ISYE6501 HW3

Keh-Harng Feng

June 4, 2017

0.1 Preface

This is a reproducible report with most of the codes doing the heavy lifting hidden in the background. **All codes are available to be audited.** If you wish to check the various scripts and code snippets used for the computations, you can download the source code of the report by clicking here. The code used in this report requires a processor capable of running 4 processing threads and a decent amount of RAM (8 GB should be fine) due to the use of parallel computation in Q4. As a general rule of thumb you should NOT run any downloaded R scripts from an untrusted source on your computer without understanding the source code first.

1 Question 1

Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

I worked on a computer model that simulates the transfer of radiation through the atmosphere. One of the factors that control the amount of radiation that reaches the surface is not suprisingly, the amount of radiation that enters the top of the atmosphere. However, the exact amount of energy outputted by the sun is highly dependent on solar activities. Usually the variation is small enough to be ignored. However if one is to be pendantic and data is lacking for a particular period of interest we can use exponential smoothing to produce an estimation.

Data required would be the measured top of the atmosphere solar irradiance in W/m^2 set at a reference distance at a fixed time step size. Note that the Earth's orbit around the Sun is not circular and thus the distance between the Earth and the Sun will change constantly. Nevertheless it is very easy to work backwards to compute the irradiance at a reference distance by knowing the exact spot Earth and the measuring satellite are at currently and assuming isotropic radiation from the Sun.

Judging by this chart there is some sort of seasonality on the time scale of decades. If data is measured monthly it would be best to set α to a small number since the time scale that is really of interest is much larger than the time step.

2 Question 2

Using the 20 years of daily high temperature data for Atlanta (July through October) from Homework 2 Question 5, build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question.) Note: in R, you can use either HoltWinters (simpler to use) or the smooth package's es function (harder to use, but more general). If you use es, the Holt-Winters model uses model="AAM" in the function call (the first and second constants are used "A"dditively, and the third (seasonality) is used "M"ultiplicatively; the documentation doesn't make that clear).

2.1 Data Preparation

The original data consists of recorded daily temperatures from July 1 to October 31 from the year 1996 to 2015. Each year is separated into a different column and each row represents measurement from a particular day in a particular month. This results in 123 measurements for each annual observation. The original data is thus melted into a long format and converted to a time-series with a frequency of 123 measurements per observation. The time series is shown in Figure 1

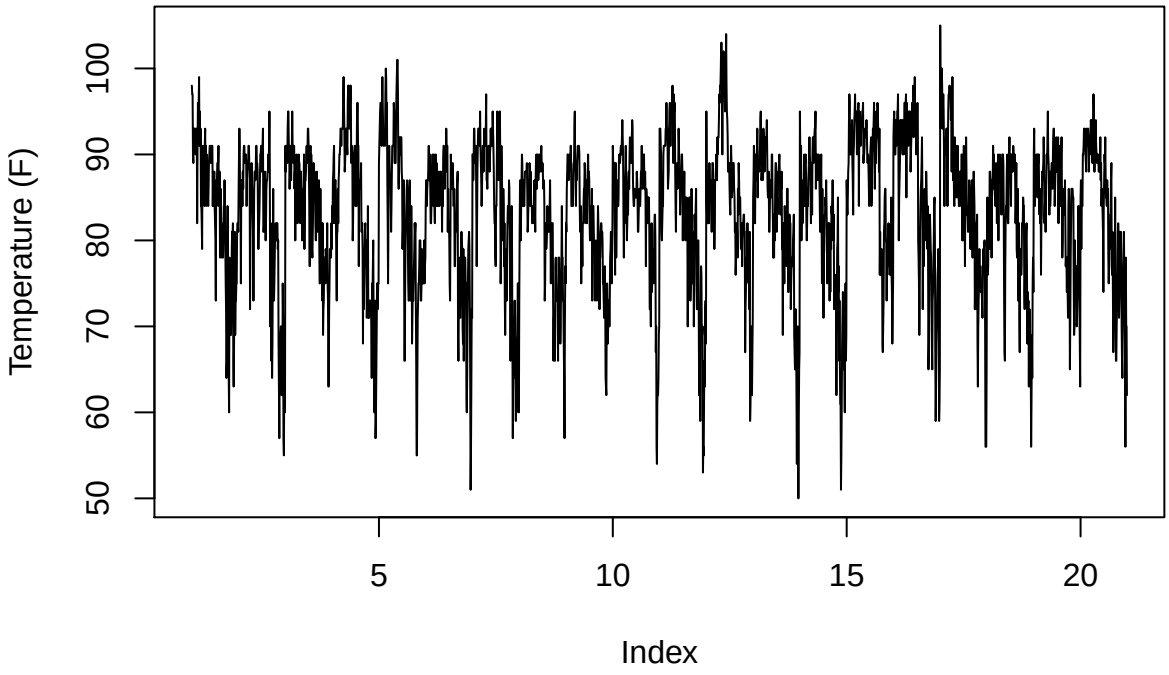

Figure 1: Summer Temperature in Atlanta. Each cycle represents data collected during all the summer months of a year (1996 - 2015).

2.2 Study using Simulated Data

Judging from the wording of the question and common sense, if there is detectable long term warming in Atlanta summertime it should manifest itself in a combination of the following ways:

1. Positive trend (long term increasing temperature).
2. Period inflation (summer periods become longer).

To figure out what kind of effect should be expected a study is carried out using simulated data. The data sets are constructed as the positive halves of a noisy sine wave sampled at 123 points per cycle repeated 20 times with three different combinations of the following additional effects:

1. A linear trend going from 0 to 5 over the entire 20 cycles.
2. A period inflation going from 0% to 100% inflation (ie: double the period) over the 20 cycles.

`es()` from the `smooth` package is used to construct triple exponential smoothing models on the three time series. Although we learned the Additive, Additive, Multiplicative model in class, since the ampltitude of the seasonality is fairly constant the more appropriate Additive, Additive, Additive (AAA) model is used. The resulting models and the corresponding time series can be found in the Appendix.

Figure 9 shows the model components on where there is a positive trend but no seasonality (scroll UP after clicking the reference link). The smoothing parameters are $\alpha = 0.0636$, $\beta = 5.8834 \times 10^{-4}$, $\gamma = 0.0723$. Interestingly, the AAA model failed to decompose the linear trend into the trend component. Instead it is reflected in the time series level (the moving average).

Figure 11 shows the component plot for a series with both trend and season. Once again trend is picked up by the moving average level. Seasonality is correctly decomposed however. The smoothing parameters are $\alpha = 0.2197$, $\beta = 0.0034$, $\gamma = 0$. With $\gamma = 0$ the seasonality component is completely stationary and each of its cycle is completely identical to all other cycles.

Finally, Figure 13 shows what happens when the time series exhibits a trend, a seasonality and a period inflation. The smoothing parameters are $\alpha = 0.2197$, $\beta = 0.0034$, $\gamma = 0$. Notice the "bumps" in level and trend. Since model seaonality has a fixed period by definition and the actual data does not, the model has compensated by setting up a seasonality with the average period of the time series. The bumps are essentially compensations to *push down* or *pull up* the model values to match up with the data values as they move in and out of sync. This is supported by the fact that the bumps become smaller towards the center of the series, where the model seasonality is in sync with the actual data.

If the data indeed contains some form of period inflation and/or trend, it seems that the level component will provide the most direct visual cues. The actual temperature data is now ready to be analyzed.

2.3 Data Analysis

Since there isn't any discernible ampltitude change in Figure 1, an additive model is chosen for the seasonality component. An AAA triple smoothing model is constructed on the actual temperature data. The smoothing parameters are $\alpha = 0.603$, $\beta = 0.0012$, $\gamma = 0.0698$. Figure 2 shows the time series components.

Interestingly there is a bit of negative trend in the beginning. However it quickly goes back to roughly 0 and does not exhibit the characteristic "bumps" as in the case with simulated period inflation data. The elephant in the room, the level component of the time series looks quite noisy and perhaps can be argued to exhibit some sort of periodicity. However its period does not seem to fixed nor does it correspond to the period of the seasonality. It also lacks a relatively straight center as expected if the seasonality is indeed using the average period of a time series an inflating period. Comparing to the effects shown using simulated data, the Atlanta summer periods have not become longer and there is no clear long term increase in Atlanta summer temperatures, either.

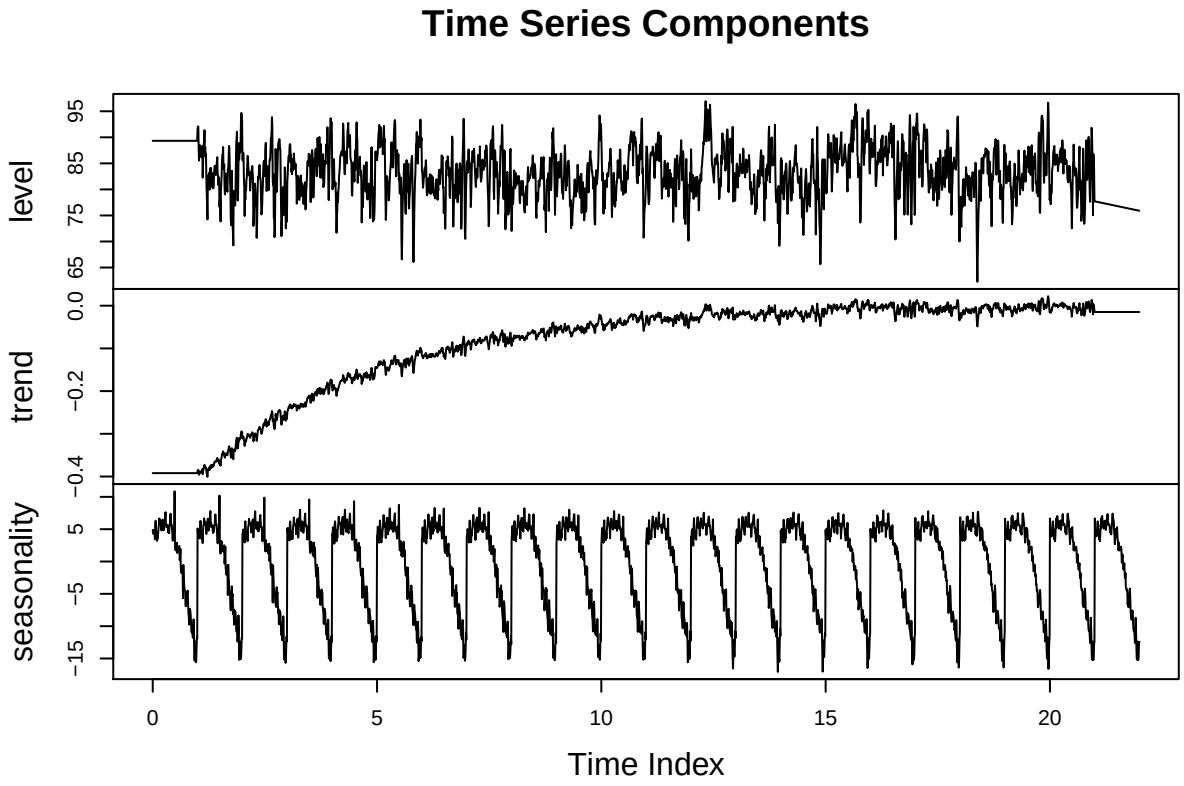

Figure 2: Time series components of Atlanta summer temperatures.

3 Question 3

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

Recently I completed a project that involves analyzing the effects of different factors on automobile fuel efficiency. The data includes measured fuel effciency of different vehicles in the forms of miles per gallon and various physical characteristics of the cars. The main question was if the types of transmission (auto or manual) had an effect on fuel effciency. My strategy to answer that question was to build a multivariate linear regression model to fit MPG on pertinent predictors and see if transmission type was a significant predictor in the model. The following predictors were chosen after feature selection:

`Transmission Type`

`Number of Engine Cylinders`

`Weight`

`Horsepower`

Hypothesis test carried out on model coefficients showed that transmission is NOT an important factor when it comes to fuel efficiency. The full report can be accessed here.

4 Question 4

Using crime data from http://www.statsci.org/data/general/uscrime.txt (description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with [the data shown on handout].

Show your model (factors used and their coefficients), the software output, and the quality of fit.

4.1 Data Preprocess & Exploratory Analysis

The data is loaded with header. The first 5 rows are shown below:

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq   Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.0846
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.0296
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.0834
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.0158
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.0414
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.0342
##   Time Crime
## 1 26.2   791
## 2 25.3  1635
## 3 24.3   578
## 4 29.9  1969
## 5 21.3  1234
## 6 21.0   682
```

The 16th column `Crime` is the response. The second column `So` is a categorical variable with two levels `0` and `1`. A quick check shows that no predictors exhibit near zero variance:

```
nzv <- nearZeroVar(data.predictors, saveMetrics = TRUE)
nzv
```

```
##         freqRatio percentUnique zeroVar   nzv
## M           1.000        65.957   FALSE FALSE
## So          1.938         4.255   FALSE FALSE
## Ed          1.000        51.064   FALSE FALSE
## Po1         1.000        80.851   FALSE FALSE
## Po2         2.000        82.979   FALSE FALSE
## LF          1.000        85.106   FALSE FALSE
## M.F         1.000        76.596   FALSE FALSE
## Pop         1.000        74.468   FALSE FALSE
## NW          1.500        93.617   FALSE FALSE
## U1          1.333        74.468   FALSE FALSE
## U2          1.667        55.319   FALSE FALSE
## Wealth      2.000        97.872   FALSE FALSE
## Ineq        1.500        89.362   FALSE FALSE
## Prob        1.000       100.000   FALSE FALSE
## Time        1.000        95.745   FALSE FALSE
```

A plot of the correlations between all *non-categorical* predictors is shown in Figure 3.

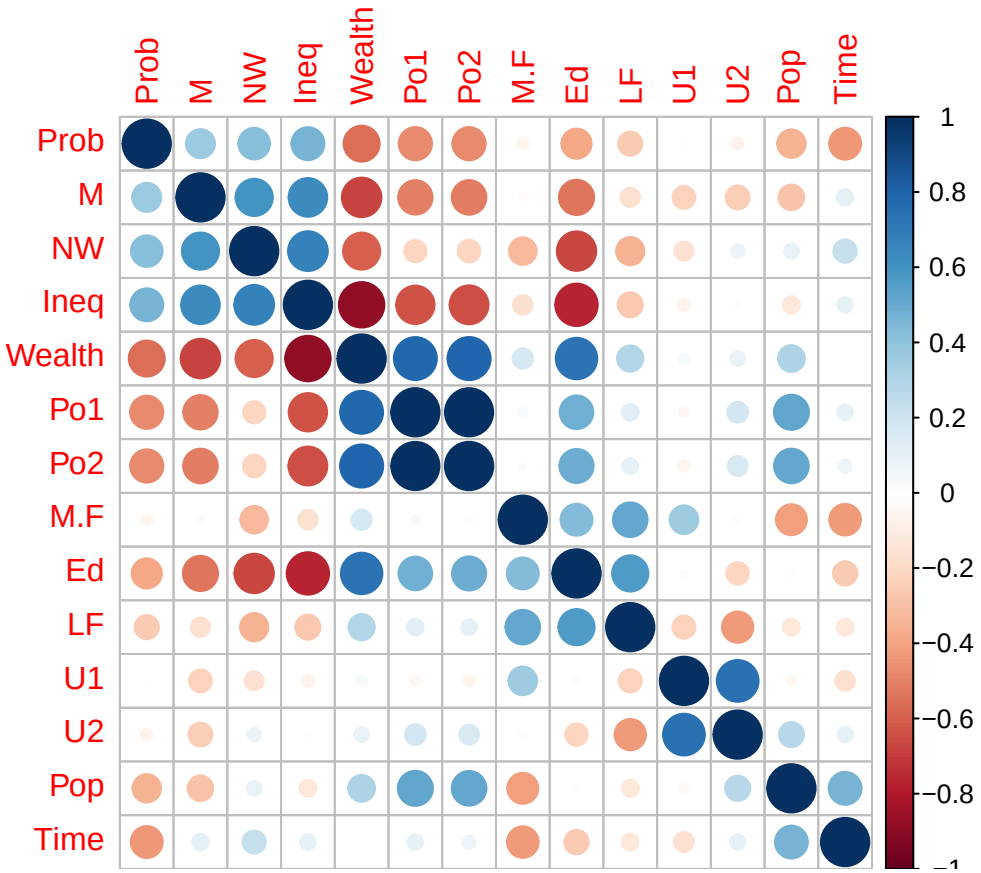

Figure 3: Cluster plot of the correlation matrix. Matrix is symmetric by design.

Strong correlations can be seen between (`Po1`, `Po2`), (`Ineq`, `Wealth`) and (`Ineq`, `Ed`). The following algorithm described by Kuhn and Johnson in Applied Predictive Modeling in the form of the `findCorrelations()` function from the `caret` package is used to identify highly colinear predictors to be removed:

1. Calculate the correlation matrix.
2. Determine the two predictors associated with the largest absolute parwise correlations, (A, B).
3. Determine the average correlation between A and other predictors. Do the same for B.
4. Remove the predictor with larger average correlation.
5. Repeat 2 - 4 until no correlations are above the threshold.

For this step the threshold value is set to the default value, 0.9. The algorithm found the following predictor(s) to be removed due to colinearity:

```
## [1] "Po1"
```

Since the question does not require us to interpret the model based on model coefficients, data transformation, scaling and centering are viable options. A list of density plots are shown in Figure 4

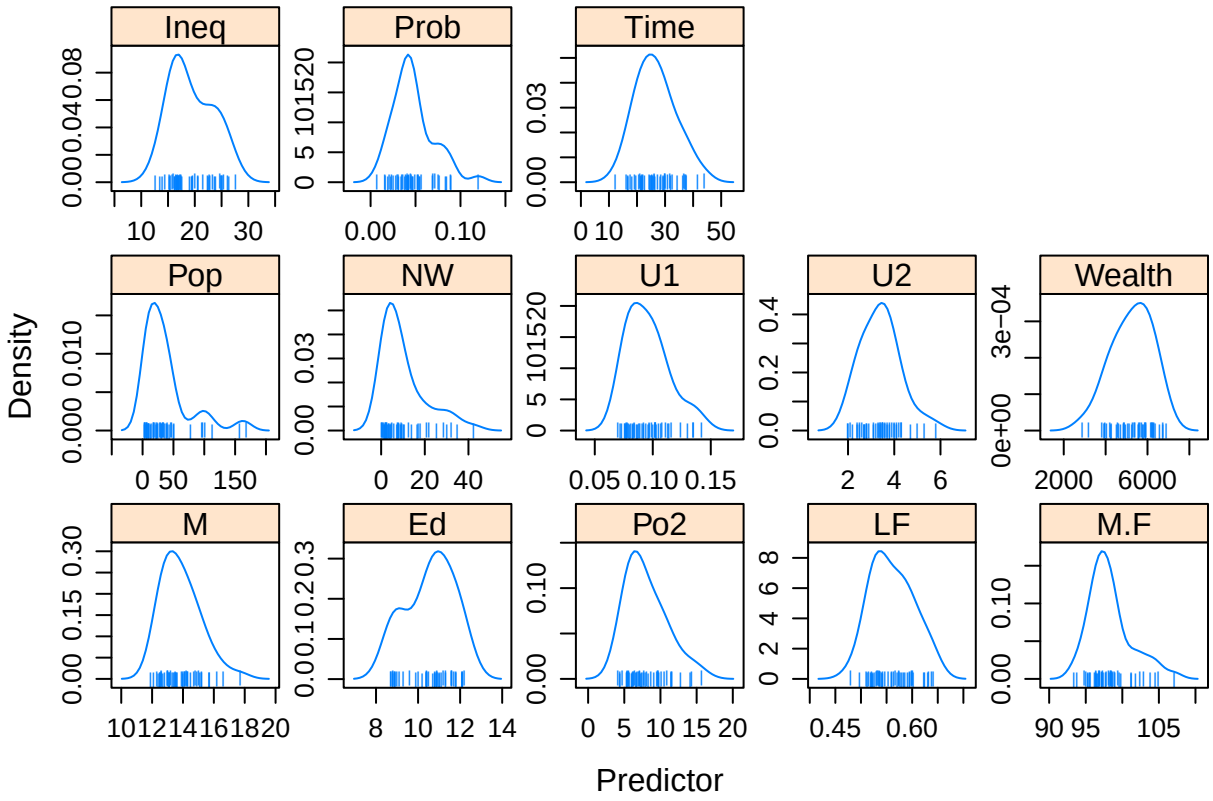

Figure 4: Density plots of untransformed predictors.

Some of the predictors exihibit right skew, in particular `Prob`, `Pop`, `NW`, `U1`, `U2`, `M`, `Po2` and `M.F`. `Wealth` exhibits left skew. To improve predictor symmetry they are transformed using Box-Cox transformation then centered and scaled[1]. The result of the transformation is shown in Figure 5.

With the exception of `M.F` the improvement to the predictor distribution is quite remarkable. This set of transformations shall be used prior to automatic feature selection.

[1]The order of transformation is correlation removal -> Box-Cox -> center/scale (either one can go first). The key here is that Box-Cox **must** be carried out before centering and scaling because it involves raising the predictors to different (and possibly negative) powers. Negative numbers can cause a bimodal dispersion due to the sign of the number. If the power is negative and you center and scale first the predictor will contain negative values and you may possibly divide by 0! If the predictor already has 0 or negative values you can apply a positive translation before Box-Cox.

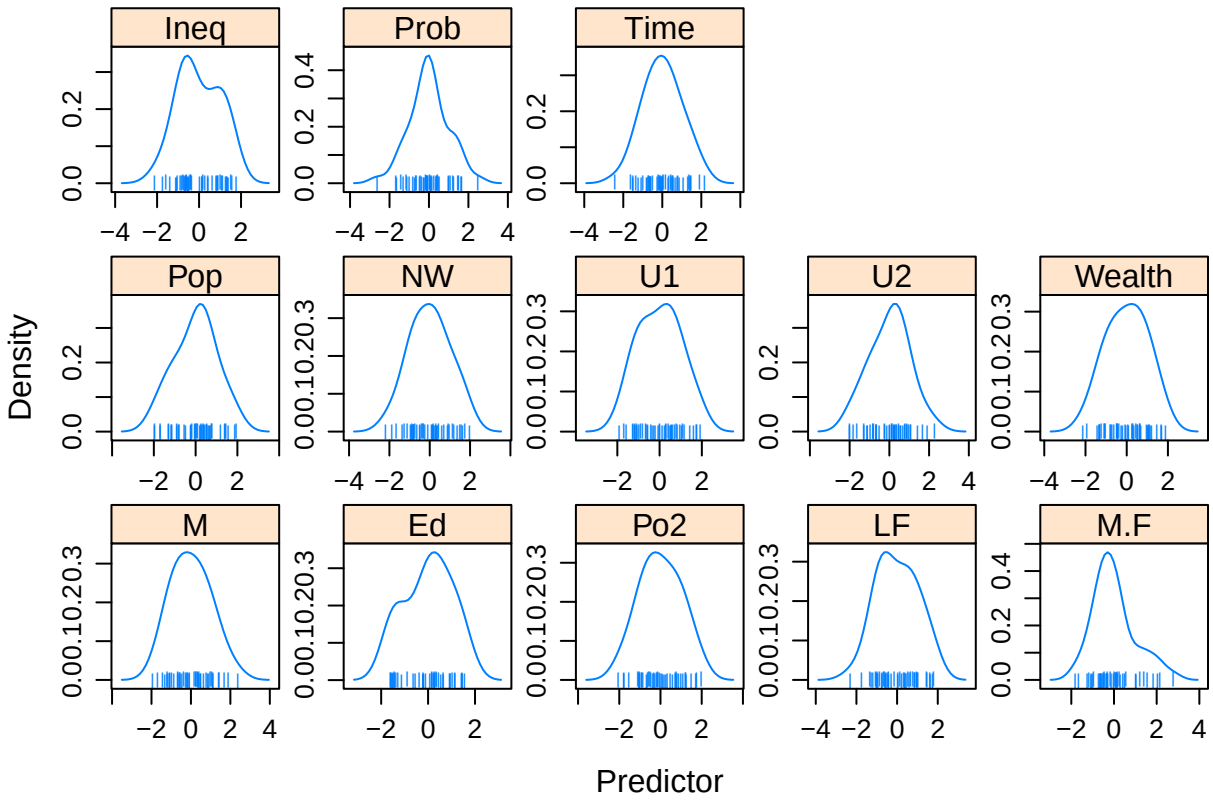

Figure 5: Transformed predictor density plots. Pre-processing applied: high colinearity removal, Box-Cox transformation, centering and normalization.

4.2 Automatic Feature Selection

The preprocessed data is now split into a test (~10%) & training set (~90%). Feature selection is done with a rather stupid approach: an exhaustive search over the entire combination space is carried out. With 14 predictors that's 1.6382×10^4 possible combinations (the trivial combination of no predictors selected is not included). Parallel processing with four PSOCK clusters is utilized to carry out 10-fold cross-validation on the training set. The average MSE from the model prediction on the validation folds is used to estimate out-of-sample performance. The predictor combination that results in the lowest out-of-sample MSE estimate is selected. The final selected combination is M, Ed, Po2, U2, Ineq, Prob.

4.3 Performance Evaluation

A MLR model is retrained using the selected predictor combination using the entire training set. Model statistics are shown below:

```
##
## Call:
## lm(formula = formula, data = data.training)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -426.8 -114.0   13.4  103.4  489.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    913.0       30.9   29.52  < 2e-16 ***
## M              114.9       42.8    2.68  0.01097 *
## Ed             185.7       47.4    3.92  0.00039 ***
## Po2            356.9       47.7    7.49  7.6e-09 ***
## U2              83.7       35.7    2.34  0.02470 *
## Ineq           291.1       54.8    5.32  5.7e-06 ***
## Prob           -92.6       38.1   -2.43  0.02018 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202 on 36 degrees of freedom
## Multiple R-squared:  0.782,  Adjusted R-squared:  0.745
## F-statistic: 21.5 on 6 and 36 DF,  p-value: 1.5e-10
```

All selected predictors display excellent P values, as expected from exhaustive feature selection. From the adjusted R^2 value the model explains about 74.5436% of the variance in training set response. Residual plots are shown in Figure 6. It can be seen that the residuals spread around the mean at 0 and exhibit a normal distribution and constant variance. The residuals furthest away from 0 also have low leverage, therefore there are no significant effects from potential outliers. Qualitatively, there does not seem to be any problems.

The performance of the model on the test set is visualized in Figure 7

The MSE of the final model on the test set is 1.3309×10^4. Compared to the MSE achieved on the training set, 3.4115×10^4, it is surprisingly better! This could be caused by how small the sample is and me getting lucky with the test set chosen. In general, out-of-sample performance is almost always worse than in-sample performance.

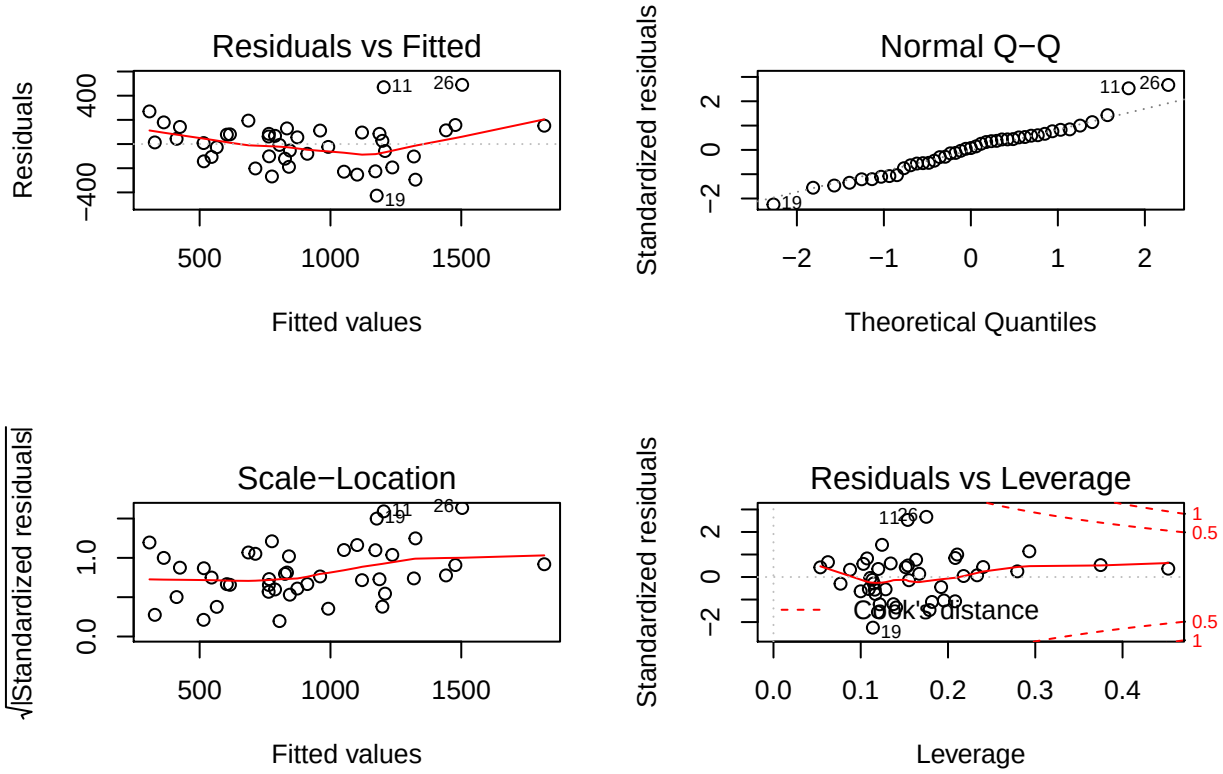

Figure 6: Residual plots of final model.

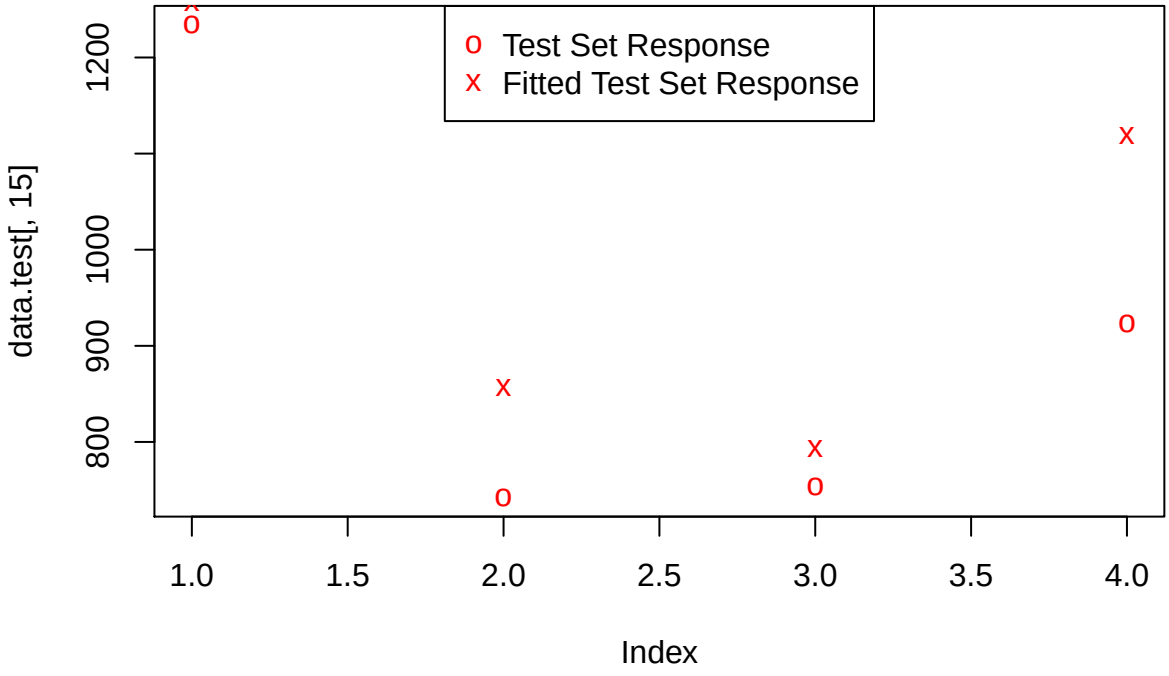

Figure 7: Labeled Response Values vs Model Fitted Values (Training + Test)

4.4 Prediction

A new data point is set at

```
##    M So Ed Po1  Po2   LF M.F Pop  NW   U1  U2 Wealth Ineq Prob Time
## 1 14  0 10  12 15.5 0.64  94 150 1.1 0.12 3.6   3200 20.1 0.04   39
```

With these predictor values the model predicts a crime rate of 1666.3567 with a prediction interval of 1205.7456 to 2126.9678 (95% confidence).

Digression: I used this question as a chance to do a bit of feature selection and get a sense on how to pre-process data effciently in R. It should be noted that exhaustive search with CV on the training set can cause overfitting on the data (although if you didn't do feature selection, you **definitely** overfitted). Ideally feature selection should be performed on a different dataset than the one used for training the model. However the given sample size is simply too small for this to be feasible. Regularization methods such as lasso regression may be a better choice in selecting the optimal features.

5 Appendix

5.1 Simulated Time Series

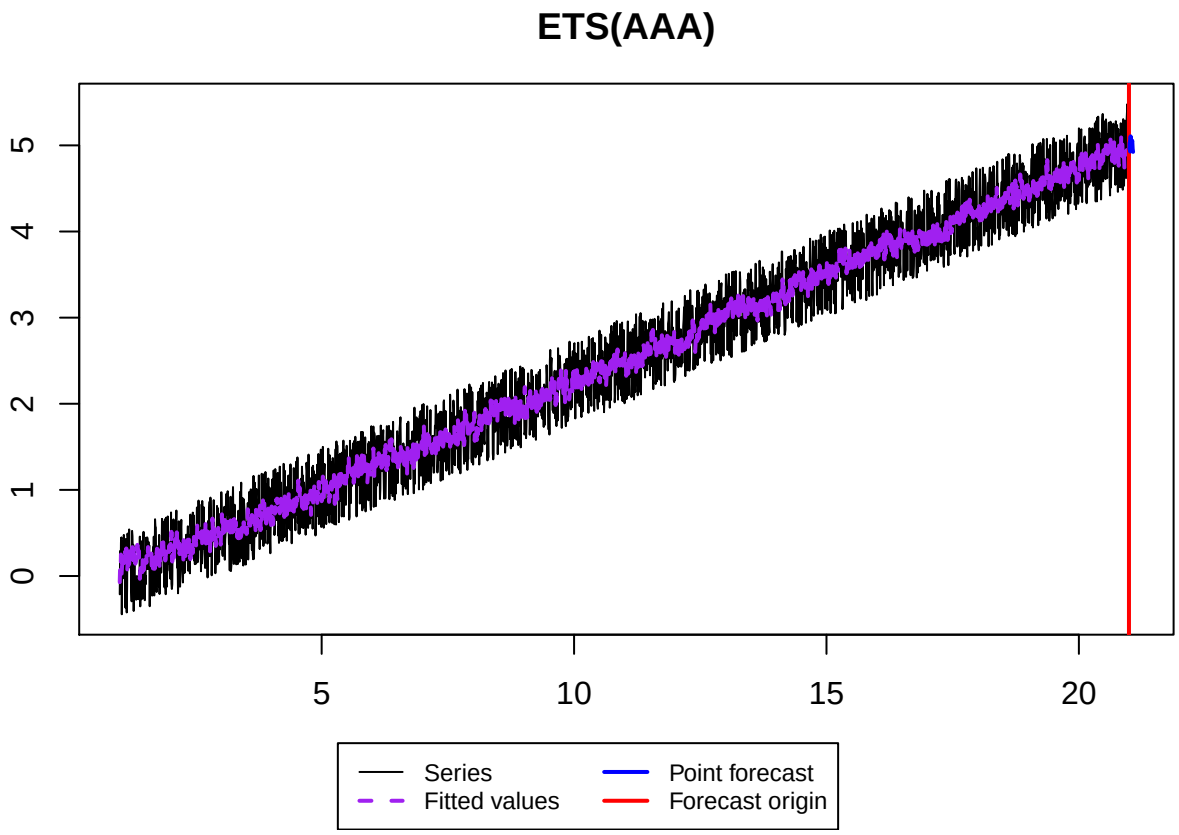

Figure 8: Trend Only

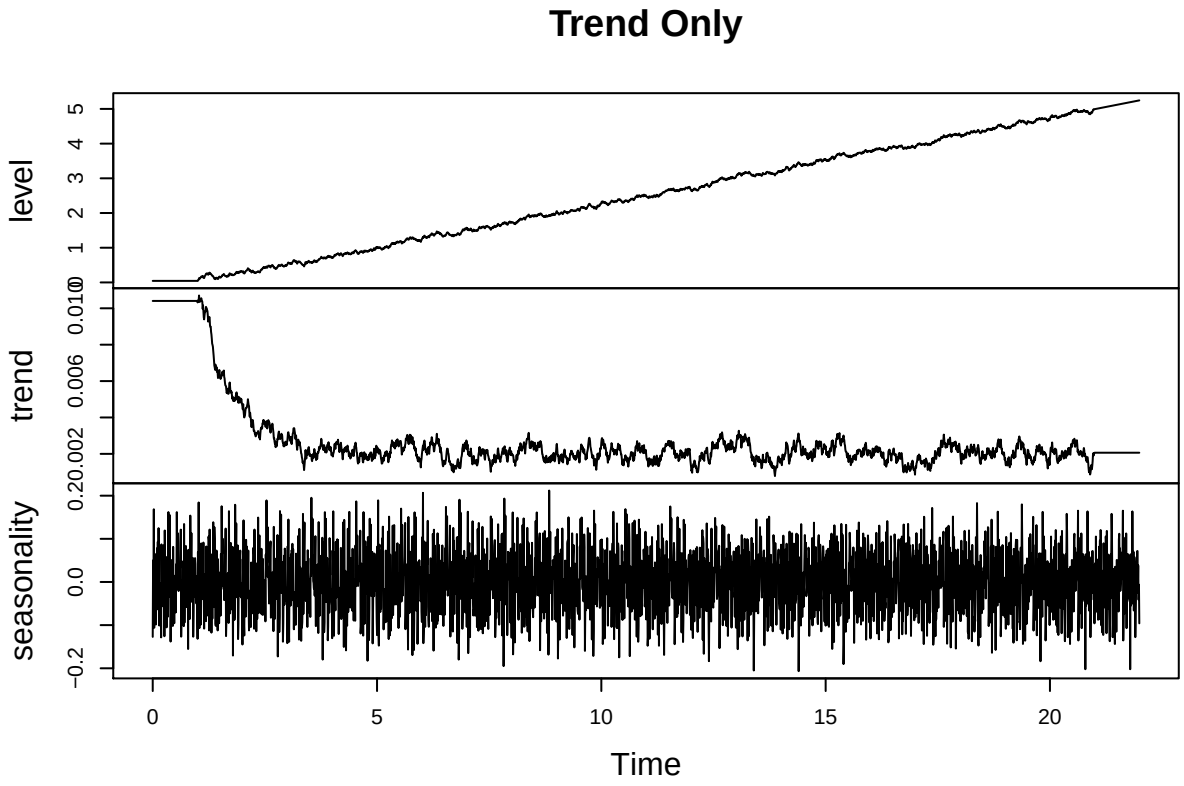

Figure 9: Time series components with only trend (no seasonality).

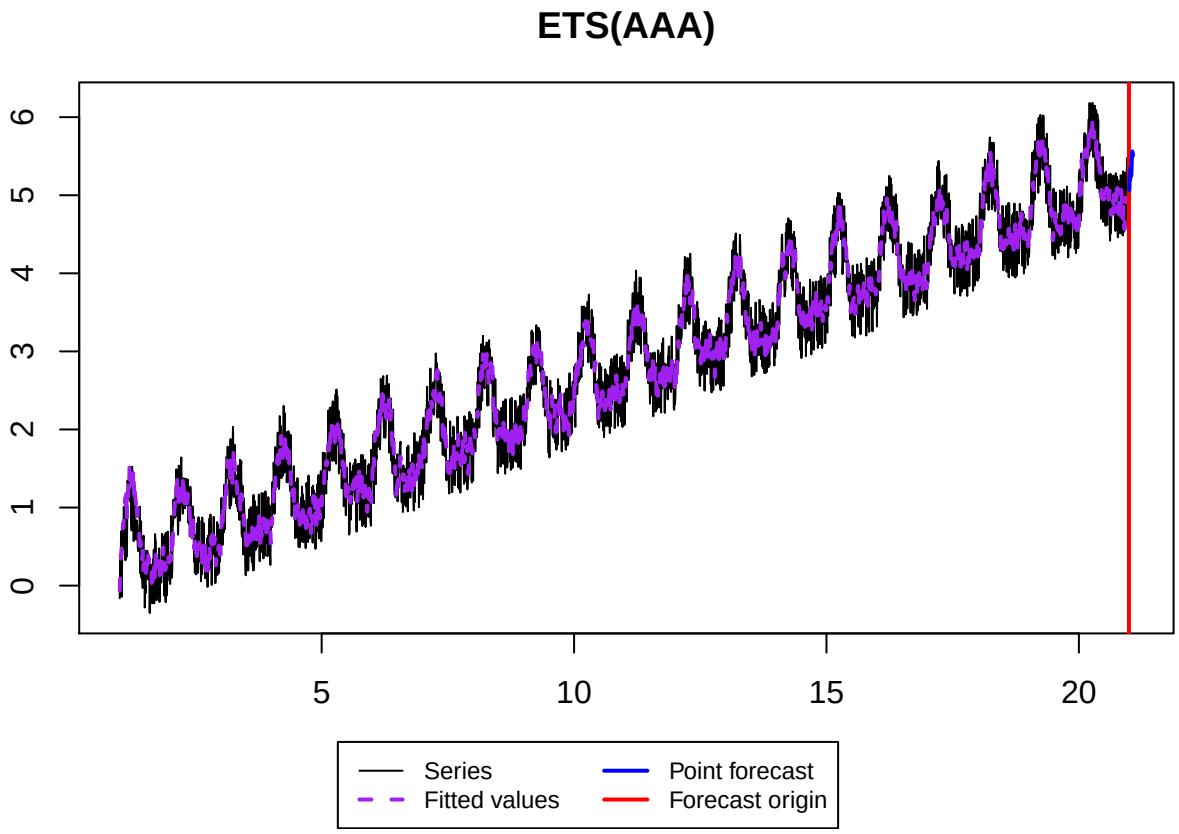

Figure 10: Trend + Season

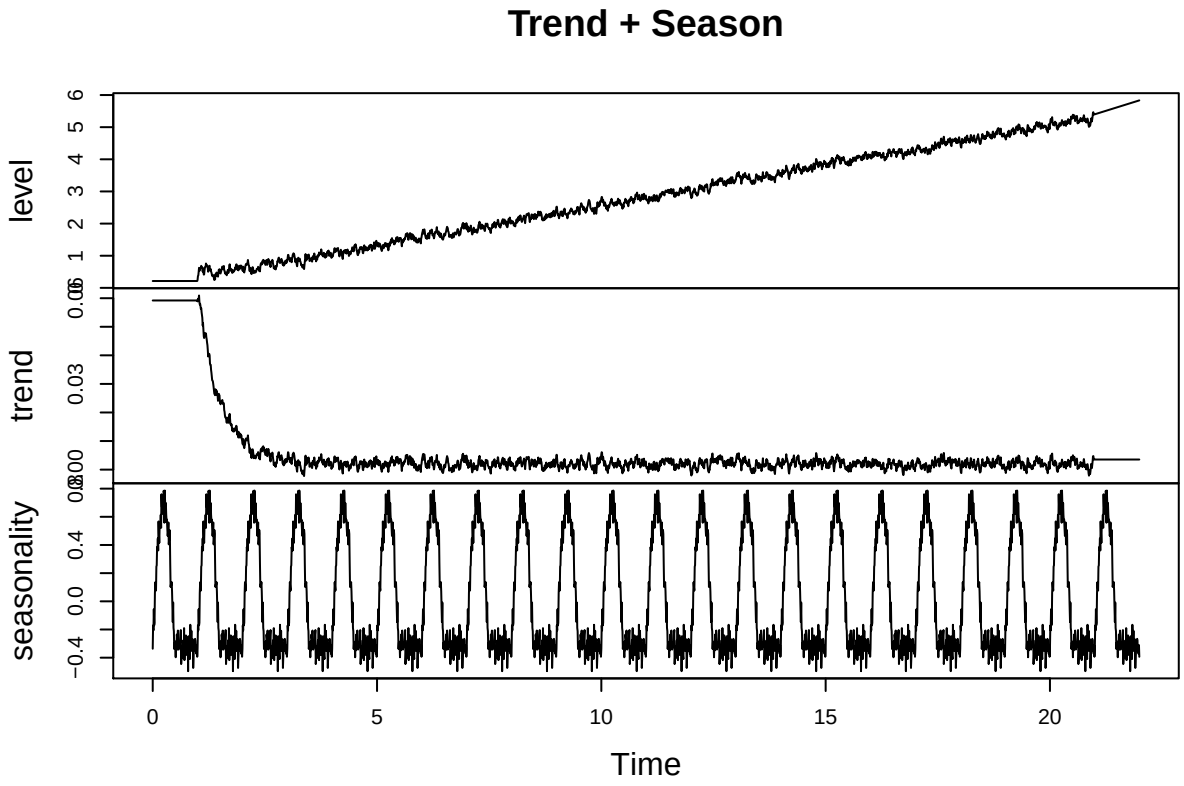

Figure 11: Time series components with trend and season (no period inflation

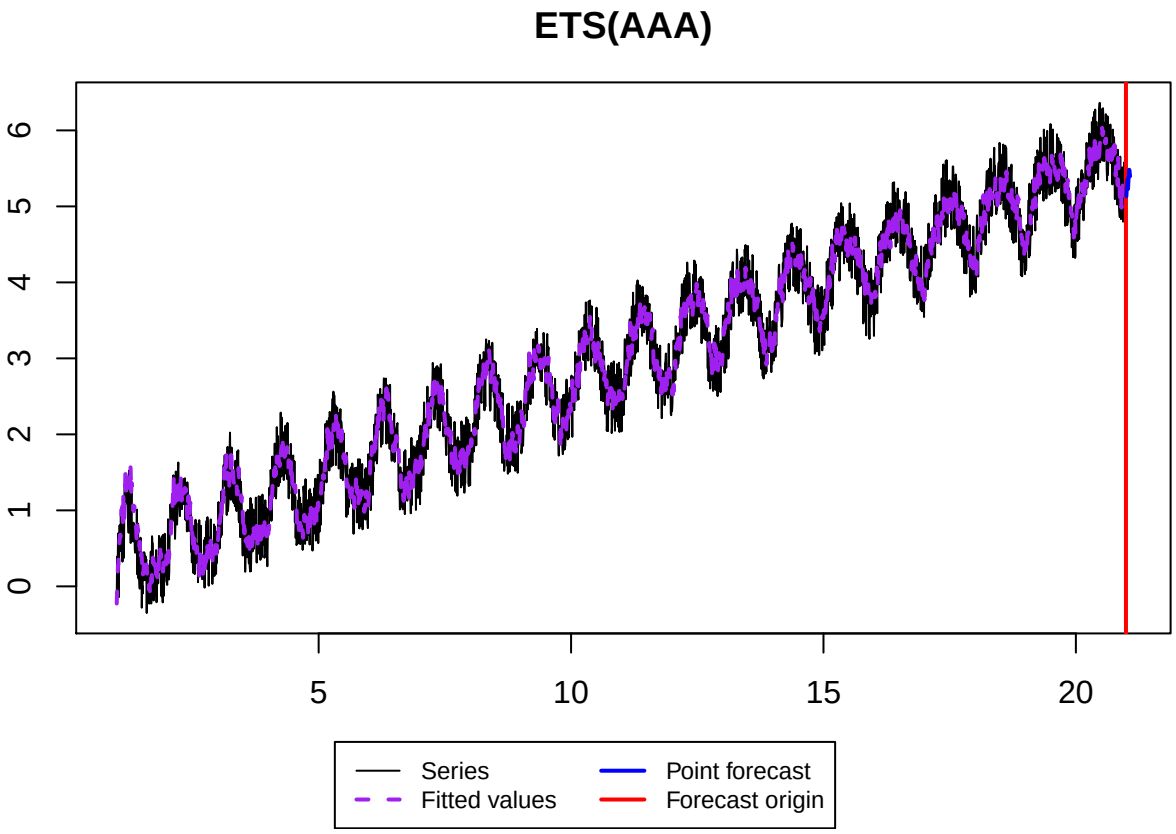

Figure 12: Trend + Season + Period Inflation

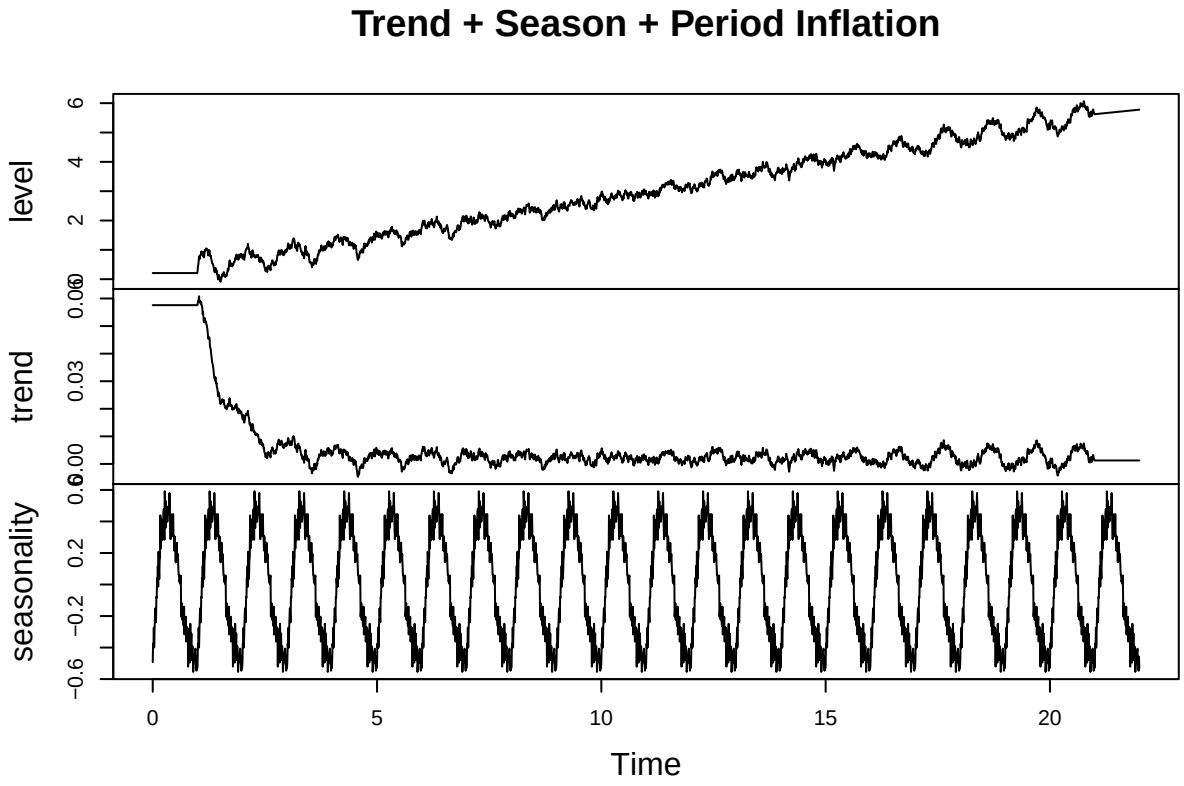

Figure 13: Time series components with trend, seasonality and period inflation.