Technical University of Denmark

# Assignment 1

02417 Time Series Analysis

Jonas Wiendl, s243543
Tais Abrahamsson, s193900

March 3, 2025

# Contents

# 1 Data Analysis

The dataset used in this report consists of daily records of the total number of vehicles in Denmark from 2018 to 2025. As a preprocessing step, the data is transformed into monthly data by creating a time variable $x$, such that January 2018 corresponds to $x_0 = 2018$, February 2018 to $x_1 = 2018 + \frac{1}{12}$, March 2018 to $x_2 = 2018 + \frac{2}{12}$, and so on. The dataset is divided into a train and test set and plotted against against $x$ to visualize trends over time, as shown in Figure 1. Overall, the data show an increasing trend, with a strong rise in 2020 and 2021, respectively. In the most recent years, however, the growth appears more stable and less pronounced.

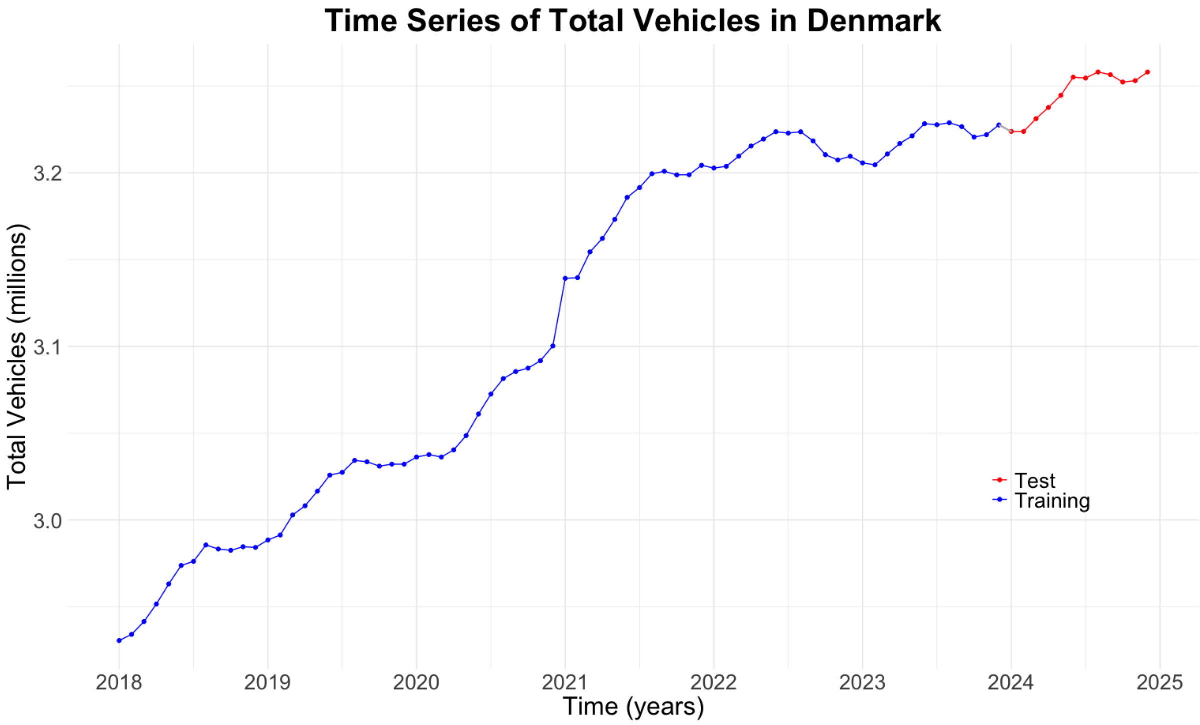



*Figure 1: Time Series of Total Vehicles in Denmark, divided into Training and Test Set, [1]*

# 2 Linear Trend Model

## 2.1 Writing up the model

This section goes through the writing up of a General Linear Trend Model using the first 3 time steps of observations from the abstract Matrix-format to the insertion of the actual target and feature variables.

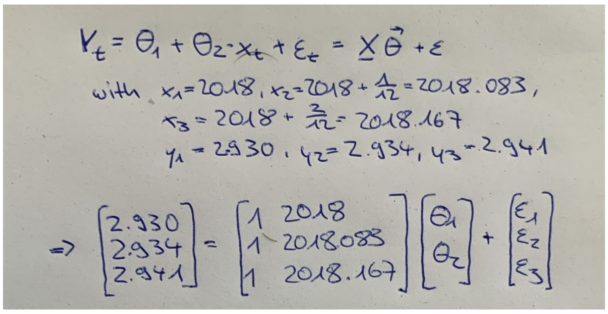



*Figure 2: Jonas's Solution to 2.1*

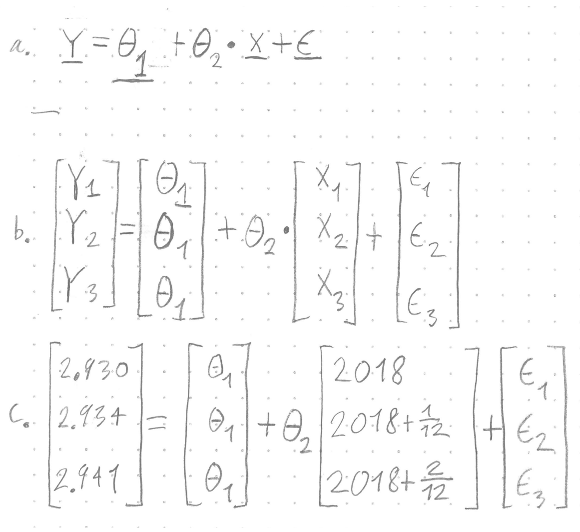


*Figure 3: Tais's Solution to 2.1*

## 2.2 Parameter Estimates and Standard Errors

Using a general linear trend model, the following parameter estimates ($\hat{\theta}$) and corresponding standard error estimates ($\hat{\sigma}_{\hat{\theta}}$) are calculated:

$$\hat{\theta}_1 = -129.520, \qquad \hat{\theta}_2 = 0.066 \tag{1}$$

$$\hat{\sigma}_{\hat{\theta}_1} = 26.684, \qquad \hat{\sigma}_{\hat{\theta}_2} = 0.0132 \tag{2}$$

Below, the fitted model is plotted against the actual (known) values.

## 2.3 Forecast of next 12 months

The following is a table containing the forecast of the next 12 months with upper and lower prediction intervals.

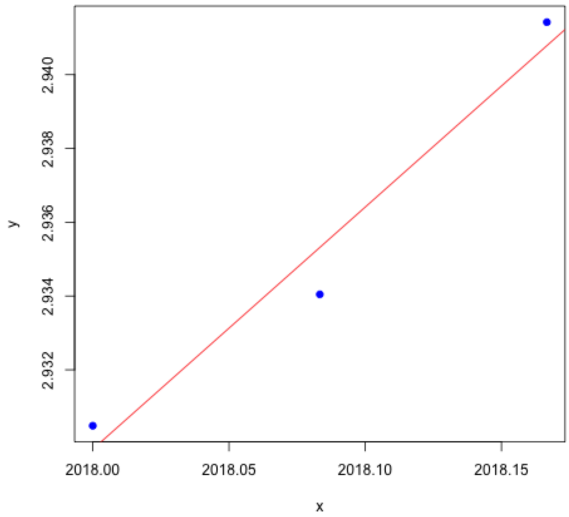



*Figure 4: Fitted model with actual values*

| $x$ | $\hat{y}_{lwr}$ | $\hat{y}$ | $\hat{y}_{upr}$ |
|---|---|---|---|
| 2018.25 | 2.903 | 2.93 | 2.957 |
| 2018.333 | 2.912 | 2.935 | 2.958 |
| 2018.417 | 2.914 | 2.941 | 2.968 |
| 2018.5 | 2.91 | 2.946 | 2.982 |
| 2018.583 | 2.904 | 2.952 | 3 |
| 2018.667 | 2.897 | 2.957 | 3.018 |
| 2018.75 | 2.889 | 2.963 | 3.036 |
| 2018.833 | 2.881 | 2.968 | 3.055 |
| 2018.917 | 2.873 | 2.974 | 3.074 |
| 2019 | 2.865 | 2.979 | 3.093 |
| 2019.083 | 2.856 | 2.985 | 3.113 |
| 2019.167 | 2.848 | 2.99 | 3.132 |
| 2018.25 | 2.84 | 2.995 | 3.151 |
| 2018.333 | 2.831 | 3.001 | 3.171 |
| 2018.417 | 2.823 | 3.006 | 3.19 |

*Table 1: Forecast of next 12 months including upper and lower bounds at a 95% prediction level*

## 2.4 Plot of fitted model against actual values

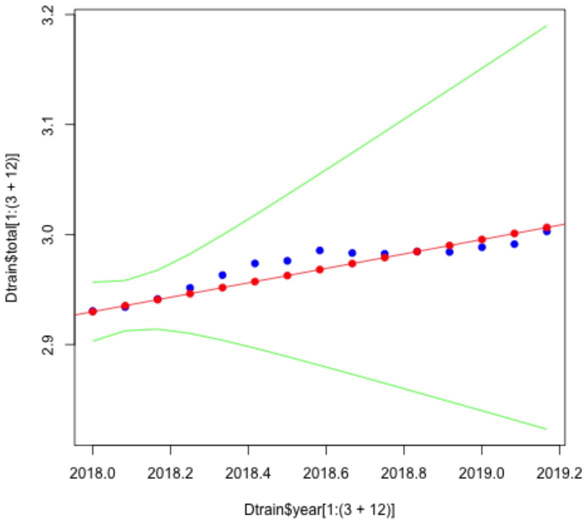



*Figure 5: Plot of model against actual values (including prediction intervals at level=0.95)*

## 2.5 Comments on forecast

The forecast itself looks quite accurate for the given time-interval. The prediction certainty of the model using only 3 observations is, of course, quite low, which is seen by the quickly expanding prediction interval.

## 2.6 Residual Investigation

This part contains a investigation of the nature of the residuals using a Normal Q-Q plot. If the model is accurate, we expect the residuals to be normally distributed, denoted by following the red line in figure 6:

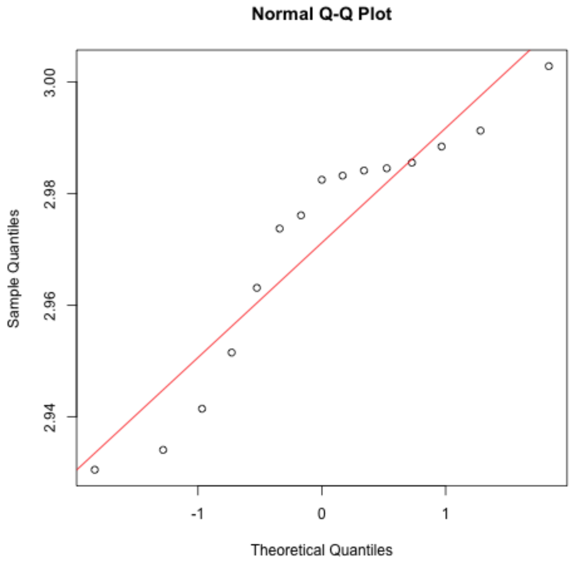



*Figure 6: Q-Q-plot of model residuals*

The Q-Q plot deviates quite a bit from a normal distribution. This may suggest some variables (e.g. seasonal, macroeconomic, etc.) that are not accounted for within the model. Of course it is very likely that a 2-parameter linear model based on only 3 observations is going to be inaccurate.

# 3 WLS - Local Linear Trend Model

In this section, a *Weighted Least Squares* (WLS) model is applied, and the impact of introducing weights is analyzed by comparing it to an *Ordinary Least Squares* (OLS) model as a reference. This extends the linear trend model from Section 2 by utilizing the entire training set.

## 3.1 Comparison Variance-Covariance Matrix OLS and WLS

The variance-covariance matrix ($\Sigma$) describes the uncertainty in residuals. Below, we compare $\Sigma_{\text{OLS}}$ (global model) and $\Sigma_{\text{WLS}}$ (local model).

### OLS Variance-Covariance Matrix

For the Ordinary Least Squares (OLS) model, $\Sigma_{\text{OLS}}$ is a constant diagonal matrix. The first and last 5×5 elements of $\Sigma_{\text{OLS}}$ confirm that variance is equal for all observations:

$$\Sigma_{\text{OLS}} = \begin{bmatrix} 3.90\times 10^{-5} & 0 & \dots & 0 & 0 \\ 0 & 3.90\times 10^{-5} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 3.90\times 10^{-5} & 0 \\ 0 & 0 & \dots & 0 & 3.90\times 10^{-5} \end{bmatrix}$$

### WLS Variance-Covariance Matrix

For the Weighted Least Squares (WLS) model, $\Sigma_{\text{WLS}}$ accounts for the decreasing influence of older observations. The variance becomes smaller as we move from older to newer data. The first and last 2×2 elements of $\Sigma_{\text{WLS}}$ show increasing variance for older observations:

$$\Sigma_{\text{WLS}} = \begin{bmatrix} 0.06916 & 0 & \dots & 0 & 0 \\ 0 & 0.06225 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 4.33\times 10^{-5} & 0 \\ 0 & 0 & \dots & 0 & 3.90\times 10^{-5} \end{bmatrix}$$

### Interpretation

In summary, OLS assumes constant variance for all residuals, treating all data equally, whereas WLS assigns higher variance to older observations, reducing their influence and making WLS more adaptive to recent trends.

## 3.2 Visualizing $\lambda$ Weights

Figure 7 illustrates the exponential weighting scheme used in WLS. The weights follow the formula:

$$w_t = \lambda^{(n-t)}$$

where $\lambda$ is the forgetting factor,$n$ the number of observations and $t$ the time index. Defining the weights in this way ensures that recent data has a stronger influence on the parameter estimation.

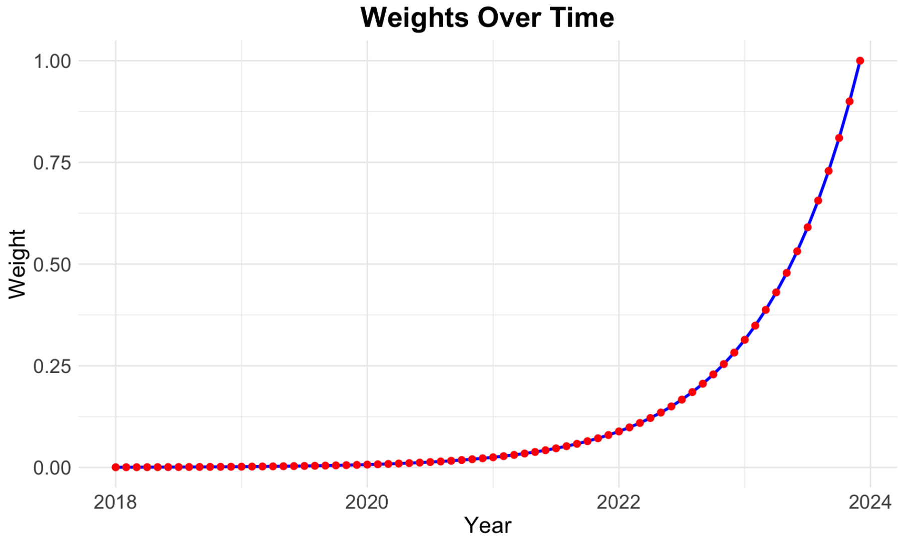



*Figure 7: Weights over time for the WLS model*

## 3.3 Sum of $\lambda$ Weights and Model Parameter Estimates

For OLS the forgetting factor is 1, leading to a equally weighted influence of each observation. Table 2 shows the sum of the $\lambda$ weights, highlighting the influence of the exponential fitting of the weights. It also shows the model parameter estimates, displaying a significant difference in intercept and slope due to the weighting of more recent training data in the WLS.

| **Weighting Method** | **Sum of Weights** | **Intercept ($\theta_1$)** | **Slope ($\theta_2$)** |
|---|---|---|---|
| OLS (Equal Weights) | 72.000 | $-110.3554 \pm 3.5936$ | $0.0561 \pm 0.0018$ |
| WLS ($\lambda = 0.9$) | 9.995 | $-52.4829 \pm 5.1310$ | $0.0275 \pm 0.0025$ |

*Table 2: Comparison of OLS and WLS Parameter Estimates with Standard Errors*

## 3.4 Forecasting Future Vehicle Numbers

Figure 8 shows the forecasts for the OLS and WLS model, together with the training and test data. As suggested in Table 2, the intercept and slope of the two models differ significantly, leading to distinct prediction behaviors. The accuracy of the forecasts can be assessed using the Mean Squared Error (MSE), which is the mean of the squared sum of residuals. The WLS model achieves a significantly lower MSE of 0.00038, compared to the OLS model's MSE of 0.00380, suggesting that for the used data set more recent data points have more impact on the prediction of the total number of vehicles in Denmark.

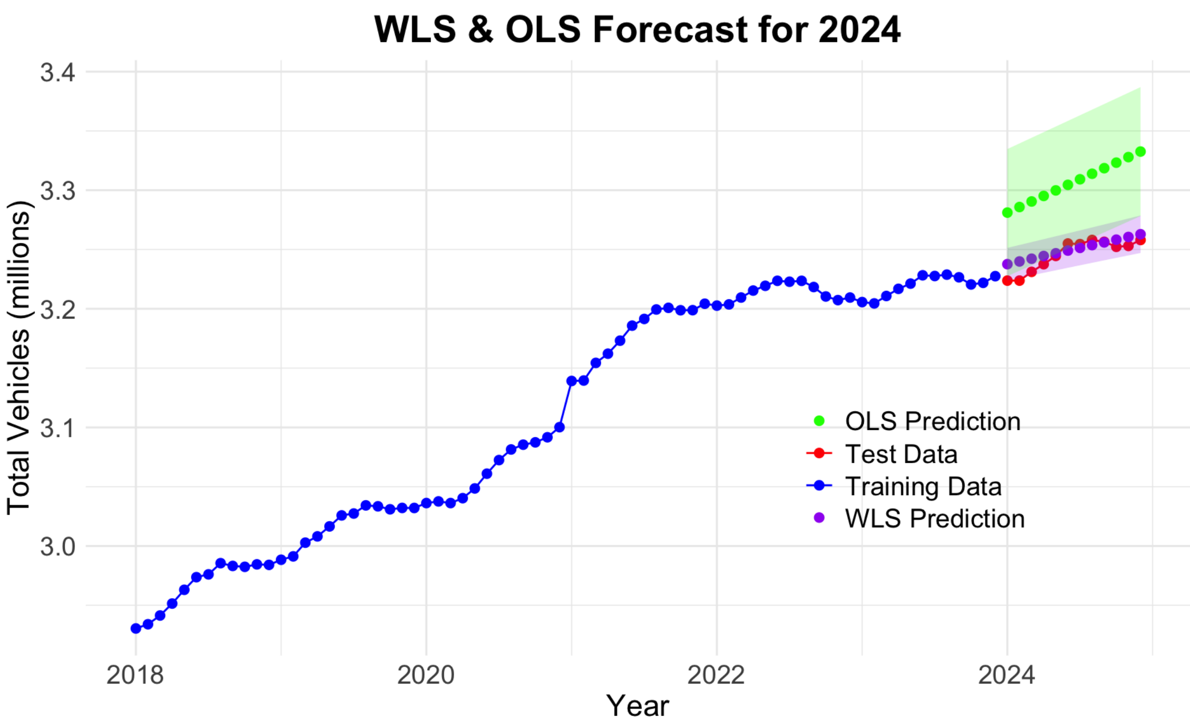



*Figure 8: WLS and OLS Forecasts*

## 3.5 Influence of $\lambda$ Variations on Model Predictions

Finally, the impact of varying the forgetting factor $\lambda$ is analyzed. Figure 9 presents the prediction results for four different WLS models with $\lambda \in \{0.6, 0.7, 0.8, 0.99\}$. The model with $\lambda = 0.99$ behaves almost like an OLS model and deviates significantly from the test data. In contrast, the models with $\lambda = 0.6, 0.7$, and $0.8$ produce similar predictions, with the $\lambda = 0.8$ model yielding the lowest test error. This suggests that, for this dataset, a moderate-to-high forgetting rate is necessary to accurately capture the trend.

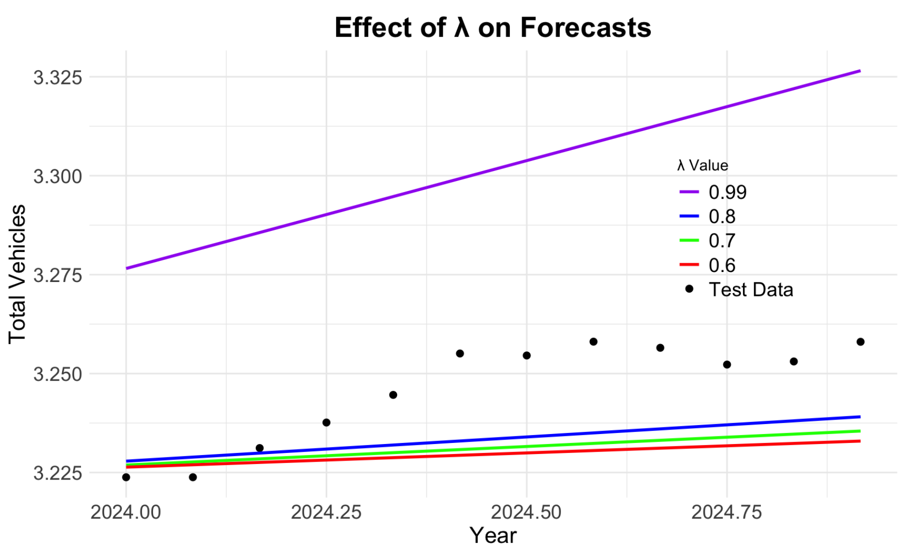
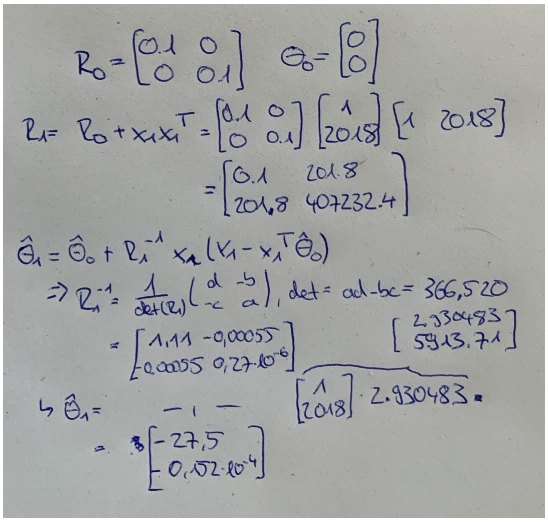


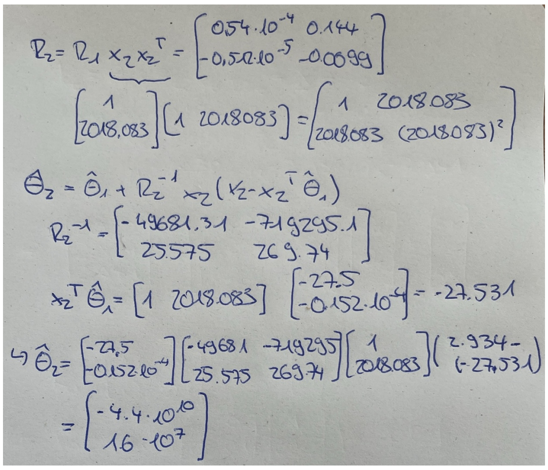

*Figure 9: Prediction of various WLS models with different $\lambda$ values*

# 4 Recursive estimation and optimization of $\lambda$

## 4.1 Manual update equations



(a) Part 1



(b) Part 2

*Figure 10: Jonas's solution to 4.1*

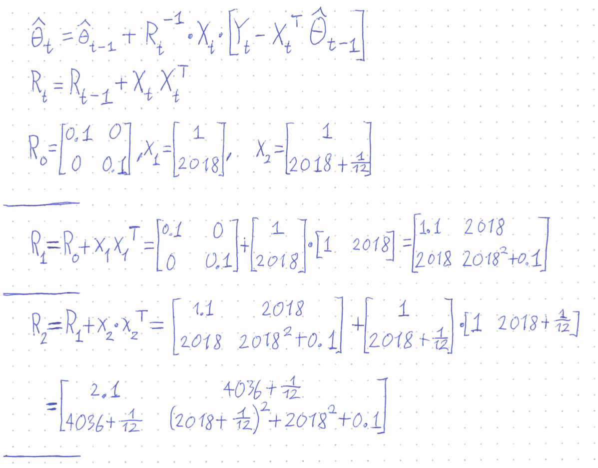


*Figure 11: Tais's solution to 4.1*

## 4.2 For-loop implemented update equations

Using a for loop to calculate $\hat{\theta}_{1...3}$ according to the RLS algorithm results in the following values:

$$\hat{\theta}_1 = \begin{bmatrix} 7.196 \cdot 10^{-7} \\ 1.452 \cdot 10^{-3} \end{bmatrix} \tag{3}$$

$$\hat{\theta}_2 = \begin{bmatrix} 9.775 \cdot 10^{-9} \\ 1.453 \cdot 10^{-3} \end{bmatrix} \tag{4}$$

$$\hat{\theta}_3 = \begin{bmatrix} -3.696 \cdot 10^{-6} \\ 1.455 \cdot 10^{-3} \end{bmatrix} \tag{5}$$

Investigating the model to understand what is actually going on can be quite intuitive, if one compartmentalizes it's different parts.

$$\hat{\theta}_t = \hat{\theta}_{t-1} + R_t^{-1} X_t \cdot [Y_t - X_t^T \hat{\theta}_{t-1}] \tag{6}$$

$$R_t = R_{t-1} + X_t X_t^T \tag{7}$$

Since RLS is a recursive algorithm it should implement its own former value in the form of $\hat{\theta}_{t-1}$. To avoid over-correcting with each new observation, a "weight" is implemented in the form of $R_t^{-1}$. As long as $X_t$ is monotonic, $R_t$ is constantly growing, each new addition to $\hat{\theta}$ will become smaller and smaller as more and more data is available.
If a new observation occurs, that breaks with the apparent pattern, then one R-matrix value will decrease, thus resulting in a larger influence of that observation on $\hat{\theta}$.
$[Y_t - X_t^T \hat{\theta}_{t-1}]$ is the error of the estimate using the model trained on data up until time $t-1$. This means that if our GLM is perfect this term will be equal to zero, and $\hat{\theta}_t = \hat{\theta}_{t-1}$.
If the model overshoots the result of $\hat{\theta}_t = \hat{\theta}_{t-1}$ will be negative, and thus the given $\hat{\theta}_t < \hat{\theta}_{t-1}$.
If the model undershoots, the term will be positive, and $\hat{\theta}_t > \hat{\theta}_{t-1}$.

## 4.3 Model parameter estimates

Calculating model parameter estimates for the entire training dataset results in the following model parameters:

$$\hat{\theta}_{N,RLS} = \begin{bmatrix} -0.058 \\ 0.002 \end{bmatrix} \tag{8}$$

Compared to our OLS-algorithm:

$$\hat{\theta}_{N,OLS} = \begin{bmatrix} -110.355 \\ 0.056 \end{bmatrix} \tag{9}$$

We see that in general our RLS-algorithm is very "slow moving". This is likely due to the magnitude of the time-variable (which starts from 2018 but increases only by intervals of $\frac{1}{12}$. One could change this by preserving the time-intervals, but subtracting 2018 from all values. Moreover, one could (if one has a decent estimate for $\hat{\theta}$ set $\hat{\theta}_{t=0}$ to this value.

## 4.4 RLS with forgetting

We see that implementation of forgetting has a large impact on the "swiftness" of RLS as a method. Allowing the model to more easily "forget" older values (like the models in fig. 12) both increases the volatility of the parameter estimation and allows the model to approximate a system that isn't *exactly* linear, but may show linear traits for some periods.

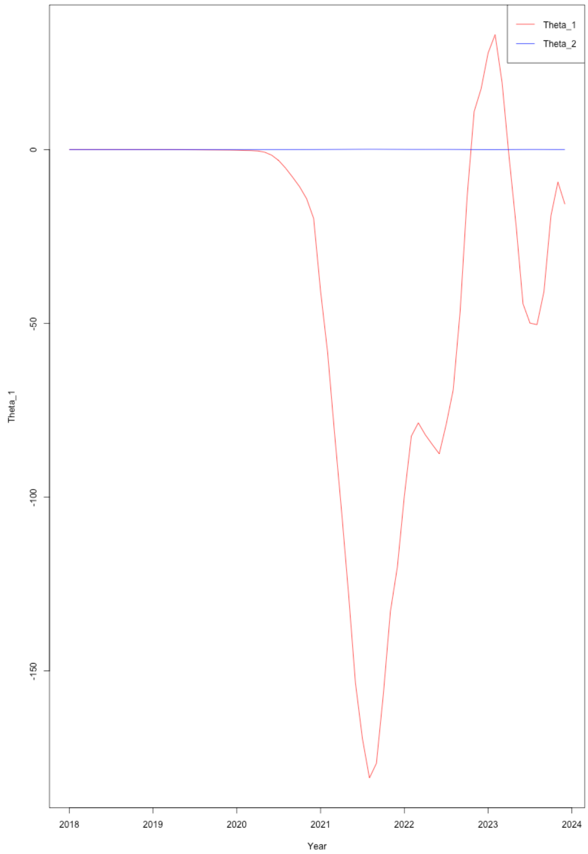



*Figure 12: $\hat{\theta}_1$ estimate, given 1:N observations at $\lambda = 0.7$*

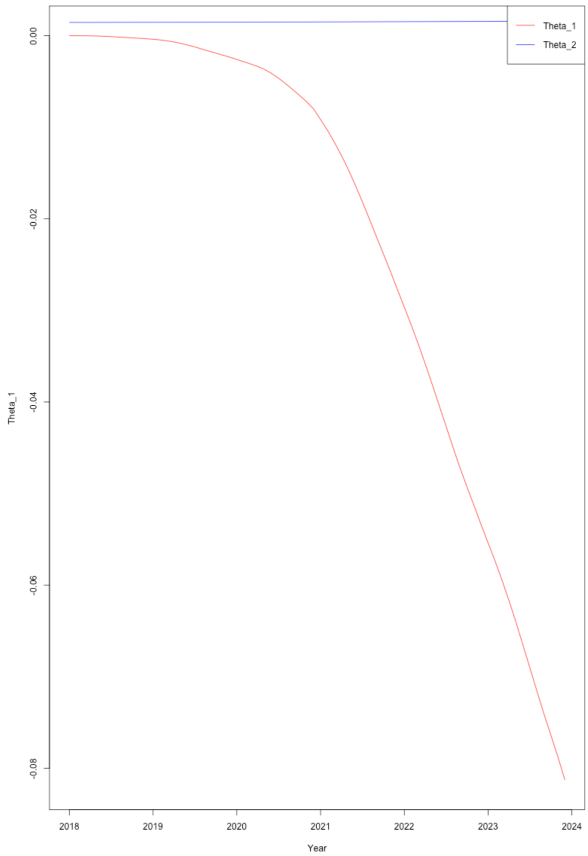



*Figure 13: $\hat{\theta}_1$ estimate, given 1:N observations at $\lambda = 0.99$*

## 4.5 One Step Predictions

This section introduces One-Step-Predictions, which can be computed as follows:

$$\hat{y}_{t+1|t} = x_{t+1}^{\top} \hat{\theta}_t \tag{10}$$

where $\hat{\theta}_t$ represents the estimated parameters at time $t$, and $x_{t+1}$ is the feature vector at time $t+1$. The residuals are given by:

$$\hat{\varepsilon}_{t|t-1} = \hat{y}_{t|t-1} - y_{t-1} \tag{11}$$

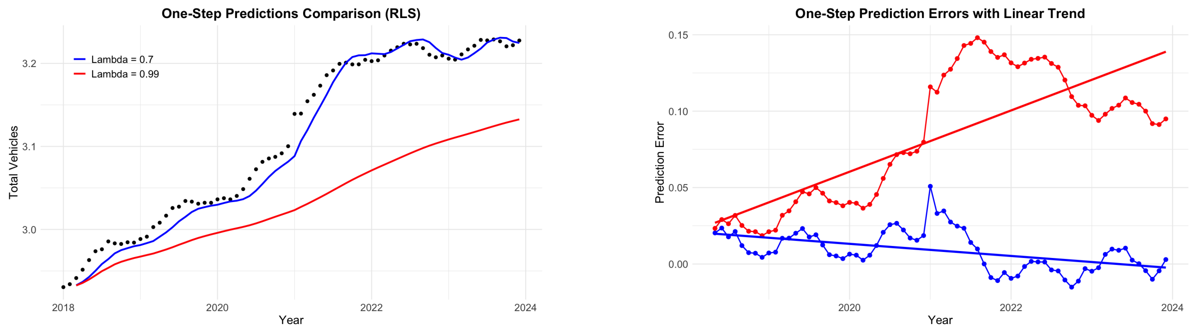
which measure the deviation of the predicted value from the actual observation.
Figure 14(a) shows the one-step predictions for two different forgetting factors, $\lambda \in \{0.7, 0.99\}$. The higher the forgetting rate, defined as $\alpha = 1 - \lambda$, the faster the model adapts to changes, making it more responsive to short-term variations. Conversely, a lower forgetting rate retains more past information, resulting in a smoother, more stable estimate.
The red curve ($\lambda = 0.99$) behaves more like an OLS model due to its long memory, leading to a delayed adaptation to new trends. This results in a noticeable offset from the actual values and increased prediction errors as shown in Figure 14(b). The reason for this offset can be found in the update dynamics of the RLS algorithm:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + R_t^{-1} x_t \left( y_t - x_t^\top \hat{\theta}_{t-1} \right), \tag{12}$$

where the information matrix $R_t$ is updated as:

$$R_t = \lambda R_{t-1} + x_t x_t^\top. \tag{13}$$

Since $\lambda = 0.99$ assigns significant weight to past observations, the parameter estimates $\hat{\theta}_t$ evolve slowly, preventing the model from reflecting recent trends.




(a) One Step Predictions

(b) Prediction Error with Trend Line

*Figure 14: Comparison of One-Step Predictions and Prediction Errors*

## 4.6 Optimization of Forgetting Factor in Relation to Horizon Length

To determine the optimal forgetting factor $\lambda$ for different forecast horizons $k$, the root mean square error (RMSE) was calculated for $\lambda \in [0.5, 0.99]$ and horizons $k = 1, \ldots, 12$. The residuals in steps k were calculated as follows:

$$\hat{\varepsilon}_{t+k|t} = \hat{y}_{t+k|t} - y_{t+k}, \tag{14}$$

and the corresponding RMSE for each horizon was obtained using:

$$\text{RMSE}_k = \sqrt{\frac{1}{N-k} \sum_{t=k}^{N} \hat{\varepsilon}_{t|t-k}^2}. \tag{15}$$

Figure **??** shows how the RMSE varies with $\lambda$ and $k$. The lowest RMSE is achieved for the smallest values of $\lambda$ and $k$. At the highest forgetting rate, an increase of the horizon length leads to a significant increase of the RMSE. A forgetting factor $\lambda > 0.75$ leads to systematically higher RMSE values across all horizons, with a slight increase as $k$ grows. For $\lambda = 0.7$ a minimum appears across horizons, with a slight increase with the horizon length.
In general, this suggests that recent observations are crucial for accurate short-term predictions but can introduce larger errors when forecasting further into the future.

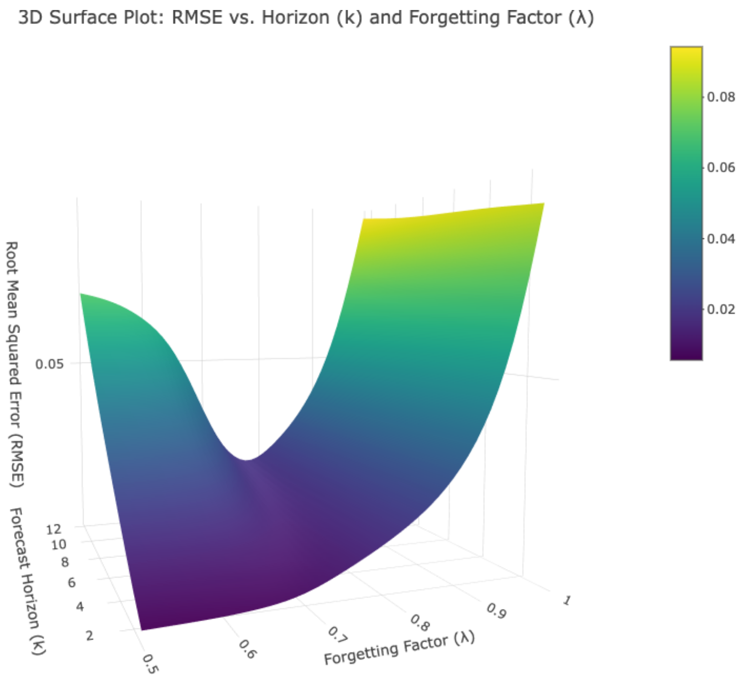



*Figure 15: Optimization of forgetting Factor*

In order to being able to compare the results with the previously developed OLS and WLS models, we select $k = 12$, as this matches the horizon of the other models. The lowest corresponding forgetting factor at $\lambda = 0.77$ is selected. If independent of these comparisons, an adaptive $\lambda$ could be chosen depending on the prediction horizon, where lower values favor shorter-term forecasts and slightly larger values offer more robustness over longer horizons.

## 4.7 Comparison of RLS, WLS, and OLS Predictions

With the chosen forgetting factor and prediction horizon, the forecast for the test set can be made and compared to the WLS and OLS performance, as shown in Figure 16.
The WLS model achieves the best results in terms of residuals, as it assigns greater weight to recent observations while still incorporating past data. In contrast, the RLS forecast appears relatively moderate due to a very moderate trend in the most recent data points and the long forecasting horizon.
The OLS model, as analyzed in Section 3.4, does not account for the stagnation in recent years and instead emphasizes the stronger growth trend observed between 2018 and 2022. As a result, its forecast trajectory is steeper than both WLS and RLS.
Overall, the results illustrate the impact of weighting schemes on forecast accuracy. While OLS provides a global fit, WLS and RLS enable dynamic adaptation, with WLS achieving the best balance between past and recent trends.

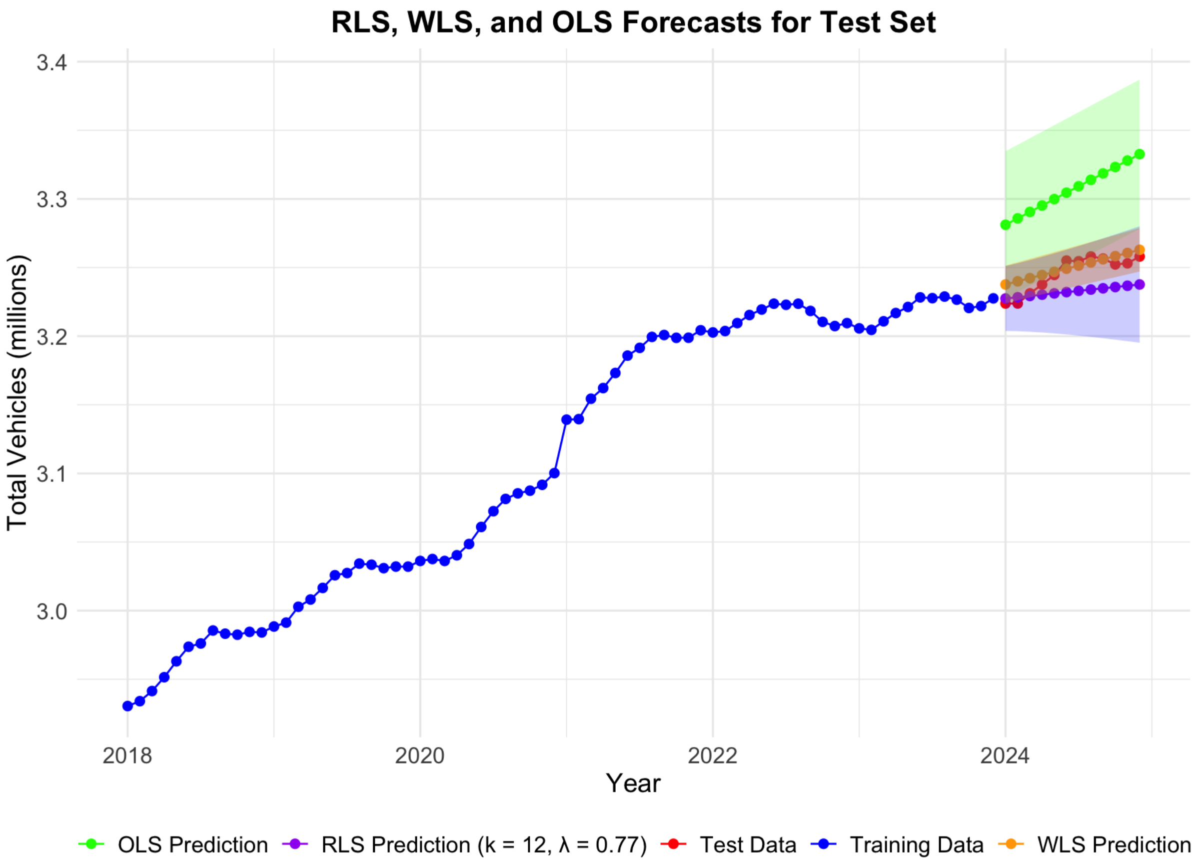



*Figure 16: Predictions for the Test Set*

## 4.8 Reflections

Time adaptive models are in general more versatily wrt. systems that change over time. Of course if an algorithm such as that of RLS without forgetting is implemented, one risks over-fitting to specific data periods without taking into account the changing of a system over time (e.g. seasonality, externalitites, etc.).
Instead of splitting training-/test data into a "before" and "after" some date it may make sense to split the dataset into different subsets that are coherent but spread over time.
While recursive estimation of one kind or another may alleviate some of the problems of an "overfitted" model, no model is stronger than the representativeness of its training data.

# 5 References

[1] Statistics Denmark. (2025). Statistikbanken [Accessed: February 21, 2025]. https://www.statistikbanken.dk/statbank5a/default.asp?w=1512