{t-1} + (0.52 \pm 0.86)X_{t-2})\ [1 - G_L(X_{t-2}, 0.192, 10)] + \\ ((0.46 \pm 8.75) + (2.91 \pm 35.39)X_{t-1} + (-1.54 \pm 27.25)X_{t-2})\ [G_L(X_{t-2}, 0.192, 10) - G_L(X_{t-2}, 0.4, 4)] + \\ ((-0.58 \pm 8.76) + (0.27 \pm 12.49)X_{t-1} + (0.88 \pm 17.49)X_{t-2})\ G_L(X_{t-2}, 0.4, 4) + \varepsilon_t \end{array}$$

$\hat{\sigma}^2_\varepsilon = 0.0262$

$MSW(2,1)$:

$$X_t = \begin{cases} (0.021 \pm 0.0173) + (0.9166 \pm 0.0533)X_{t-1} + \varepsilon_t & \text{if } s_t = 1 \\ (-0.047 \pm 0.0208) + (0.2837 \pm 0.0987)X_{t-1} + \varepsilon_t & \text{if } s_t = 2 \end{cases} \quad \hat{\sigma}^2_{\varepsilon,1} = 0.0328124 \ , \ \hat{\sigma}^2_{\varepsilon,2} = 0.0501344$$

$MSW(2,3)$:

$$X_t = \begin{cases} (0.03 \pm 0.19) + (0.35 \pm 0.04)X_{t-1} + (-0.06 \pm 0.24)X_{t-2} + \varepsilon_t & \text{if } s_t = 1 \\ (0.1 \pm 0.29) + (1.03 \pm 0.02)X_{t-1} + (0.05 \pm 0.09)X_{t-2} + \varepsilon_t & \text{if } s_t = 2 \\ (-0.07 \pm 0.08) + (0.54 \pm 0.19)X_{t-1} + (0.35 \pm 0.04)X_{t-2} + \varepsilon_t & \text{if } s_t = 3 \end{cases}$$

$\hat{\sigma}^2_{\varepsilon,1} = 0.046$, $\hat{\sigma}^2_{\varepsilon,2} = 0.048$, $\hat{\sigma}^2_{\varepsilon,3} = 0.036$

$ANN(2,2)$:

$$X_t = 0.1582 + -1.006G(\gamma_1^\top Y_t) + 1.2628G(\gamma_2^\top Y_t) + \varepsilon_t$$

$$\gamma_1 = (5.3672, -8.4333, 1.6972)^\top \ , \ \gamma_2 = (5.3672, 2.9786, 0.9963)^\top$$
$$Y_t = (1, X_{t-1}, X_{t-2})^\top \ , \ G(x) = 1/(1+\mathrm{e}^{-x})$$
$$\hat{\sigma}^2_\varepsilon = NA$$