Danmarks
Tekniske
Universitet

# Assignment 1


Authors

Bao Ngoc Thai - s242504
Erika Samantha Young - s242666
Raul Labarthe Saric - s233840


July 14, 2025

# Contents

# 1 Plot data

This assignment will be utilizing data from Statistics Denmark, looking at the number of motor driven vehicles in Denmark from January 2018 to December 2024. The data is split into two, training data and testing data, for model building purposes. The training data is sectioned from January 201 8 to December 2023, while the testing data is all months of 2024.

A time variable is made by adding months as fractions to the year to create a continuous variable. For example, February of 2019 will be $(2019 + 1/12)$. This time variable plotted against the total vehicles registered in Denmark for the training data is shown in Figure 1. Note that the x-axis is converted back to original time scale for visibility purposes.

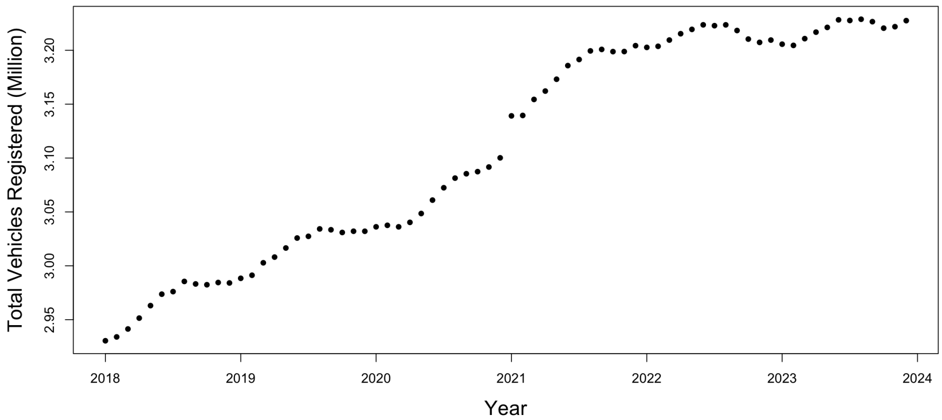



Figure 1: Number vehicles in registered in Denmark from 2018 to 2023

Figure 1 exhibits an overall increasing trend over time, suggesting the presence of potential non-linear patterns. The trend levels in recent years, potentially showing diminishing growth. Additionally, there is a noticeable jump in registered vehicles between late 2020 and early 2021.

# 2 Linear trend model

Assignment 1, Q2.1

1) Model on matrix fom : $\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_t \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_t \end{bmatrix}$

2) Model : $Y_t = \theta_1 + \theta_2 X_t + \varepsilon_t$

3) First 3 values : $\begin{bmatrix} 2.930 \\ 2.934 \\ 2.941 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0.083 \\ 1 & 0.167 \end{bmatrix}, + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$

2.1

$$Y_t = \theta_1 + \theta_2 \cdot x_t + \varepsilon_t \quad , \quad \varepsilon_t \sim N(0, \sigma^2) \text{ iid}, \quad t = 1, \dots, N$$

matricial form

$$y = X\theta + \varepsilon \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2,930 \\ 2,934 \\ 2,941 \end{pmatrix} \; ; \; X = \begin{pmatrix} 1 & 2018.000 \\ 1 & 2018.083 \\ 1 & 2018.167 \end{pmatrix}$$

$$\begin{pmatrix} 2.930 \\ 2.934 \\ 2.941 \end{pmatrix} = \begin{pmatrix} 1 & 2018.000 \\ 1 & 2018.083 \\ 1 & 2018.167 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix}$$

2.1 Linear Trend Model

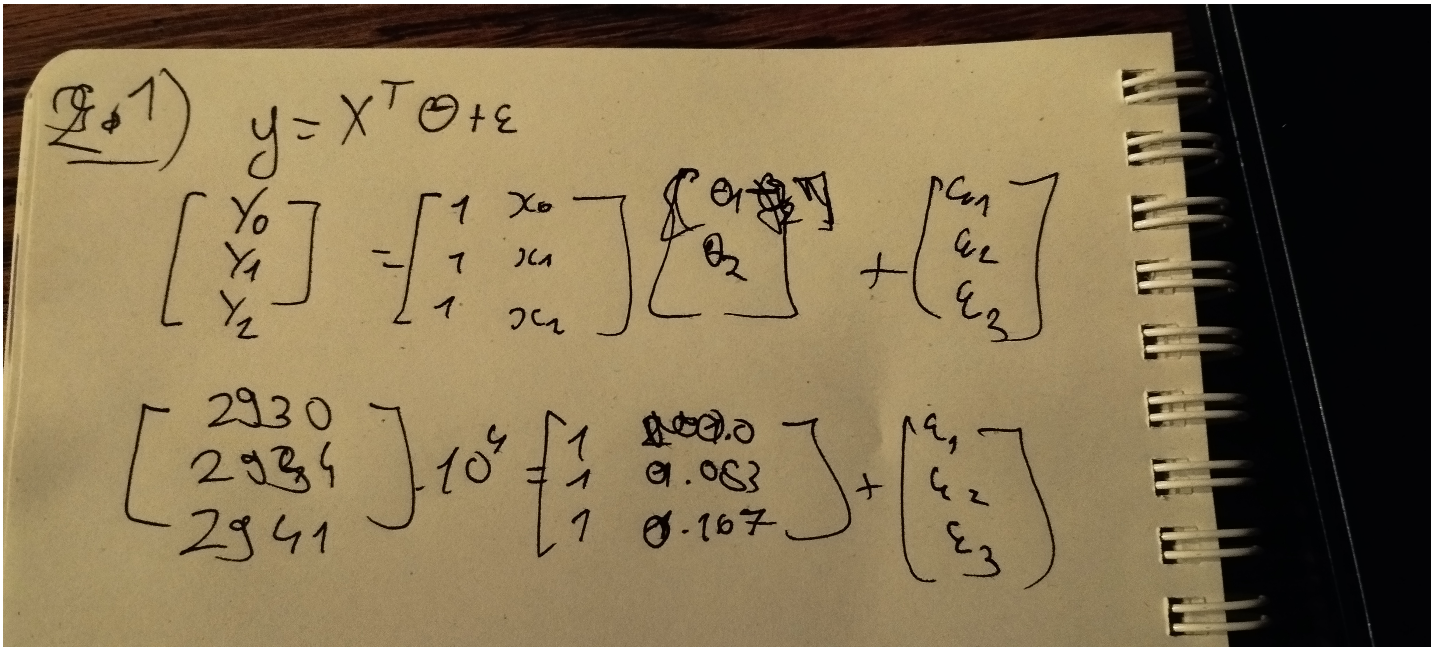
$$Y_t = \theta_1 + \theta_2 \cdot x_t + \varepsilon_t$$

$$\begin{pmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_N^T \end{pmatrix}$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix} \qquad [\, Y = X\theta + \varepsilon \,]$$

First 3 time points:

$$\begin{pmatrix} 2.930483 \\ 2.934044 \\ 2.941422 \end{pmatrix} = \begin{pmatrix} 1 & 2018.000 \\ 1 & 2018.083 \\ 1 & 2018.167 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix}$$



# 3 OLS - global linear trend model

## 3.1 Parameter and standard error estimates: Methodology

Parameter estimation is done by solving for $(X^TX)^{-1}X^Ty$, where $X$ represents the design matrix and $y$ is the observation vector. To facilitate this computation, a transformed time variable was introduced, setting January 2018 as the reference point. For instance, February 2019 is represented as $(2019 + 1/12) - 2018$. This allows to make sense of the intercept value and not be an arbitrary value representing registered vehicles in year zero.

## 3.2 Parameter and standard error estimates: Results

The estimated parameters and their corresponding standard errors are as follows;

$$\hat{\theta}_1 = 2.944, \quad \hat{\theta}_2 = 0.0561$$
$$\hat{\sigma}_{\hat{\theta}_1} = 0.00610, \quad \hat{\sigma}_{\hat{\theta}_2} = 0.00178$$

A visualization of the estimated prediction mean using these coefficients plotted against the data points are shown in Figure 2.

## 3.3 Forecast: Table

Table1 presents the predicted number of motorized vehicles in Denmark based on the OLS model with year given in rolling time from 2018-Jan.

## 3.4 Forecast: Figure

Figure 3 presents the predicted number of motorized vehicles in Denmark in 2024 January-December using the OLS model.

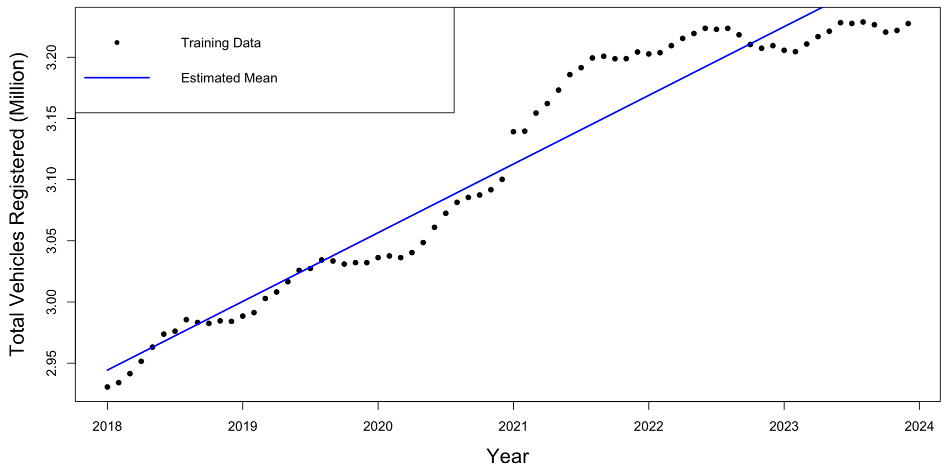



Figure 2: Training Data plotted against estimated mean

Table 1: Prediction for no. of vehicles for 12 months (2024 Jan-Dec).

| Year-Month | Year | No. of vehicles (million) |
|---|---|---|
| 2024-1 | 6.000 | 3.281 |
| 2024-2 | 6.083 | 3.286 |
| 2024-3 | 6.167 | 3.291 |
| 2024-4 | 6.250 | 3.295 |
| 2024-5 | 6.333 | 3.300 |
| 2024-6 | 6.417 | 3.305 |
| 2024-7 | 6.500 | 3.309 |
| 2024-8 | 6.583 | 3.314 |
| 2024-9 | 6.667 | 3.319 |
| 2024-10 | 6.750 | 3.323 |
| 2024-11 | 6.833 | 3.328 |
| 2024-12 | 6.917 | 3.333 |

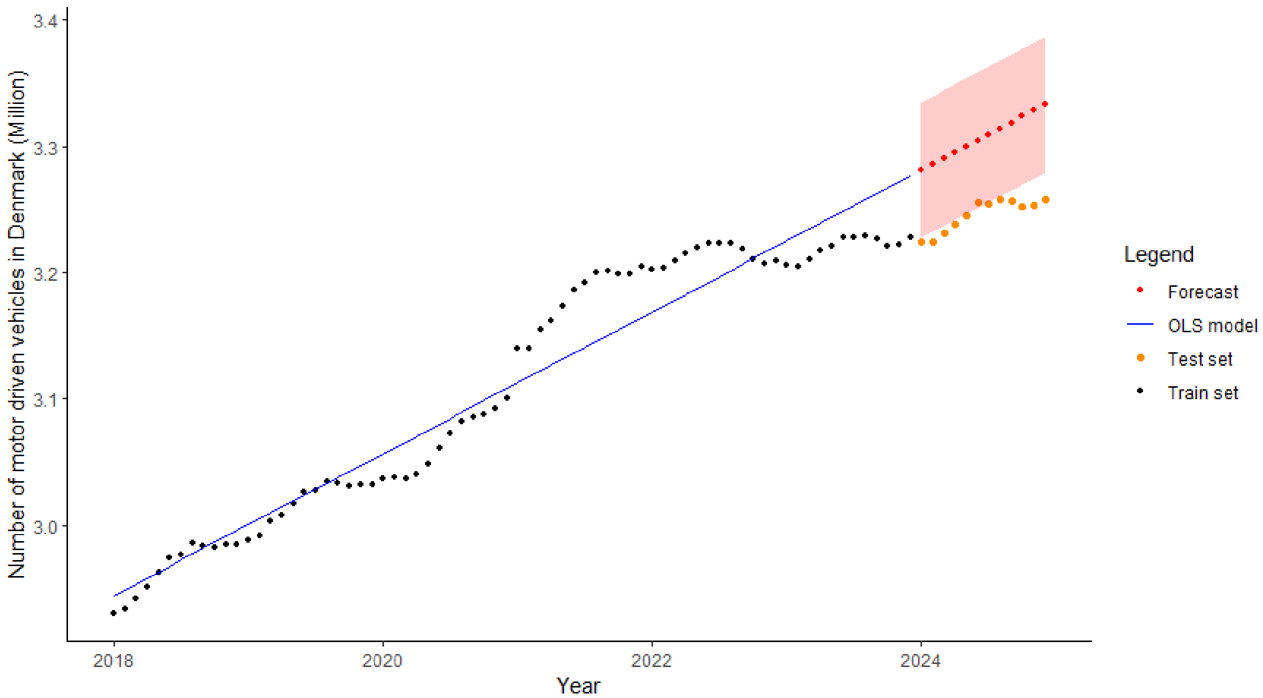



Figure 3: Forecast motor driven vehicles in Denmark from 2024 Jan to 2024 Dec.

### 3.5 Forecast: Discussion

This forecast (red dots) deviates greatly from the observed values (orange dots). The current OLS model captures well the increasing trend from 2018 to mid-2020 but could not capture the dynamic change overtime, e.g. sudden increase from 2020 to 2022, the lower rate of increase and fluctuation from 2023 onwards.

### 3.6 Model: Residuals

Figure 4 presents the residual analysis. Note that residual is calculated as observed minus predicted values, i.e. a positive residual indicates observed value larger than predicted one (i.e. the model underestimates) and vice versa.

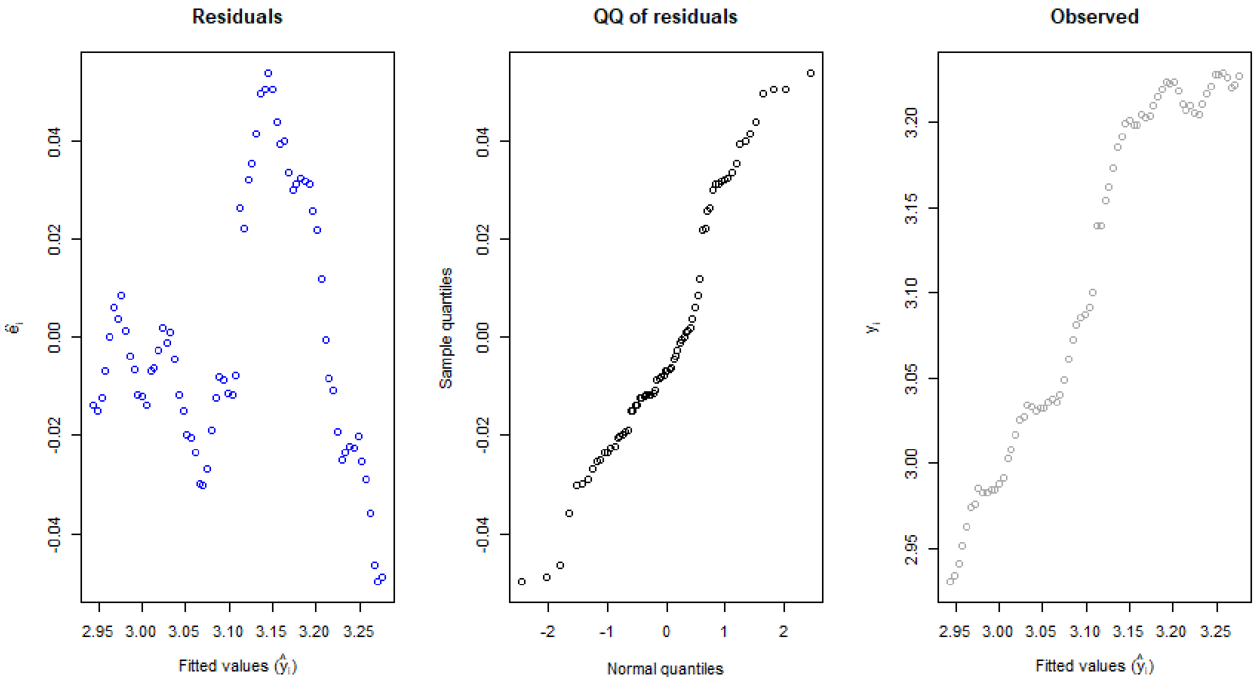



Figure 4: Residuals for OLS model.

The Ordinary Least Squares (OLS) model assumes the following:

- Linearity: The relationship between the independent variables and the dependent variable is linear. For current data, the OLS model can be applied piecewise.
- Independence: The residuals (errors) are independent of each other. This means that there is no correlation between the residuals. The first plot in Figure 4 indicates that the residuals are not randomly scattered around 0, indicating a certain degree of autocorrelation.
- Normality: The residuals are normally distributed. Currently with the OLS model, the residuals do not follow a normal distribution as seen in the Q-Q plot, which does not satisfy the current assumption $\epsilon \sim N(0, \sigma^2)$.

Autocorrelation can be checked using the autocorrelation function (ACF) and partial autocorrelation function (PACF). ACF measures the correlation between a time series and its lagged values over different time intervals, and PACF measures the correlation between a time series and its lagged values, controlling for the values of the time series at all shorter lags. Figure 5 indicates that there is a gradual decay in ACF indicating the autoregressive (AR) process. In addition, the figure also shows a cutoff after lag 1 in PACF indicating an AR(1) process, suggesting that the current value of the series is based on the immediately preceding value.

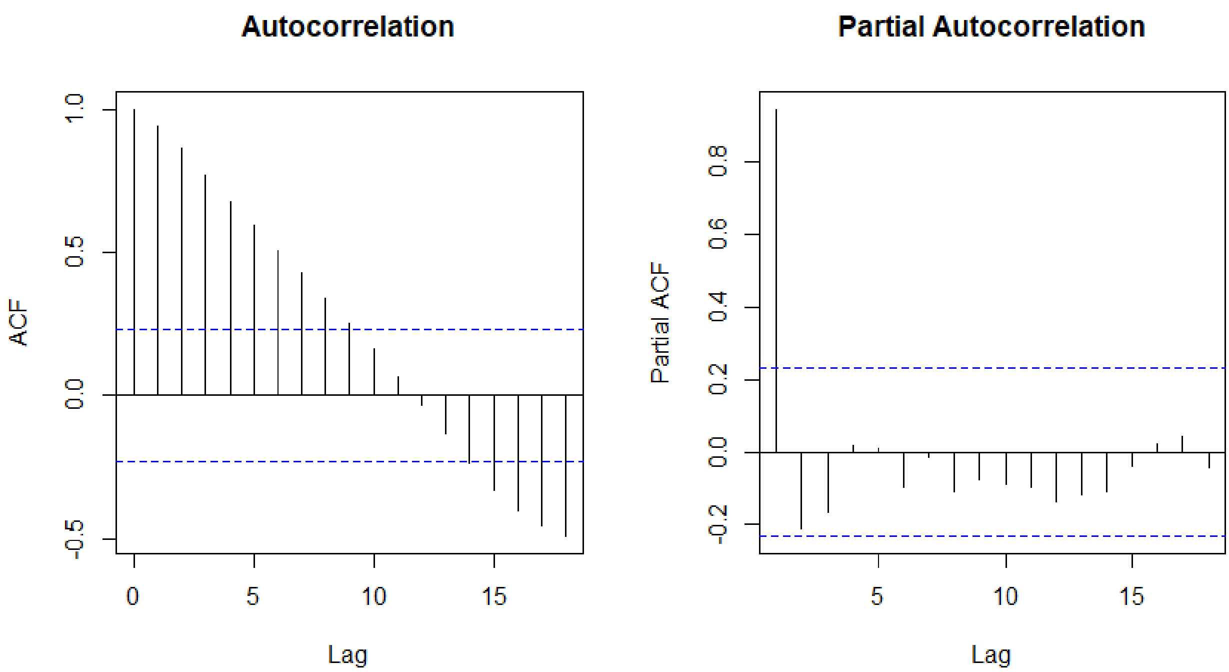



Figure 5: ACF and PACF analysis.

# 4 WLS - local linear trend model

## 4.1 Variance-covariance matrix ($\Sigma$)

The variance-covariance matrix $\Sigma$ for the local model is an $N \times N$ diagonal matrix representing a prior assumption about the variance of the observations. In this case, $N = 72$, corresponding to the number of observations in the training set. The model assumes that recent observations are more informative than older ones, a common approach in time series models with exponential forgetting mechanisms.

The hyperparameter that calibrates the model is the forgetting factor $\lambda$, where $\lambda \in (0, 1)$. A value of $\lambda = 0$ implies absolute forgetting, while $\lambda = 1$ assigns equal weight to all observations. Here, $\lambda = 0.9$ determines the elements of $\Sigma$ as:

$$\Sigma_{ii} = \frac{1}{\lambda^{N-i}} = \frac{1}{\omega_i}, \quad i = 1, \ldots, N$$

A representative portion of $\Sigma$ is:

$$\Sigma_{local} = \begin{pmatrix} 1773.3 & 0 & 0 & \cdots & 0 \\ 0 & 1596.0 & 0 & \cdots & 0 \\ 0 & 0 & 1436.4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

The diagonal elements decrease exponentially, reflecting higher confidence in more recent observations, since the weights $\omega_i$ are the inverses of the $\Sigma_{ii}$ elements. The off-diagonal zeros indicate that no covariance is assumed between different observations.

This assumption about the variance is used in cases of heteroscedasticity. Moreover, in the case of time series, WLS is applied to impose the common sense intuition that the expansion waves are strongest when closest to the cause. In contrast, the global model assigns equal weight to all observations ($\lambda = 1$), which implies constant variance (homoscedasticity) and a 'full memory' of the series. The variance-covariance matrix for the global model is:

$$\Sigma_{global} = \sigma^2 I_N$$

The diagonal elements decrease exponentially, indicating higher confidence in more recent observations. The off-diagonal elements are zero, reflecting no assumed covariance between different observations. In contrast, the global model assigns equal weight to all observations, implying constant variance. The variance-covariance matrix for the global model is:

$$\Sigma_{global} = \sigma^2 I_N = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

where $\sigma^2$ is a constant variance and $I_N$ is the $N \times N$ identity matrix. The key difference between the two matrices is the time-varying nature of the local model's variances, which allows the model to adapt more quickly to recent changes, whereas the global model treats all observations as equally informative regardless of their position in time.

## 4.2 Plot: $\lambda$-weights vs. time

The following figure shows how the weight has an exponential increasement that goes from almost zero to the oldest observation, to one on the earliest. This means that the last observation has the full impact weight compared to OLS, while all the rest keeps reducing until nothing.

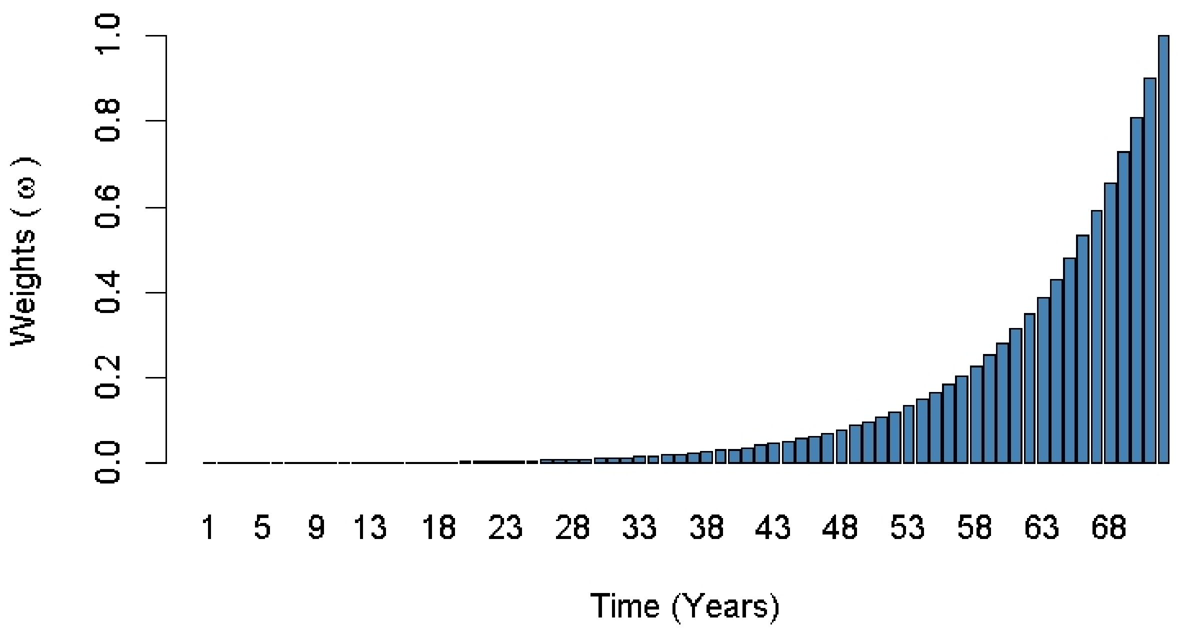



Figure 6: $\omega_j$ vs. time in train set.

## 4.3 Compare the sum of weights ($\omega_i$) in WLS and OLS

In the local model, each observation is assigned a weight that decays exponentially with its longevity. Specifically, the weights are defined as $\omega_j = \lambda^j$ for $j = 0, 1, \ldots, N-1$. Therefore, the total weight is given by

$$T = \sum_{j=0}^{N-1} \lambda^j = \sum_{j=0}^{N-1} \omega_j.$$

For example, when $\lambda = 0.9$, we obtain $T \approx 9.994925$, indicating that the effective total weight is significantly lower than the actual number of observations. In contrast, in an OLS model each observation has a weight of 1, yielding $T = N = 72$. The variance estimator in the WLS model is defined as

$$\hat{\sigma}^2 = \frac{(Y - X_N \hat{\theta_N})^T \, \Sigma^{-1} \, (Y - X_N \hat{\theta_N})}{T - p},$$

where $Y$ is the vector of observations, $X$ is the design matrix, $\hat{\theta}$ is the vector of estimated parameters, and $p$ is the number of parameters.

Since in the WLS model $T$ is lower than $N$, the effective number of observations is reduced, leading to a higher estimated error variance. This is because older observations are down-weighted, thereby reducing the overall informational content. Moreover, this imposes a lower bound on $\lambda$ such that $T > p$, ensuring that there are enough effective observations for a reliable parameter estimation.

## 4.4 Parameters in WLS

The estimated parameters for the OLS and WLS models are presented in Table 2. The WLS model is estimated using the following equation:

$$\hat{\theta}_{WLS} = \left( X^\top \Sigma^{-1} X \right)^{-1} X^\top \Sigma^{-1} y$$

Table 2: Estimated parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ for OLS and WLS models with $\lambda = 0.9$.

| **Model** | $\hat{\theta}_1$ | $\hat{\theta}_2$ |
|---|---|---|
| OLS | 2.9443 | 0.0561 |
| WLS ($\lambda = 0.9$) | 3.0725 | 0.0275 |

## 4.5 Forecast: OLS vs. WLS

Figure 7 shows the confidence interval for the predicted values in the training set for both regression methods. About the training set predicted values, we can see that the uncertainty of the WLS is way higher with respect to OLS; this it due to the information loss implied in weighting less the olders observations.

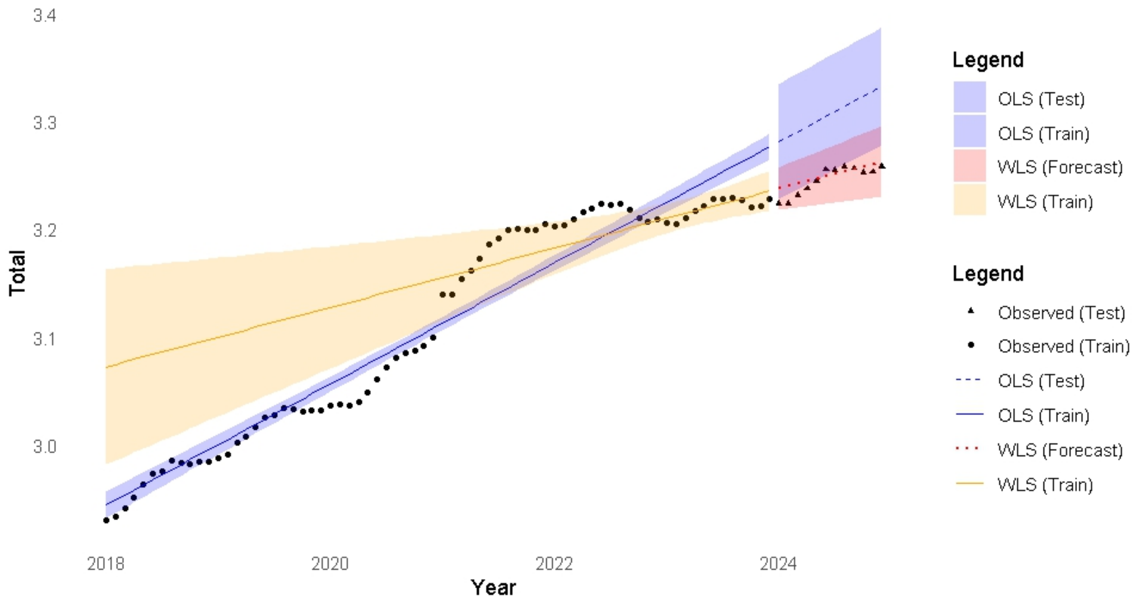



Figure 7: WLS vs. OLS Regression: Training, Forecast, and Test Sets.

Table 3 shows that the WLS model significantly outperforms the OLS model in terms of predictive accuracy. Specifically, the Mean Absolute Error (MAE) for WLS is approximately 0.00688, while for OLS it is 0.06117, indicating that WLS is around 8.9 times more accurate on average. Similarly, the Root Mean Squared Error (RMSE) and the Relative Absolute Error (RAE), are substantially lower for WLS than for OLS.

Table 3: Performance indicators comparison between OLS and WLS predictions.

| **Model** | **MAE** | **MSE** | **RMSE** | **RAE** |
|---|---|---|---|---|
| OLS | 0.06117 | $3.8034 \times 10^{-3}$ | 0.06167 | 5.4325 |
| WLS | 0.00688 | $6.7448 \times 10^{-5}$ | 0.00821 | 0.6112 |

These differences suggest that the use of WLS, by accounting for the temporality relevance of the data, greatly enhances the predictive performance in the test set. The WLS model proves to be considerably more powerful than the OLS model, yielding predictions with errors that are almost 9 times smaller on average. So WLS would be the right choice with respect to OLS.

## 4.6 Optional: varying $\lambda$

In our analysis, we repeated the estimation of parameters and produced 12-month forecasts for several values of $\lambda$. Recall that the effective number of observations in this weighted framework is given by $T = \sum_{j=0}^{N-1} \lambda^j$. When $\lambda$ is very low (around 0.7 or lower), $T$ approaches the number of parameters $p$, causing the residual variance (computed as RSS divided by $T - p$) to collapse due to an insufficient effective sample size.

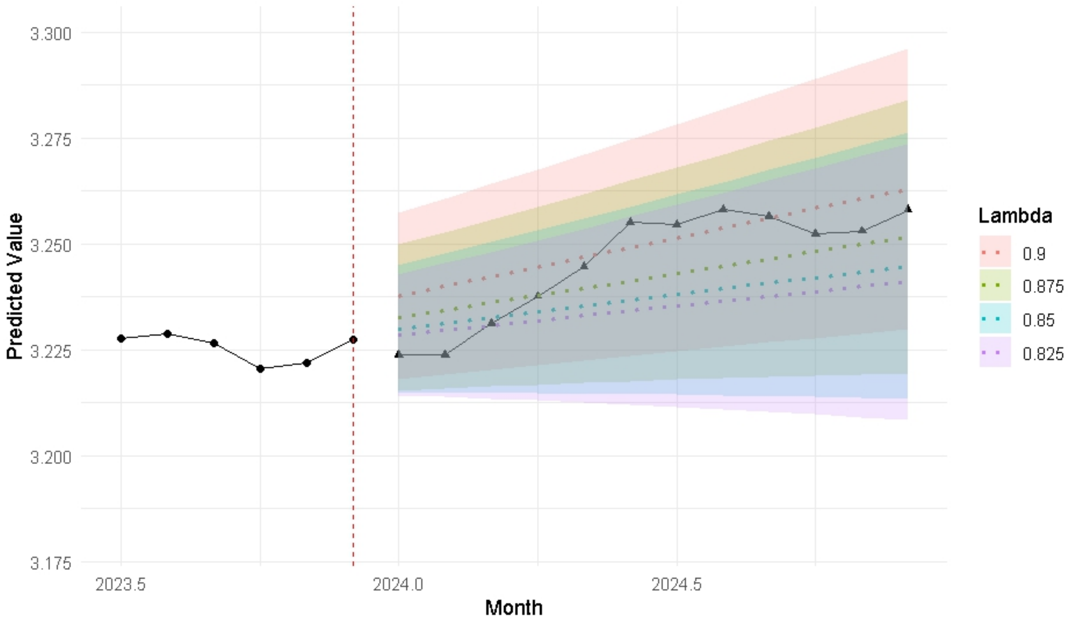



Figure 8: 12-Month Forecasts for Different $\lambda$ Values.

Our simulations were conducted around a narrower interval around the baseline of $\lambda = 0.9$. We observed that higher $\lambda$ values (e.g., 0.925 or 0.95) tend to elevate the forecast without increasing its precision, while conversely lower $\lambda$ values (below 0.85) produce lower forecast values.

Overall, the best performance appears to be achieved for $\lambda$ values between approximately 0.875 and 0.9, offering a balanced trade-off between responsiveness to recent changes and stability, as evidenced by the forecast curves in Figure 8.

# 5 Recursive estimation and optimization of $\lambda$

## 5.1 Update equations of $R_t$ and $\theta_t$

Assignment 1, Q 4.1

$$R_t = \lambda(t) R_{t-1} + X_t X_t^T \qquad h_t = h_{t-1} + x_t y_t$$

$$\hat{\theta}_t = R_t^{-1} h_t$$

⇒ Without forgetting: $\lambda(t) = 1$ ⇒ remember all training dataset.

First step: $X_1 = [1 \ x_1] = [1 \ 0]$

$$R_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \qquad \theta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$R_1 = \lambda(t) R_0 + X_1 X_1^T$$

$$= 1 \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [1 \ 0]$$

$$= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$h_1 = h_0 + x_1 y_1$$

$$= 0 + [1 \ 0] \ 2.930 = [2.930 \ 0]$$

$$\hat{\theta}_1 = R_1^{-1} h_1 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}^{-1} [2.930 \ 0] = \begin{bmatrix} 2.664 \\ 0 \end{bmatrix}$$

Second step: $X_2 = [1 \ x_2] = [1 \ 0.083]$

$$R_1 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$R_2 = \lambda(t) R_1 + X_2 X_2^T$$

$$= 1 \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} [1 \ 0.083] = \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.10689 \end{bmatrix}$$

$$h_2 = h_1 + x_2 y_2$$

$$= [2.930 \ 0] + [1 \ 0.083] \ 2.934 = [5.865 \ 0.245]$$

$$\hat{\theta}_2 = R_2^{-1} h_2 = \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.10689 \end{bmatrix}^{-1} [5.865 \ 0.245] = \begin{bmatrix} 1.788 \\ 0.119 \end{bmatrix}$$

5.1

$$\hat{\theta}_t = (x_t^T x_t)^{-1} x_t^T y_t = R_t^{-1} h_t$$

$$R_0 = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,1 \end{bmatrix} ; \quad \theta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boxed{\hat{\theta}_t = R_t^{-1} h_t}$$

$$\boxed{h_{t-1} = R_{t-1} \hat{\theta}_{t-1}}$$

$$\boxed{R_t = R_{t-1} + x_t x_t^T}$$

$$\boxed{h_t = h_{t-1} + x_t Y_t}$$

$$\boxed{\hat{\theta}_t = \hat{\theta}_{t-1} + R_t^{-1} x_t (Y_t - x_t^T \hat{\theta}_{t-1})}$$

$$y_{1-3} = \begin{bmatrix} 2,930483 \\ 2,934044 \\ 2,941422 \end{bmatrix} ; \quad X_{1-3} = \begin{bmatrix} 1 & 0,000 \\ 1 & 0,083 \\ 1 & 0,167 \end{bmatrix}$$

① 1st iteration

$$h_0 = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} ;$$

$$R_1 = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$h_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \times 2,930483$$

$$R_1 = \begin{bmatrix} 1,1 & 0 \\ 0 & 0,1 \end{bmatrix}$$

$$h_1 = \begin{bmatrix} 2,930483 \\ 0 \end{bmatrix}$$

$$\hat{\theta}_1 = R_1^{-1} \cdot h_1$$

$$\hat{\theta}_1 = \begin{bmatrix} 0,909 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 2,930483 \\ 0 \end{bmatrix} = \begin{bmatrix} 2,664075 \\ 0 \end{bmatrix}$$

② iteration

$$R_2 = R_1 + x_2 x_2^T = \begin{bmatrix} 1,1 & 0 \\ 0 & 0,1 \end{bmatrix} + \begin{bmatrix} 1 & 0,083 \end{bmatrix} \begin{bmatrix} 1 \\ 0,083 \end{bmatrix} = \begin{bmatrix} 2,1 & 0,0833 \\ 0,0833 & 0,10694 \end{bmatrix}$$

$$h_2 = h_1 + x_2^T y_2 = \begin{bmatrix} 2,930483 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0,083 \end{bmatrix} \times 2,934044 = \begin{bmatrix} 5,864527 \\ 0,2445037 \end{bmatrix}$$

$$\hat{\theta}_2 = R_2^{-1} \cdot h_2 = \begin{bmatrix} 0,4913848 & -0,3828973 \\ -0,3828973 & 9,6490108 \end{bmatrix} \begin{bmatrix} 5,864527 \\ 0,2445037 \end{bmatrix} = \begin{bmatrix} 2,7881197 \\ 0,1137072 \end{bmatrix}$$

5.1. RLS.

$$\begin{cases} \hat{\theta}_t = \underbrace{(X_t^T X_t)}_{R_t}{}^{-1} \underbrace{X_t^T y_t}_{h_t} = R_t^{-1} h_t & h_t = h_{t-1} + x_t Y_t \\ R_t = R_{t-1} + x_t x_t^T \\ \hat{\theta}_t = \hat{\theta}_{t-1} + R_t^{-1} x_t (Y_t - x_t^T \hat{\theta}_{t-1}) \end{cases}$$

$$R_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad \theta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 0 \\ 1 & 0.083 \\ 1 & 0.167 \end{bmatrix}, \quad Y = \begin{bmatrix} 2.930 \\ 2.934 \\ 2.941 \end{bmatrix}$$

1st

$$R_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$\left( \begin{aligned} h_1 &= h_0 + [1 \ 0](2.930) \\ &= 0 + [2.930 \ \ 0] = [2.930 \ \ 0] \end{aligned} \right)$$

$$\hat{\theta}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + R_1^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(2.930 - [1 \ 0] \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)$$

$$= \begin{bmatrix} \frac{1}{1.1} & 0 \\ 0 & \frac{1}{0.1} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} (2.930)$$

$$= \begin{bmatrix} 0.9091 \\ 0 \end{bmatrix} (2.930)$$

$$= \begin{bmatrix} 2.662 \\ 0 \end{bmatrix}$$

2nd

$$R_2 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} \begin{bmatrix} 1 & 0.083 \end{bmatrix} = \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.106889 \end{bmatrix}$$

$$\hat{\theta}_2 = \begin{bmatrix} 2.662 \\ 0 \end{bmatrix} + R_2^{-1} \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} \left( 2.934 - \begin{bmatrix} 1 & 0.083 \end{bmatrix} \begin{bmatrix} 2.662 \\ 0 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2.662 \\ 0 \end{bmatrix} + \frac{1}{0.2175} \begin{bmatrix} 0.10689 & -0.083 \\ -0.083 & 2.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} (0.272)$$

$$= \begin{bmatrix} 2.662 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.49 & -0.38 \\ -0.38 & 9.65 \end{bmatrix} \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} (0.272)$$

$$= \begin{bmatrix} 2.662 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.4585 \\ 0.421 \end{bmatrix} (0.272)$$

$$= \begin{bmatrix} 2.787 \\ 0.115 \end{bmatrix}$$

Assignment 1, Q 4.1

$R_t = \lambda(t) R_{t-1} + X_t X_t^T$ $\qquad h_t = h_{t-1} + x_t y_t$

$\hat{\theta}_t = R_t^{-1} h_t$

⇒ Without forgetting: $\lambda(t) = 1$ ⇒ remember all training dataset.

First step: $X_1 = [1 \; x_1] = [1 \; 0]$

$R_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \qquad \theta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$R_1 = \lambda(t) R_0 + X_1 X_1^T$

$= 1 \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [1 \; 0]$

$= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

$h_1 = h_0 + x_1 y_1$

$= 0 + [1 \; 0] \, 2.930 = [2.930 \; 0]$

$\hat{\theta}_1 = R_1^{-1} h_1 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}^{-1} [2.930 \; 0] = \begin{bmatrix} 2.664 \\ 0 \end{bmatrix}$

Second step: $X_2 = [1 \; x_2] = [1 \; 0.083]$

$R_1 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

$R_2 = \lambda(t) R_1 + X_2 X_2^T$

$= 1 \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.083 \end{bmatrix} [1 \; 0.083] = \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.106894 \end{bmatrix}$

$h_2 = h_1 + x_2 y_2$

$= [2.930 \; 0] + [1 \; 0.083] \, 2.934 = [5.865 \; 0.245]$

$\hat{\theta}_2 = R_2^{-1} h_2 = \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.10694 \end{bmatrix}^{-1} [5.865 \; 0.245] = \begin{bmatrix} 2.788 \\ 0.119 \end{bmatrix}$



5.1) Step 1-0

$X_1 = [1; 0] \qquad R_0 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$

$R_1 = \alpha(t) R_0 + X_1 X_1^T$

$= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

$\hat{\theta}_1 = R_1^{-1} h_1 = R_1^{-1} (2.930 \; . \; 0] = \begin{bmatrix} 2.664 \\ 0 \end{bmatrix}$

Step 2

$X_2 = [1, 0.083]$

$R_2 = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{pmatrix} 1 \\ 0.083 \end{pmatrix} [1, 0.083]$

$= \begin{bmatrix} 2.1 & 0.083 \\ 0.083 & 0.1069 \end{bmatrix}$

$\tilde{\theta}_2 = R_2^{-1} [5.885 \; 0.245] = \begin{bmatrix} 2.788 \\ 0.4 \end{bmatrix}$

## 5.2 Update equations of $R_t$ and $\theta_t$, up to t=3

Table 4 presents the $\theta_t$ (coefficients) and $\hat{y}$ (one-step predictions) up to t = 3 without initializing $R_0$, hence there is no results for $\theta_0$ the first time step. The predicted value is for the next time step $\hat{y}_{t+1}$, hence there is no result for the 2nd time step. This is because $x_1$ and $x_2$ are used to predict the next time step, $\hat{y}_3$.

Table 4: Estimates of $\theta_t$ and $\hat{y}$ for one-step predictions.

| $t$ | $\theta_{t,1}$ | $\theta_{t,2}$ | $\hat{y}$ |
|---|---|---|---|
| 1 | - | - | - |
| 2 | 2.930 | 0.043 | - |
| 3 | 2.930 | 0.066 | 2.938 |
| 4 | - | - | 2.946 |

RLS with forgetting is a time-adaptive model where recent data has higher weights compared to older ones. A forgetting factor $\lambda$ ($0 < \lambda < 1$) is in the matrix which exponentially increases with past observations (i.e. updated in matrix $R_t$). RLS with forgetting allows $\theta$ to change over time by accounting for the changing $R_t$ matrix of the weights.

## 5.3 Estimates of $\theta_N$

The final estimates $[\theta_{N,1}, \theta_{N,2}]$ and $[\theta_{OLS,1}, \theta_{OLS,2}]$ are exactly the same $[2.944, 0.056]$. This makes sense because without a forgetting factor $\lambda = 1$, $\theta_N$ is simply the OLS estimate computed using the complete training dataset, so the endpoint would—and must—be identical. Initial values affect the early iterations of RLS, and the algorithm often requires a "burn-in" period because initial parameter estimates may be inaccurate and it takes time for the algorithm to converge to reasonable values. Thus, even though the final RLS estimate $\theta_{N_{RLS}}$ is similar to $\theta_{OLS}$, the estimates from previous iterations would have been different; in fact, it is only possible to obtain a meaningful $\theta_{RLS}$ by the second iteration (using two observations).

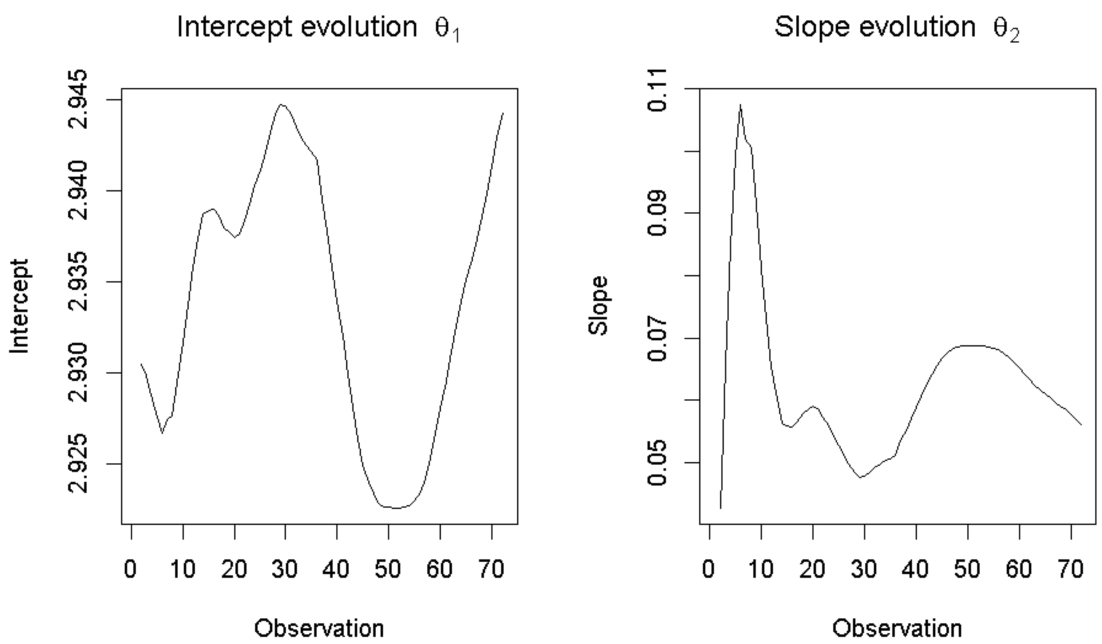



Figure 9: Iterative evolution of RLS regression coefficients: intercept ($\theta_1$) and slope ($\theta_2$) for $\lambda = 1$.

## 5.4 RLS with forgetting

Recursive Least Squares (RLS) with forgetting was implemented using two different forgetting factors: $\lambda = 0.7$ and $\lambda = 0.99$. The parameter estimates $\hat{\theta}_{1,t}$ (intercept) and $\hat{\theta}_{2,t}$ (slope) were computed recursively over time, with the first four points excluded as a burn-in period to mitigate initialization effects. Figure 10 presents the evolution of these parameter estimates. The blue line represents the estimates for $\lambda = 0.7$, while the red line corresponds to $\lambda = 0.99$. Additionally, the orange dot at $t = 72$ indicates the Weighted Least Squares (WLS) estimate for reference.

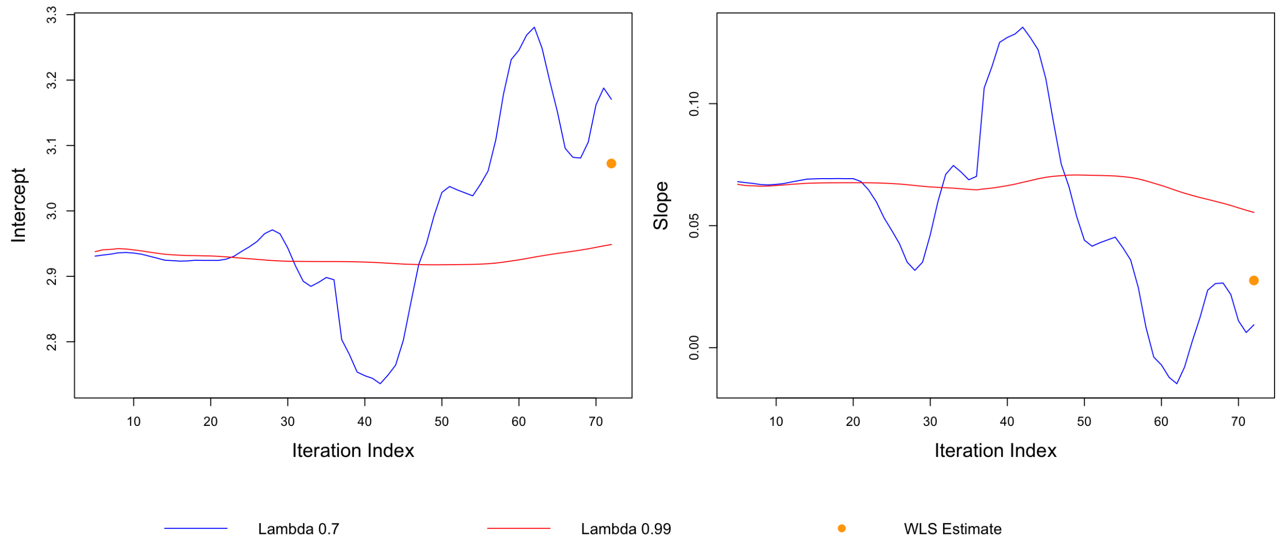



Figure 10: Coefficients for RLS forgetting model for lambda = 0.7 and 0.99

Figure 10 shows that for $\lambda = 0.7$, more recent observations receive higher weights (Iteration index 30 onward), resulting in substantial fluctuations. In contrast, when $\lambda = 0.99$, past observations retain greater influence, leading to a smoother and more stable estimate.

Finally, the WLS estimate, computed with a forgetting factor of $\lambda = 0.9$, logically falls between the RLS estimates for $\lambda = 0.7$ and $\lambda = 0.99$. Since WLS with $\lambda = 0.9$ applies a weighting scheme that is intermediate between these two cases, its estimate is expected to be more adaptive than $\lambda = 0.99$ but more stable than $\lambda = 0.7$.

## 5.5 One-step predictions

The one-step-ahead predictions for the number of motor-driven vehicles in Denmark were obtained using RLS with forgetting factors $\lambda = 0.7$ and $\lambda = 0.99$. Figure 11 shows a comparison of these predictions to the observed data, as well as predictions from WLS and OLS. The observed data points are plotted in black for reference.

Similarly as discussed in section 5.4, the estimate with $\lambda = 0.7$ reacts more quickly to fluctuations in the data, showing greater sensitivity to short-term variations. In contrast, the estimate with $\lambda = 0.99$ exhibits a smoother trajectory, as it places greater weight on past observations, leading to a more stable estimate.

To assess the quality of the one-step-ahead predictions, residuals were computed. Figure 13 displays the residuals for both RLS models across all iterations.

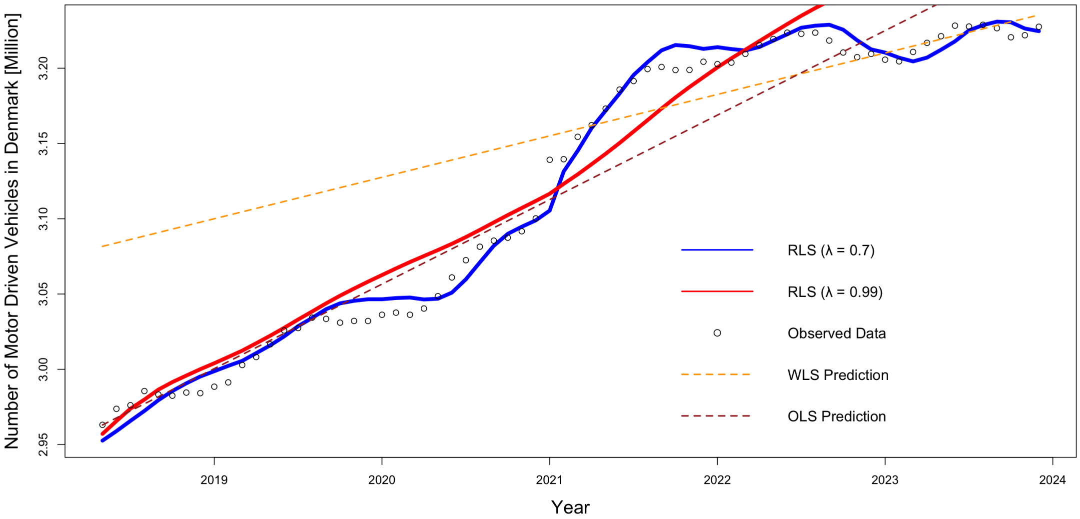



Figure 11: One step predictions compared against different models

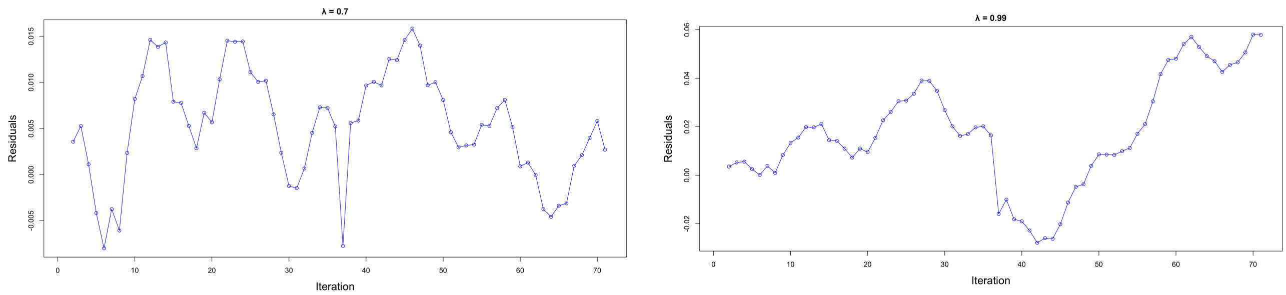



Figure 12: Residuals for One-Step Predictions for Different Lambda Values

The residuals for the $\lambda = 0.7$ model exhibit noticeable fluctuations above zero, rather than oscillating symmetrically around it. This suggests that the model introduces a systematic bias, potentially due to overreacting to recent data points while failing to correct for long-term trends. On the other hand, the residuals for the $\lambda = 0.99$ model start closer to zero but drift upward over time, with larger deviations in later iterations. This suggests that while the higher forgetting factor smooths predictions and reduces short-term noise, it may struggle to adapt quickly to changes, leading to an accumulation of systematic errors.

## 5.6 Optimize forgetting

Analyzing the RMSE curves obtained from the Recursive Least Squares (RLS) model for a sequence of forgetting factors, $\lambda \in [0.1, 0.99]$, reveals a clear pattern. Specifically, the RMSE tends to decrease as $\lambda$ decreases, and shorter forecast horizons yield lower RMSE values. This behavior indicates a trade-off between overfitting and underfitting:

- **Lower $\lambda$ values:** These assign less weight to older observations, increasing the model's responsiveness to recent changes. This is beneficial for short–term forecasts because it avoids overfitting to data that may no longer be relevant. However, with lower $\lambda$ the model may lack stability for longer horizons, as it does not sufficiently incorporate historical trends.

- **Higher $\lambda$ values:** These retain more historical data, which can stabilize long–term predictions by leveraging past information. In non–stationary systems—where the underlying data distribution changes over time—this approach may lead to overfitting, as outdated information can dominate the estimates and reduce the model's ability to quickly adapt to new trends.

Moreover, when considering both $\lambda$ and the forecast horizon together –âas in the contour plotâ– the RMSE surface often shows a bell shaped (or parabolic) pattern, indicating that for a fixed horizon there is an optimal $\lambda$ that minimizes the RMSE. In practice, this suggests that the optimal forgetting factor should depend on the forecast horizon. In other words, one might choose a lower $\lambda$ for shorter horizons and a higher $\lambda$ for longer horizons to achieve the best performance.

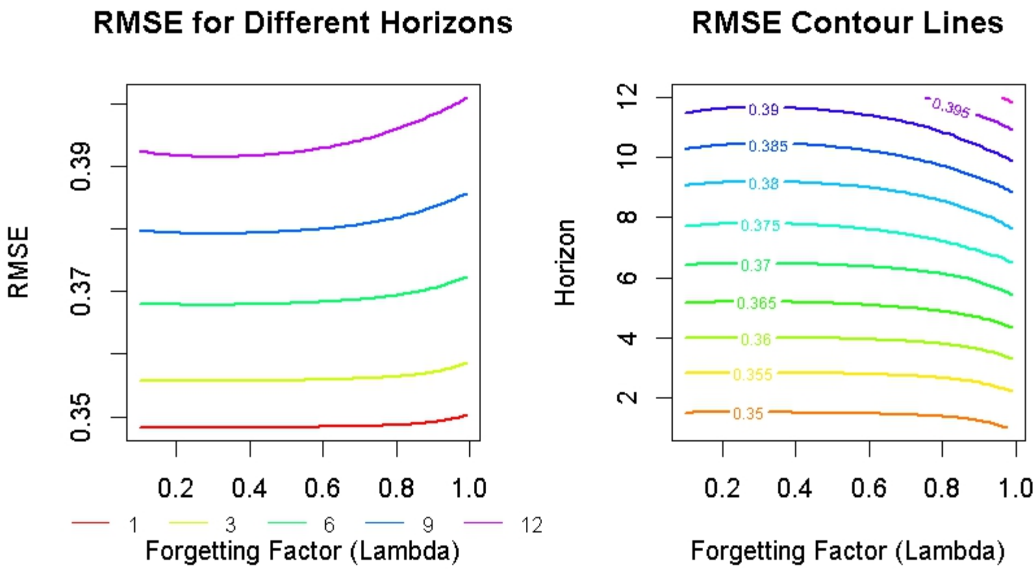



Figure 13: Contour and Line Plots of RMSE vs. Lambda for Different Forecast Horizons

## 5.7 Predictions of test set

As we could expect a lambda of 0.99 seems to behave like a OLS (since we don't allow forgetting), On the other hand a more reasonable lambda we get a result close to what we had with WLS, which make sense since the emphasis is on the end, different lambda make the result change a lot, which also make sense.

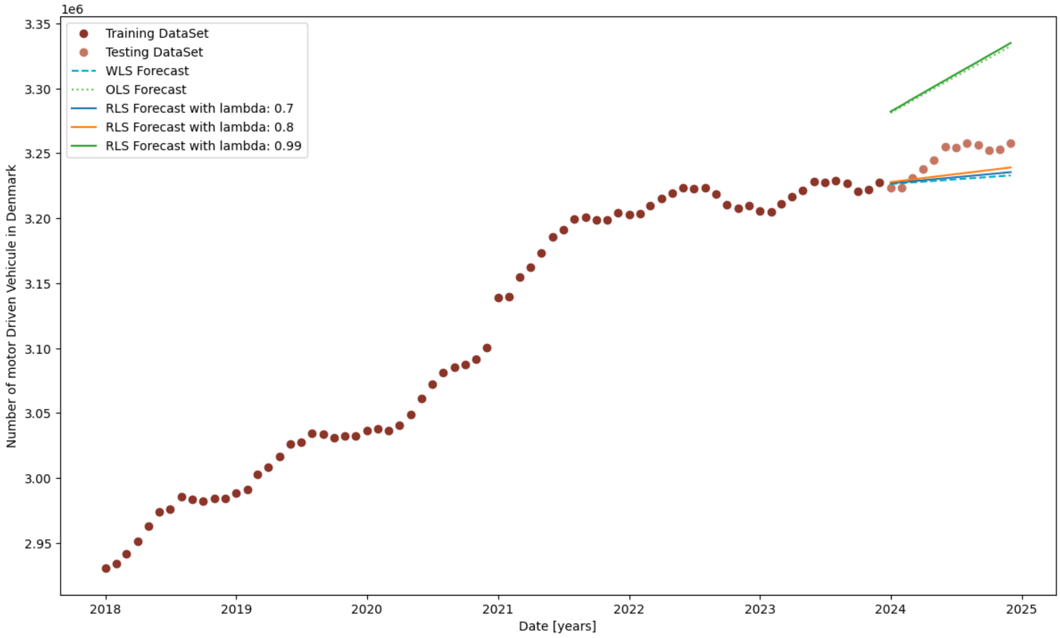



Figure 14: Line Plots of RLS for Different lambda and all horizon compared to OLS and WLS

## 5.8 Reflections

Time-adaptive models such as Recursive Least Squares (RLS) with forgetting offer flexibility in handling changing trends in time series data. However, improper tuning of the forgetting factor can lead to overfitting if too low, making the model sensitive to noise, or underfitting if too high, causing slow adaptation to new trends. An optimal should minimize RMSE across multiple horizons to balance flexibility and stability.

Unlike traditional regression models, time series data is not independent and identically distributed, making test set selection crucial. Data leakage from using future information can artificially improve performance, while assuming stationarity may not hold in practice. Rolling-window validation helps ensure robustness.

Recursive estimation continuously updates parameters as new data arrives, enabling adaptive learning while reducing storage requirements. However, early estimates may be unreliable due to limited data, so excluding initial predictions (burn-in period) can improve accuracy.

An alternative technique is the Kalman filter, which extends RLS by incorporating process noise, making it more robust in dynamic environments. Compared to RLS, Kalman filters can adjust to state changes more effectively.

Time-adaptive models improve forecasting accuracy but require careful tuning to avoid overfitting or underfitting. Recursive estimation methods like RLS and Kalman filters provide flexibility and efficiency but must be evaluated rigorously to ensure reliability in changing conditions.

## List of Figures



## List of Tables