

Time Series Analysis 02417

Spring 2025

Assignment 1

Tuesday 4th March, 2025 14:46

Instructions:

The assignment is to be handed in via DTU Learn "FeedbackFruits" latest at Monday 3rd at 23:59. You are allowed to hand in in groups of up to max 4 persons. You must hand in a single pdf file presenting the results using text, math, tables and plots (do not include code in the main report - your code must be uploaded as a separate file, it's not being evaluated directly). Arrange the report in sections and subsections according to the questions in this document. Please indicate your student numbers on the report.

This document includes a solution guide for Assignment 1. In the peer-review process you must choose, for each section, one of the following four possibilities:

- 0: The group did not answer the questions or the answer was extremely flawed.
- 1: The group provided a partial answer to the questions, but some parts are uncorrect or missing.
- 2: The group provided a satisfactory answer to the questions (only few parts are missing and minor details may be missing or uncorrect).
- 3: The group provided an excellent answer to the questions.

However, the most important part of the review is that you give constructive feedback in comments, you must at least give the required number of comments set in the peer-review platform.

1 Plot data

In this assignment we will be working with data from Statistics Denmark, describing the number of motor driven vehicles in Denmark. The data is provided to you, but if you are interested you can find it via www.statistikbanken.dk (search for the table: "BIL54"). Together with this document is a file with data `DST_BIL54.csv`, it holds timeseries of monthly data starting from 2018-Jan.

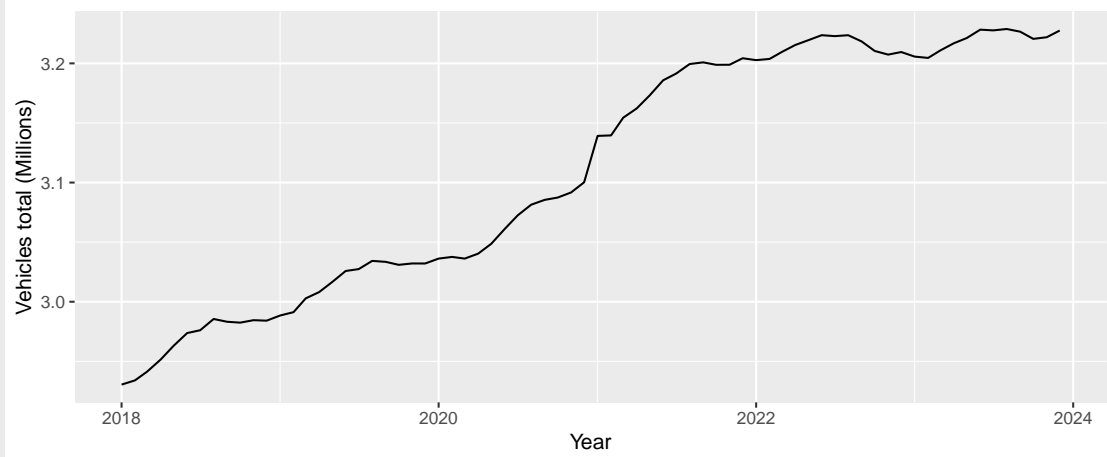
You can decide how to read the data – a script is available in the file `read_data.R`, where the data is read and divided into a training and a test set: The **training set** is from the beginning to 2023-Dec, the **test set** is the last 12 months (2024-Jan to 2024-Dec). **To begin with we will ONLY work with the training set.**

The variable of interest is `total`, which is the number vehicles in registered in Denmark at a given time (in Danish "Drivmidler i alt"). We will ignore the other variables in the dataset.

Do the following:

- 1.1. Make a time variable, x , such that 2018-Jan has $x_1 = 2018$, 2018-Feb has $x_2 = 2018 + 1/12$, 2018-Mar has $x_3 = 2018 + 2/12$ etc. and plot the training data versus x .

The answer of this question should include a plot looking something like this:



Of course different plotting styles can be used (e.g., point plot instead of line plot etc.). The x-axis on the plot you can have x or have the time as years or similar.

- 1.2. Describe the time series in your own words.

The answer to this question is quite open. Some points that could be included are:

- The time series describes total number in vehicles from January 2018 to December 2023.
- The time series seems to have an increasing trend with time.
- The time series seems to exhibit a slight seasonal behavior, with steeper increase in the first half of the year compared to the last half of the year.
- Around year 2022 some change in the increasing trend seems to happen, and the increase slows down.
- At the very last observations a slight increase is seen – but this is only for very few observations, it could keep increasing if the slight seasonal behavior repeats as it did historically.

2 Linear trend model

We will now make a linear trend model, which is a general linear model (GLM) of the form:

$$Y_t = \theta_1 + \theta_2 \cdot x_t + \epsilon_t \quad (1)$$

where $\epsilon_t \sim N(0, \sigma^2)$ is assumed i.i.d. The time is $t = 1, \dots, N$.

- 2.1. Write up the model on matrix form for the first 3 time points: First on matrix form (as vectors and matrices), then insert the elements in the matrices and vectors and finally, insert the actual values of the output vector \mathbf{y} and the design matrix \mathbf{X} (keep max 3 digits). All group participants do it – include picture for each in the report.

Matrix form:

$$Y = X\theta + \varepsilon$$

for $n=3$:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

with inserted values:

$$\begin{bmatrix} 2.930 \\ 2.934 \\ 2.941 \end{bmatrix} = \begin{bmatrix} 1 & 2018 \\ 1 & 2018.083 \\ 1 & 2018.167 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

or:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

3 OLS - global linear trend model

Parameters of the model as a global linear trend model:

- 3.1. Estimate the parameters θ_1 and θ_2 using the training set (call it the Ordinary Least Squares (OLS) estimates). Describe how you calculated the estimates.

This question can be answered in multiple ways. One can provide equations or simply refer to the book. To obtain an excellent answer there must be an explanation for how to estimate both the point estimates and their uncertainty (standard error or variance).

The parameters, θ_1 and θ_2 , can be estimated using Eq. (3.40) from the book:

$$\begin{bmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \end{bmatrix} = \hat{\theta} = (\mathbf{X}^T \mathbf{\Sigma}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{\Sigma}^{-1} \mathbf{Y}$$

where this is the (OLS) case, i.e. $\mathbf{\Sigma} = \mathbf{I}$ (the identity matrix), and we have defined the design matrix \mathbf{X} and the column vector \mathbf{Y} as:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}$$

The standard errors on the parameters, $\sigma_{\hat{\theta}_1}$ and $\sigma_{\hat{\theta}_2}$, can be estimated via Eq. (3.43) from the book, which gives the variance-covariance matrix of the estimates. The standard errors of the individual estimates are then found by taking the square root of the diagonal elements in this variance-covariance matrix:

$$\begin{bmatrix} \hat{\sigma}_{\hat{\theta}_1} \\ \hat{\sigma}_{\hat{\theta}_2} \end{bmatrix} = \sqrt{\text{diag}(\sigma^2(\mathbf{X}^T \mathbf{\Sigma}^{-1} \mathbf{X})^{-1})}$$

where again $\Sigma = I$.

For σ^2 we use the estimated value, $\hat{\sigma}^2$, given by Eq. (3.44) in the book:

$$\hat{\sigma}^2 = \frac{(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\theta}})^T \Sigma^{-1} (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\theta}})}{N - p}$$

with $N = 72$ (number of observations) and $p = 2$ (number of estimated parameters).

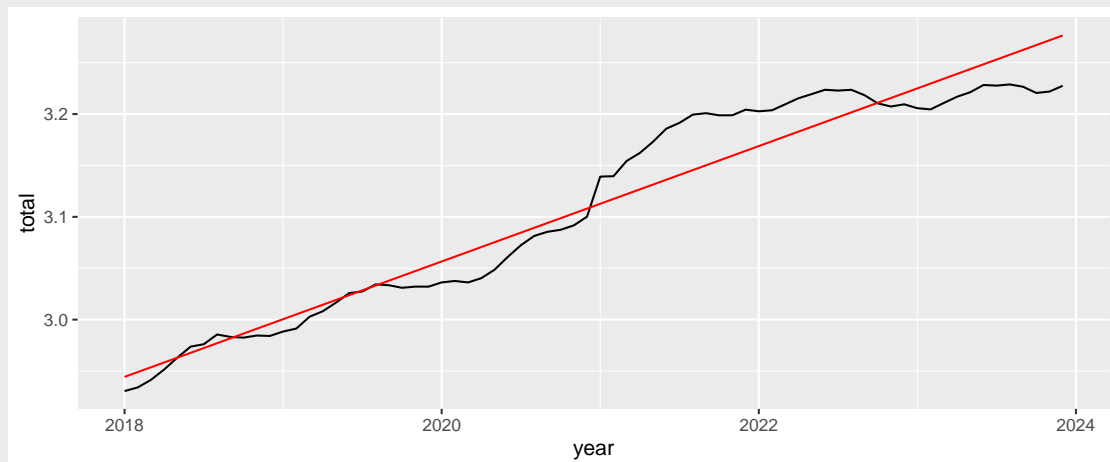
- 3.2. Present the values of the parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ and their estimated standard errors $\hat{\sigma}_{\hat{\theta}_1}$ and $\hat{\sigma}_{\hat{\theta}_2}$. Plot the estimated mean as a line with the observations as points.

The answer to this question should include both point estimates and their standard errors. Note that if the x- or y-axis has been defined differently the parameter estimates will also be different. To obtain an excellent answer in this question the x-axis must have been defined as we did here, i.e., using the x-variable defined in Question 1.1.

The point estimates and standard errors of the parameters are:

| Parameter: | Point estimate: | Standard error: |
|------------|----------------------------|--|
| θ_1 | $\hat{\theta}_1 = -110.36$ | $\hat{\sigma}_{\hat{\theta}_1} = 3.593$ |
| θ_2 | $\hat{\theta}_2 = 0.05614$ | $\hat{\sigma}_{\hat{\theta}_2} = 0.001778$ |

Here θ_1 is the intercept at year zero (January year zero) and has unit "mill. vehicles in total". θ_2 is the slope and has unit "mill. vehicles in total per year".



- 3.3. Make a forecast for the test set, hence the following 12 months - i.e., compute predicted values with corresponding prediction intervals for 2024-Jan to 2024-Dec. Present these values in a table.

Predicted values can be calculated using Eq. (3.59) from the book and prediction intervals can be calculated using Eq. (3.61) from the book.

In order to calculate 95% prediction intervals we must use the 2.5% quantile from the t -distribution with $N - p = 72 - 2$ degrees of freedom. One can also use the 2.5% quantile from the normal distribution (as we have done in the lectures), in the present case it is fine, since the number of observation is not low (thumb rule: $N > 30$ independent observations is ok). Here the result using both distributions is included, and we can see there is very little difference.

Using the t -distribution:

| x | predicted value | prediction interval | |
|----------|-----------------|---------------------|-------|
| 2024.000 | 3.281 | 3.228 | 3.335 |
| 2024.083 | 3.286 | 3.232 | 3.339 |
| 2024.167 | 3.291 | 3.237 | 3.344 |
| 2024.250 | 3.295 | 3.241 | 3.349 |
| 2024.333 | 3.300 | 3.246 | 3.354 |
| 2024.417 | 3.305 | 3.251 | 3.358 |
| 2024.500 | 3.309 | 3.255 | 3.363 |
| 2024.583 | 3.314 | 3.260 | 3.368 |
| 2024.667 | 3.319 | 3.264 | 3.373 |
| 2024.750 | 3.323 | 3.269 | 3.377 |
| 2024.833 | 3.328 | 3.274 | 3.382 |
| 2024.917 | 3.333 | 3.278 | 3.387 |

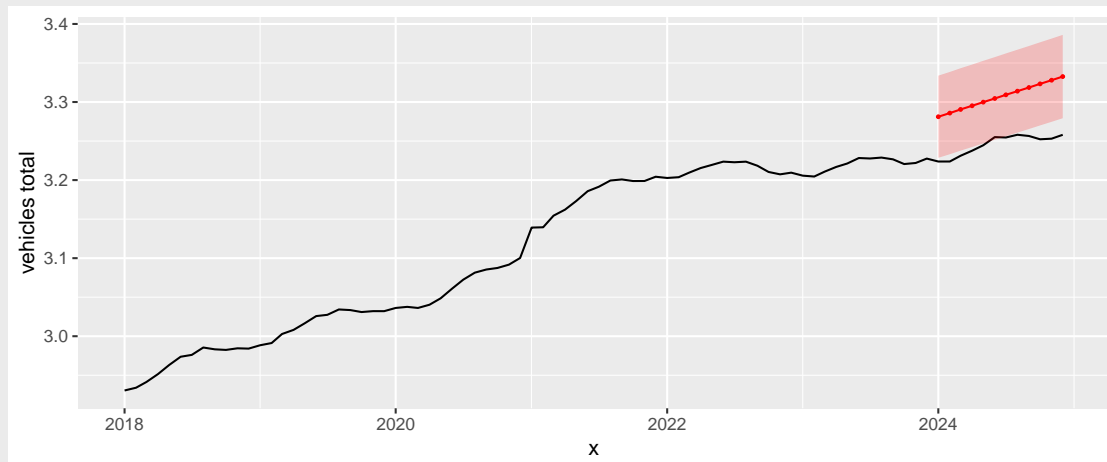
Using the normal distribution:

| x | predicted value | prediction interval | |
|----------|-----------------|---------------------|-------|
| 2024.000 | 3.281 | 3.229 | 3.334 |
| 2024.083 | 3.286 | 3.233 | 3.339 |
| 2024.167 | 3.291 | 3.238 | 3.343 |
| 2024.250 | 3.295 | 3.242 | 3.348 |
| 2024.333 | 3.300 | 3.247 | 3.353 |
| 2024.417 | 3.305 | 3.252 | 3.358 |
| 2024.500 | 3.309 | 3.256 | 3.362 |
| 2024.583 | 3.314 | 3.261 | 3.367 |
| 2024.667 | 3.319 | 3.265 | 3.372 |
| 2024.750 | 3.323 | 3.270 | 3.376 |
| 2024.833 | 3.328 | 3.275 | 3.381 |
| 2024.917 | 3.333 | 3.279 | 3.386 |

As can be seen the prediction intervals are marginal smaller using the normal distribution.

- 3.4. Plot the fitted model together with the training data and the forecasted values (also plot the prediction intervals of the forecasted values).

The plot should look something like this:



The black line is the training data. The red line is the regression line and red dots (with red shaded interval) are prediction (and prediction intervals).

3.5. Comment on your forecast – is it good?

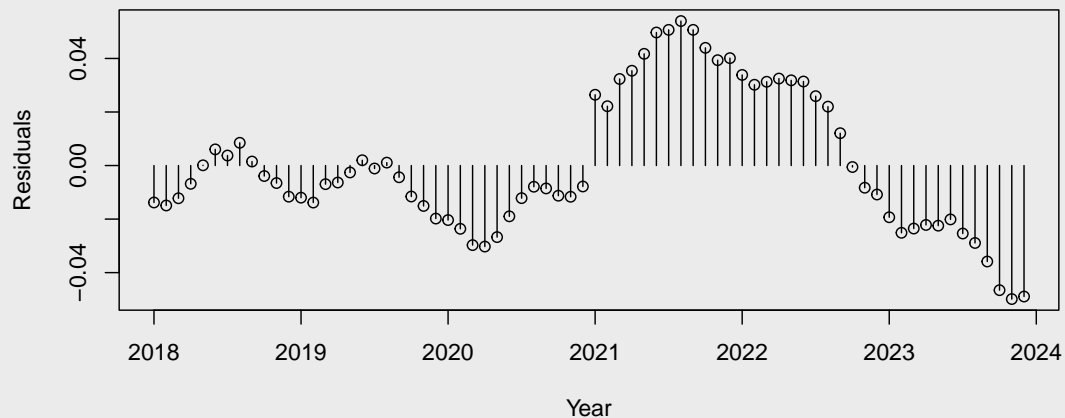
The answer to this question is quite open. Some points that could be included are:

- The fit seems better for the years 2018-2021 than it does for the years after 2022.
- The model does not capture the seasonal pattern at all.
- We have not (yet) investigated whether the model assumptions are valid.
- For later data the model does not seem to fit very good.
- Because the model is global, the predictions end up at a quite high level - they do not seem very plausible. The jump from last observation to first predicted value is quite large.
- Maybe it would be better to make a model that adapts more to data through the period.

3.6. Investigate the residuals of the model. Are the model assumptions fulfilled?

This question can be answered in multiple ways. Here we provide some points (and plots) that could be included.

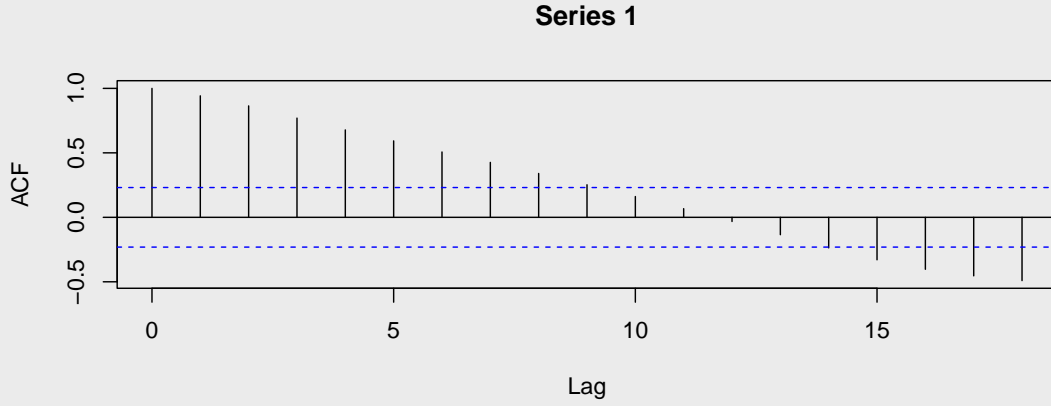
The (OLS) model assumes that residuals are i.i.d. To investigate if the model assumptions are fulfilled we plot the residuals:



Some comments on this plot could be:

- the residuals do not look i.i.d. (plot does not look like white noise with constant variance).
- the residuals in (last part of) 2022 are very large - larger than the rest.
- the residuals look correlated in time.

To inspect correlations in time, we can also try to plot the autocorrelation of the residuals:



4 WLS - local linear trend model

We will now use WLS to fit the linear trend model in Eq. (1) as a local trend model, i.e., the observation at the latest timepoint (N) has weight $\lambda^0 = 1$, the observation at the second latest timepoint ($N - 1$) has weight λ^1 , the third latest observation ($N - 2$) has weight λ^2 etc.

We start by setting $\lambda = 0.9$.

- 4.1. Describe the variance-covariance matrix (the $N \times N$ matrix Σ (i.e. 72×72 matrix, so present only relevant parts of it)) for the local model and compare it to the variance-covariance matrix of the corresponding global model.

The answer to this question should include some representation of Σ , but it is not necessary to write out the full $N \times N$ matrix as long as there is some accompanying explanation.

Sigma is the variance-covariance matrix of the residuals. In this case Σ is a diagonal matrix with inverse "weights" in the diagonal, where weight on x_i is $w_i = \frac{1}{\lambda^{((N-i)+1)}}$ (here the index $i = 0, 1, \dots, (N - 1)$). Sigma is a quite large matrix, so here we show only top left and bottom right values:

$$\Sigma = \begin{bmatrix} 1/\lambda^{N-1} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1/\lambda^{N-2} & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1/\lambda^{N-3} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1/\lambda^2 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1/\lambda^1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1/1 \end{bmatrix}$$

$$= \begin{bmatrix} 1773 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1596 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1436 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1.235 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1.111 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

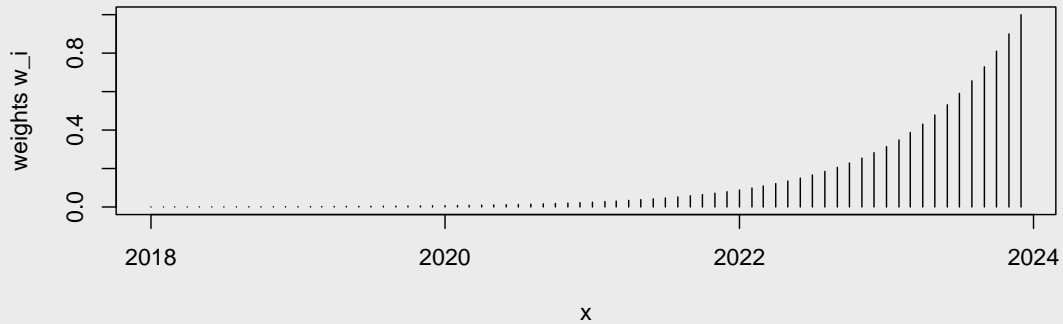
The corresponding matrix for the global model (the OLS model) is the identity matrix:

$$\Sigma_{\text{OLS}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

In the OLS case all the residuals have equal weight (= 1).

- 4.2. Plot the "λ-weights" vs. time in order to visualise how the training data is weighted. Which time-point has the highest weight?

The plot should look something like this:



The latest (i.e. most recent) time-point has the highest weight (= 1). Time-points back in time have weights that are exponentially decaying.

- 4.3. Also calculate the sum of all the λ -weights. What would be the corresponding sum of weights in an OLS model?

The sum of all 72 weights is 9.9949.

In an OLS model the sum of weights would equal $N = 72$.

- 4.4. Estimate and present $\hat{\theta}_1$ and $\hat{\theta}_2$ corresponding to the WLS model with $\lambda = 0.9$.

$$\hat{\theta} = (\mathbf{X}^T \Sigma^{-1} \mathbf{X})^{-1} \mathbf{X}^T \Sigma^{-1} \mathbf{Y}$$

As we do not ask explicitly for standard errors on the estimates, it is satisfactory to provide point estimates for θ_1 and θ_2 .

The parameters are estimated using the same equations as mentioned under Question 2.1, except we use another Σ . If standard errors are estimated, one should substitute N with $T = \sum_i w_i$ (T is the sum of all the weights) in the estimation of σ^2 :

$$\hat{\sigma}^2 = \frac{(\mathbf{Y} - \mathbf{X}\hat{\theta})^T \Sigma^{-1} (\mathbf{Y} - \mathbf{X}\hat{\theta})}{T - p}$$

If the model is defined w.r.t. the x-variable defined above, we have:

| Parameter: | Point estimate: | Standard error: |
|------------|-----------------------------|---|
| θ_1 | $\hat{\theta}_1 = -52.4829$ | $\hat{\sigma}_{\hat{\theta}_1} = 15.1824$ |
| θ_2 | $\hat{\theta}_2 = 0.0275$ | $\hat{\sigma}_{\hat{\theta}_2} = 0.0075$ |

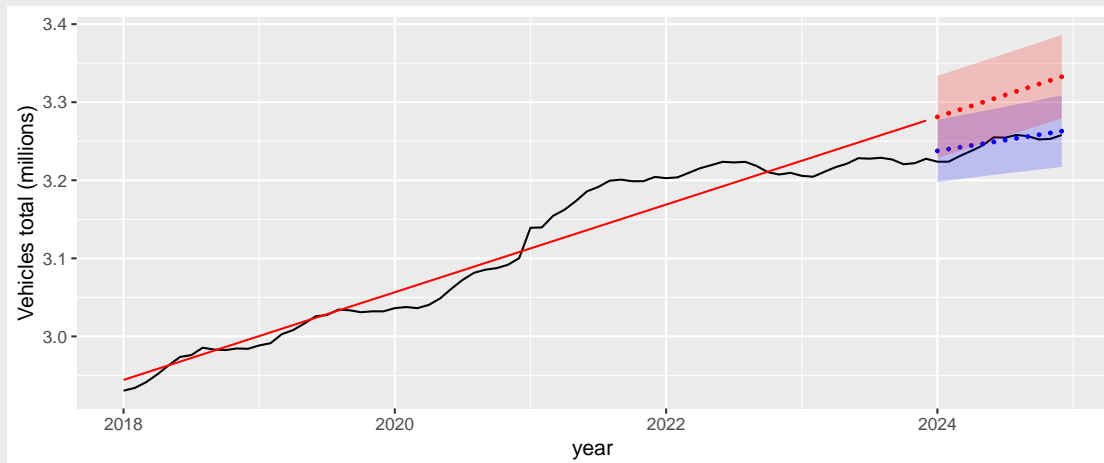
Here θ_1 is the intercept at year zero (January year zero) and has unit "mill. vehicles in total". θ_2 is the slope and has unit "mill. vehicles in total per year".

- 4.5. Make a forecast for the next 12 months - i.e., compute predicted values corresponding to the WLS model with $\lambda = 0.9$.

Plot the observations for the training set and the OLS and WLS the predictions for the test set (you are welcome to calculate the std. error also for the WLS and add prediction intervals to the plots).

Comment on the plot, which predictions would you choose?

The plot could look something like this:



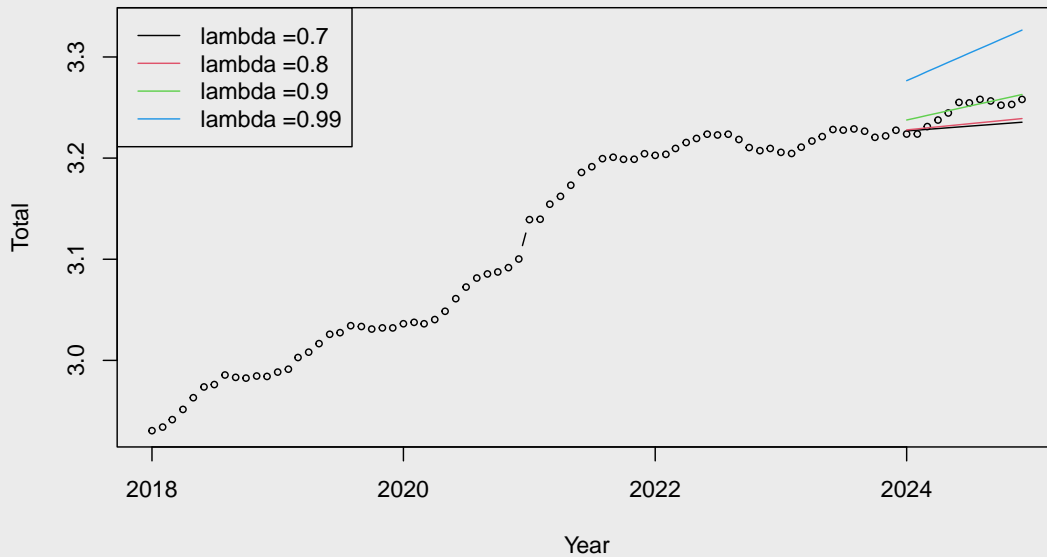
Explanation/comments to the plot:

- Regression lines are plotted as full lines and predictions are plotted as points.
- The black line is the data.
- The blue regression line, corresponds to $\lambda = 0.9$.
- It's clear that the WLS also has adapted more to the level in the end of the period (i.e. the intercept parameter has adapted)
- It's clear that the WLS has adapted to the lower increasing trend in the end of the period.
- The WLS with $\lambda = 0.9$ was actually very good.

4.6. **Optional:** Repeat (estimate parameters and make forecast for the next 12 months) for $\lambda = 0.99, \lambda = 0.8, \lambda = 0.7$ and $\lambda = 0.6$. How does the λ affect the predictions?

Comment on the forecasts - do the slopes of each model correspond to what you would (roughly) expect for the different λ 's?

A plot of the WLS predictions with different λ values could look like:



Some points that could be included in an answer are:

- The regression line with highest λ has "longest memory" and therefore is closer to the global trend model.
- The regression line with the smallest λ ($\lambda = 0.6$) actually couldn't calculate got error "Error in solve.default(Sigma) : system is computationally singular: reciprocal condition number = 1.77312e-16"
- The lower lambdas (0.7 and 0.8) (has relatively short "memory" and therefore the model fits "too much" trend observed only at the very end of the time series.
- A balance of lambda (usually between 0.8 and 0.999) will probably in most cases be optimal – but how to optimize it is not always easy:
 - The model with the smallest λ seems risky because the trend it captures is only present for VERY few observations.
 - The model with smallest λ will be more strongly influenced (fluctuate faster) with new data.
 - The optimal choice of λ should be chosen by inspecting which model performs best in "historic data" (which is what we will do in the next section).

5 Recursive estimation and optimization of λ

Now we will fit the local trend model using Recursive Least Squares (RLS). The smart thing about recursive estimation is that we can update the parameter estimates with a minimum of calculations, hence it's very fast and we don't have to keep the old data.

5.1. Write on paper the update equations of \mathbf{R}_t and $\hat{\boldsymbol{\theta}}_t$.

For \mathbf{R}_t insert values and calculate the first 2 iterations, i.e. until you have the value of \mathbf{R}_2 .

Initialize with

$$R_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

and

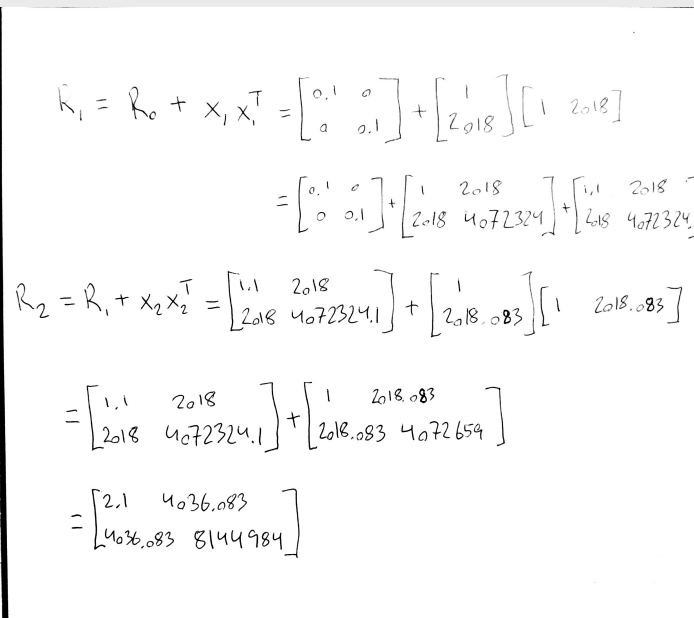
$$\theta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Note, that now the parameters are noted as a vector and thus the subscript is time, so for the linear trend model

$$\hat{\theta}_t = \begin{bmatrix} \theta_{1,t} \\ \theta_{2,t} \end{bmatrix}$$

Everyone in the group must do this on paper and put a picture with the result for each in the report.

Something like this for each group participant, more or less digits could be included:



Handwritten calculations for the Kalman filter update equations:

$$\begin{aligned}
 R_1 &= R_0 + x_1 x_1^T = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2018 \end{bmatrix} \begin{bmatrix} 1 & 2018 \end{bmatrix} \\
 &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 & 2018 \\ 2018 & 4072324 \end{bmatrix} = \begin{bmatrix} 1.1 & 2018 \\ 2018 & 4072324.1 \end{bmatrix} \\
 R_2 &= R_1 + x_2 x_2^T = \begin{bmatrix} 1.1 & 2018 \\ 2018 & 4072324.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2018.083 \end{bmatrix} \begin{bmatrix} 1 & 2018.083 \end{bmatrix} \\
 &= \begin{bmatrix} 1.1 & 2018 \\ 2018 & 4072324.1 \end{bmatrix} + \begin{bmatrix} 1 & 2018.083 \\ 2018.083 & 4072659 \end{bmatrix} \\
 &= \begin{bmatrix} 2.1 & 4036.083 \\ 4036.083 & 8144984 \end{bmatrix}
 \end{aligned}$$

Pheuw, it's nice to have a computer! ;)

- 5.2. Implement the update equations in a for-loop in a computer. Calculate $\hat{\theta}_t$ up to time $t = 3$. Present the values and comment: Do you think it is intuitive to understand the details in the matrix calculations? If yes, give a short explanation.

Something like this should be included (digits and details can vary):

$$\begin{aligned}
 R_1 &= \begin{bmatrix} 1.1 & 2018 \\ 2018 & 4072324.1 \end{bmatrix} \hat{\theta}_1 = \begin{bmatrix} 7.1961 \times 10^{-7} \\ 0.0015 \end{bmatrix} \\
 R_2 &= \begin{bmatrix} 2.1 & 4036.0833 \\ 4036.0833 & 8144984.4403 \end{bmatrix} \hat{\theta}_2 = \begin{bmatrix} 9.7751 \times 10^{-9} \\ 0.0015 \end{bmatrix}
 \end{aligned}$$

$$\mathbf{R}_3 = \begin{bmatrix} 3.1 & 6054.25 \\ 6054.25 & 1.2218 \times 10^7 \end{bmatrix} \hat{\boldsymbol{\theta}}_3 = \begin{bmatrix} -0.0000037 \\ 0.0015 \end{bmatrix}$$

For me it is not intuitive to follow these matrix calculations! It's simply difficult to "see" why multiplying and adding the values in this way calculates the parameters which give least squares predictions.

Reading a model (at least as long as it is linear) gives quite a clear intuitive understanding of the relations between the input variables and the output. Setting up the design matrix X makes sense and one can see how a model and data is set up of the parameter estimation, however following the matrix calculations is very hard – at least in an intuitive way.

The estimated parameters can again be understood and how they evolve over time can often lead to useful insights.

- 5.3. Calculate the RLS estimates at time $t = N$ (i.e. $\hat{\boldsymbol{\theta}}_N$) and compare them to the OLS estimates, are they close? Can you find a way to decrease the difference by modifying some of the RLS initial values and explain why initial values are important to get right?

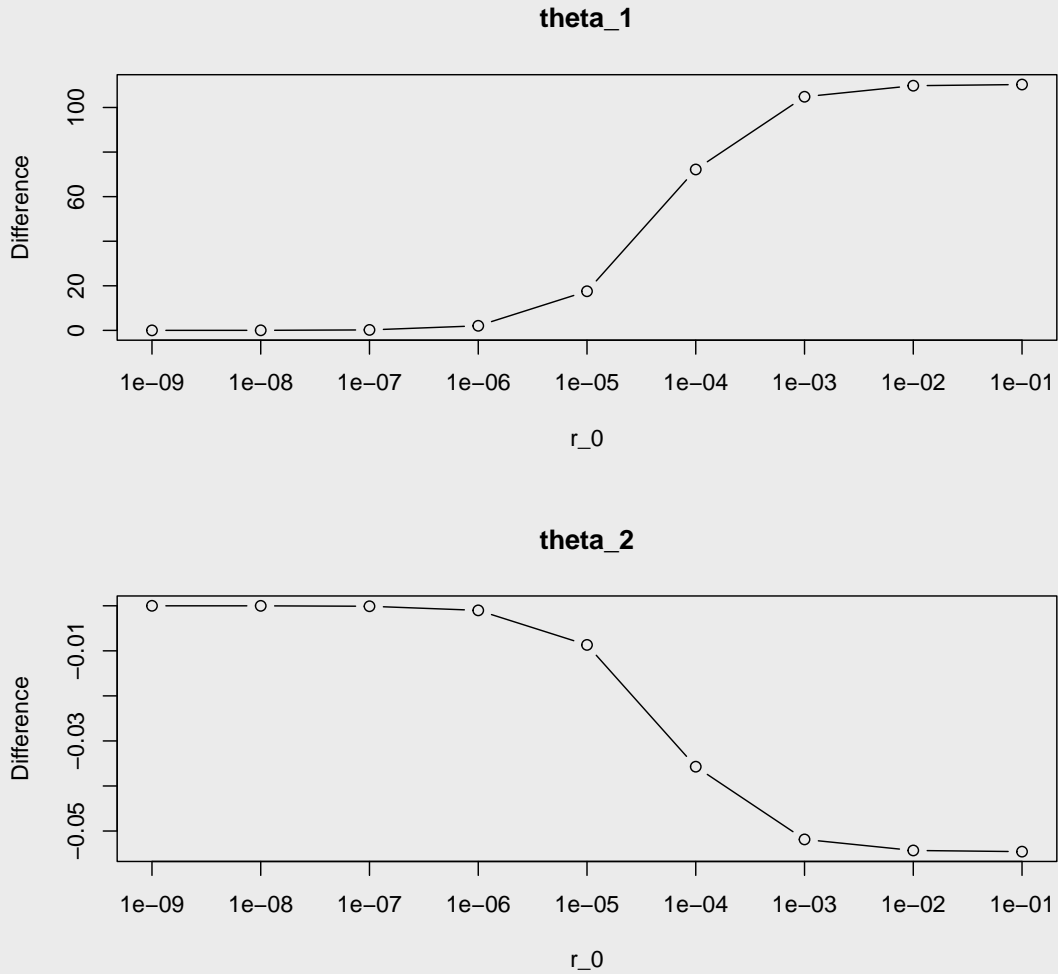
Either, if one knows something about the values of the parameters in the beginning, one can set θ_0 to those, however most of the time, one gets fine results by setting R_0 to a very low diagonal values matrix. If we set

$$R_0 = \begin{bmatrix} r_0 & 0 \\ 0 & r_0 \end{bmatrix} \hat{\boldsymbol{\theta}}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and calculate the difference

$$\hat{\boldsymbol{\theta}}_N - \hat{\boldsymbol{\theta}}_{\text{OLS}}$$

where $\hat{\boldsymbol{\theta}}_{\text{OLS}}$ are the OLS estimates, and run for a sequence of increasing values of r_0 , we get:



5.4. Now implement RLS with forgetting (you just have to multiply with λ at one position in the R_t update).

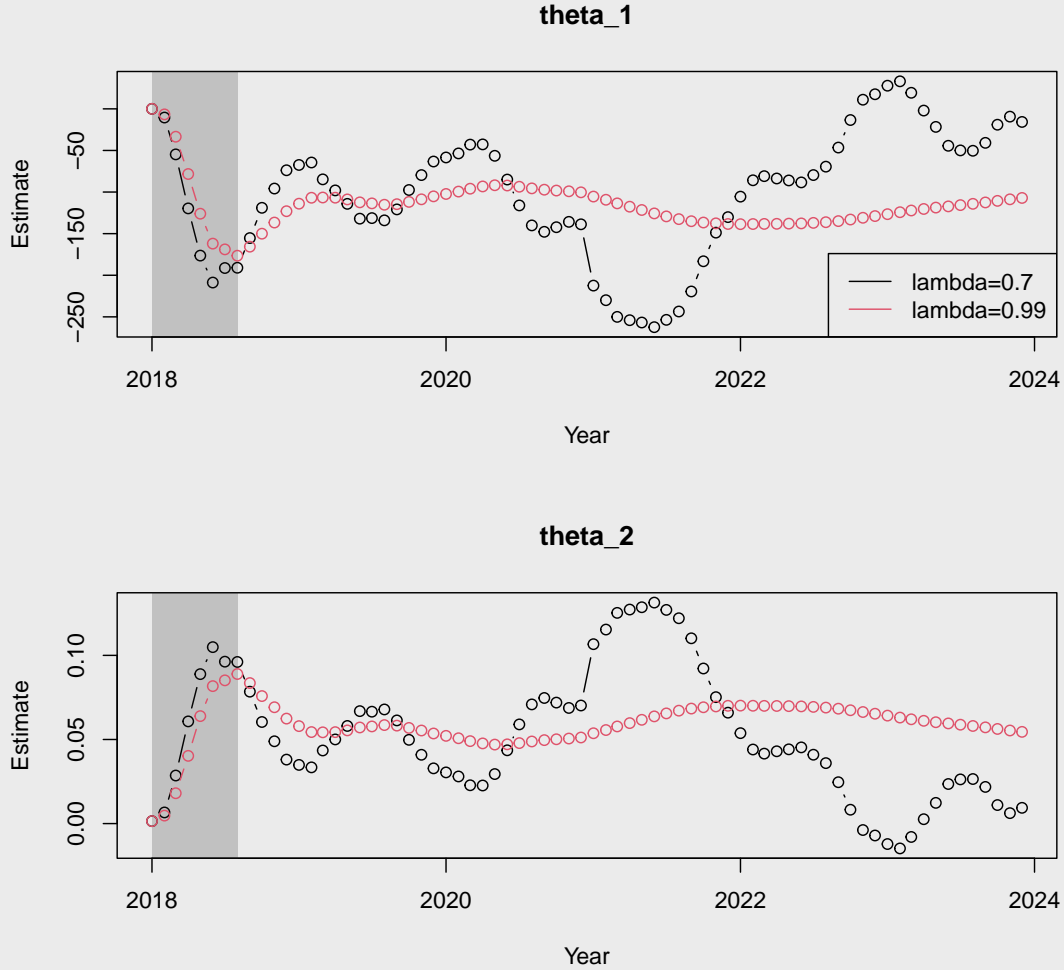
Calculate the parameter estimates: $\hat{\theta}_{1,t}$ and $\hat{\theta}_{2,t}$, for $t = 1, \dots, N$ first with $\lambda = 0.7$ and then with $\lambda = 0.99$. Provide a plot for each parameter. In each plot include the estimates with both λ values (a line for each). Comment on the plots.

You might want to remove the first few time points in the plot, they are what is called a “burn-in” period for a recursive estimation.

Tip: It can be advantageous to put the loop in a function, such that you don’t repeat the code too much (it’s generally always a good idea to use functions, as soon as you need to run the same code more than once).

You might want to compare the estimates for $t = N$ with the WLS estimates for the same λ values. Are they equal?

We make the plots for the two forgetting values:



Some comments should generally be about:

- We clearly see, that the parameters vary more with the lower lambda value, which makes sense, since the estimates will be calculated with more weight on the most recent data.
- We can see some seasonal variation for low lambda value, which must be caused by the seasonal variation in the data.
- For the high lambda value the change is much slower, thus responding to the slower changing trends in the data.
- Lower lambda makes the model more adaptive to changes.

Comparing $\hat{\theta}_N$ to $\hat{\theta}_{\text{WLS}}$ with the same λ value, and setting a low diagonal R_0 , then they are very close.

5.5. Make one-step predictions

$$\hat{y}_{t+1|t} = \mathbf{x}_{t+1|t} \hat{\boldsymbol{\theta}}_t$$

The notation $t + 1|t$ means the variable one-step ahead, i.e. at time $t + 1$, given information available at time t . So this notation is used to denote predictions. For $\mathbf{x}_{t+1|t}$ we do have the values ahead in time for a trend model – in most other situations we must use forecasts of the model inputs.

Now calculate the one-step ahead residuals

$$\hat{\varepsilon}_{t+1|t} = \hat{y}_{t+1|t} - y_{t+1}$$

Note, they could also be written

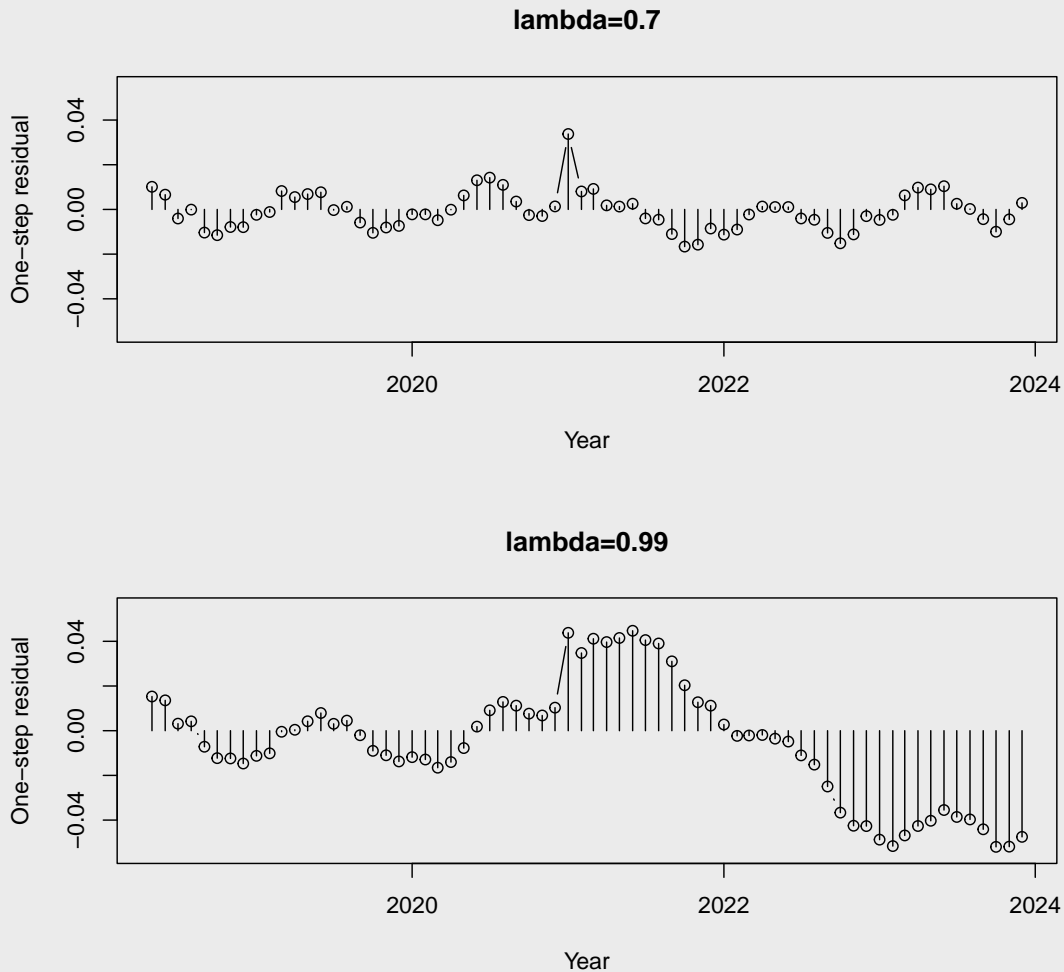
$$\hat{\varepsilon}_{t|t-1} = \hat{y}_{t|t-1} - y_t$$

Applying a shift from “ $t + 1|t$ ” to “ $t|t - 1$ ” makes no difference.

Plot them for $t = 5, \dots, N$ first with $\lambda = 0.7$ and then $\lambda = 0.99$ (note, we remove a burn-in period ($t = 1, \dots, 4$ or more, might not be necessary, but usually a good idea when doing recursive estimation – depends on the initialization values)).

Comment on the residuals, e.g. how do they vary over time?

We make the plots:



Comments could be something like:

- For the one-step residuals, it's clear that the lower lambda (i.e. higher forgetting) is better, since the residuals are generally lower.
- With the lower lambda the model becomes less biased and more varying – as was seen with the variation of the parameter estimates over time.
- Hence, we need to consider the balance between variance and bias, when setting the

5.6. Optimize the forgetting for the horizons $k = 1, \dots, 12$. First calculate the k -step residuals

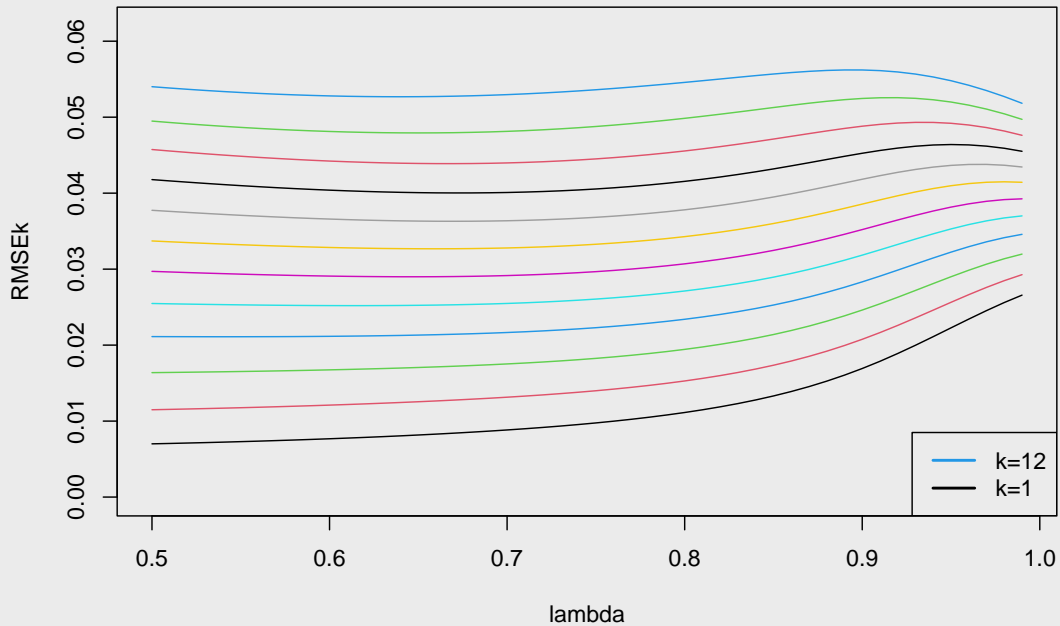
$$\hat{\varepsilon}_{t+k|t} = \hat{y}_{t+k|t} - y_{t+k}$$

then calculate the k -step Root Mean Square Error (RMSE_k)

$$RMSE_k = \sqrt{\frac{1}{N-k} \sum_{t=k}^N \hat{\varepsilon}_{t|t-k}^2}$$

Do this for a sequence of λ values (e.g. 0.5, 0.51, ..., 0.99) and make a plot.

Comment on: Is there a pattern and how would you choose an optimal value of λ ? Would you let λ depend on the horizon?



Comments in line with:

- For the shortest horizons ($k = [1, 2, 3]$) then lambda should be low.
- In the middle horizons ($k = [4, \dots, 9]$) the best lambda should neither be too low nor too high.
- For the longer horizons ($k = [10, 11, 12]$) it seems that no forgetting is best.
- Generally, shorter horizons should have lower forgetting.

- It's not so easy to conclude what is best for longer horizons, there seems to be a local minimum for which could perhaps be caused by:
 - “Too few observations” the trends for longer horizons generally requires more data, hence the results get affected by not having enough data, since asymptotically (having infinite data) the optimum should be in a range
 - A burn-in period can play a role.

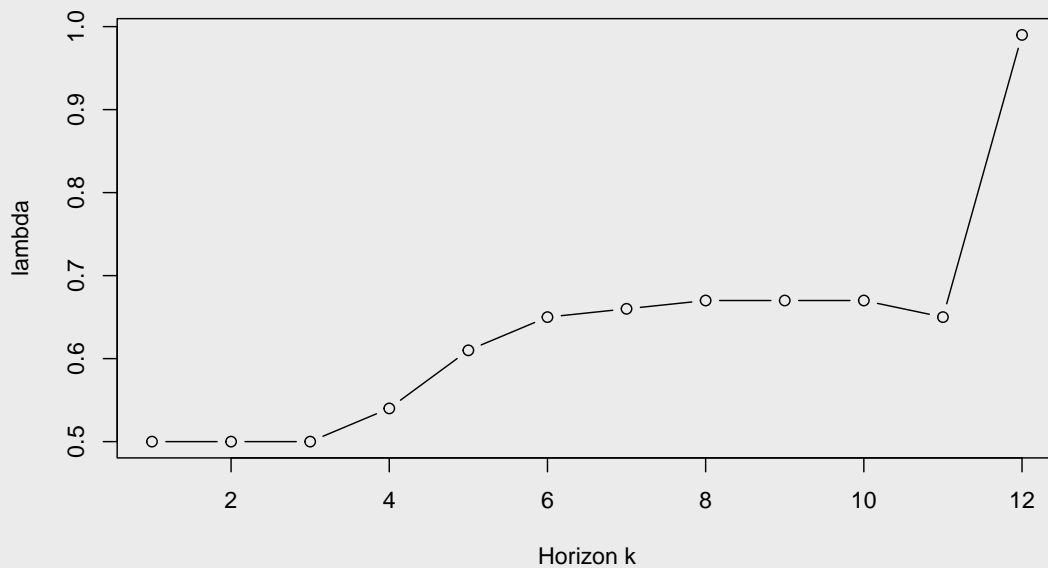
5.7. Make predictions of the test set using RLS. You can use a single λ value for all horizons, or choose some way to have different values, and run the RLS, for each horizon.

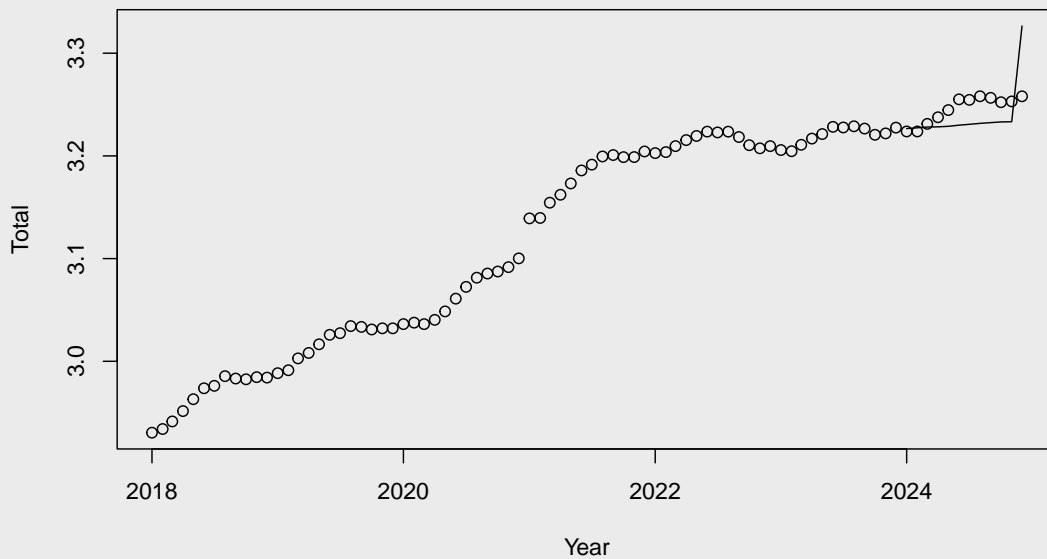
Make a plot and compare to the predictions from the other models (OLS and WLS).

You can play around a bit, for example make a plot of the 1 to 12 steps forecasts at each time step to see how they behave.

This question is somewhat open and it's fine with different choices etc.

One approach would simply be to choose the lambda for each horizon as the one that minimize the $RMSE_k$ for the horizon. That give the sequence in the upper plot and the forecast for the test set in the lower plot:





5.8. Reflexions on time adaptive models - are there pitfalls!?

Again, the considerations can be different, it's fine. Some general insights could be:

- Consider overfitting vs. underfitting.
 - The forgetting must be considered and it will be a balance between bias and variance: high lambda create a bias – low lambda create high variance.
 - Short horizons should generally use higher forgetting.
 - The level of forgetting can affect which model is best to use.
- Are there challenges in creating test sets when data depends on time (in contrast to data not dependent on time)?
 - Yes, since we can not just choose some random points for test set, we should train on the past and then predict future values.
- Can recursive estimation and prediction alleviate challenges with test sets for time dependent data?
 - Yes, recursive estimation is very nice, since we can just predict the next values as we go through the period.

- Can you come up with other techniques for time adaptive estimation?

- There are plenty of ways, e.g. simply a sliding window where at each time point the oldest values are removed and the new values are added.

- Additional thoughts and comments?

- There could be many...