Adaptive forecasting: Implementation (R package forecastADAPT)

Violetta Dalla
National and Kapodistrian University of Athens

Liudas Giraitis
Queen Mary University of London

George Kapetanios
King's College London

Abstract

Forecasting strategies that are robust to structural breaks and other forms of structural change have attracted renewed attention in the literature. These simple approaches are particularly appealing to applied econometricians, as they are based on weighted averages that downweight past information and can adapt in real time. They are easy to implement and remain effective under a wide range of non-stationary environments. Our aim is to introduce the R package **forecastADAPT**, which implements such robust forecasting strategies through a range of illustrative examples and empirical applications. The methodology underlying the package, which covers forecast construction, estimation of forecast errors, and data-driven selection of tuning parameters, is based on the framework developed by Giraitis, Kapetanios, and Price (2013).

Keywords: one-step ahead forecasting, exponential smoothing, tuning parameter selection, R.

1. Introduction

Exponential smoothing is a widely used method for forecasting univariate time series. The method relies on a tuning parameter, which can either be specified by the user or selected optimally. For instance, the function `HoltWinters` in the **stats** (R Core Team 2026) package implements exponential smoothing, with the tuning parameter chosen by minimizing the sum of squared forecast errors. Similarly, the function `ets` in the **forecast** (Hyndman *et al.* 2026; Hyndman and Khandakar 2008) package and the function `es` in the **smooth** (Svetunkov 2026) package perform exponential smoothing within a state space framework, allowing for optimal selection of the tuning parameter.

In these approaches, the full history of the data is typically used to estimate the tuning parameter. However, when the data exhibit structural changes or other forms of non-stationarity, relying on the entire sample may be inappropriate. In contrast, the method proposed in Giraitis, Kapetanios, and Price (2013) provides a theoretical justification for selecting the tuning parameter based on the recent past, thereby allowing the procedure to adapt to changes in the data without requiring prior knowledge of its underlying structure; see also Giraitis, Kapetanios, Mansur, and Price (2015).

The R (R Core Team 2026) package **forecastADAPT** implements this approach through the function `forAD`, which performs adaptive exponential smoothing with data-driven selection of the tuning parameter. Moreover, the method can be combined with AR forecasting to account for remaining dependence in the data. Section 2 describes the forecasting procedure in detail and provides illustrative examples.

2. Adaptive forecasting: `forAD`

In this section we outline the basic theoretical ideas behind the practical implementation of the adaptive forecasting method for a univariate time series $\{x_t\}$, as developed in Giraitis *et al.* (2013). The time series is assumed to have an unspecified and a priori unknown structure: it may be stationary or non-stationary, may include deterministic or stochastic trends, and may undergo structural changes.

The forecasting method consists of two steps. First, the adaptive forecast is used to predict the persistent part of the data and to decompose the data series into a persistent component (signal) and a "stationary" remainder. Second, an AR forecasting method is applied to the stationary remainder by fitting an AR model.

The resulting procedure thus combines adaptive forecasting for the persistent (non-stationary) component with AR forecasting for the component exhibiting stationary dynamics.

Suppose we are given a sample $x_1, ..., x_N$, and we are interested in the forecasting of x_{N+1}.

The adaptive forecast of x_t given $x_1, ..., x_{t-1}$, as developed in Giraitis *et al.* (2013), accounts for the potential structural change by extracting the persistent component of the data. It is computed as the weighted average:

$$\hat{x}_{t|t-1,\rho} = w_{t1,\rho} x_{t-1} + ... + w_{t,t-1,\rho} x_1, \tag{1}$$

where the non-negative weights $w_{tj,\rho}$, $j = 1, ..., t-1$ are parameterized by a tuning parameter ρ and sum up to 1:

$$w_{t1,\rho} + w_{t2,\rho} + ... + w_{t,t-1,\rho} = 1.$$

We use the exponentially decaying weights

$$w_{tj,\rho} = \frac{\rho^j}{\sum_{s=1}^{t-1} \rho^s}, \quad j = 1, ..., t-1$$

with $0 < \rho \leq 1$. A key element in the adaptive forecast (1) of x_t is a data-based selection of the tuning parameter $\hat{\rho}_t$, which is used to compute the forecast $\hat{x}_{t|t-1,\hat{\rho}_t}$. The optimal tuning parameter $\hat{\rho}_t$ as shown in Giraitis *et al.* (2013) can be found by minimizing the mean squared forecast error, $Q_{t,\rho}$, over a training period $t - t_0, ..., t-1$ of length t_0:

$$\hat{\rho}_t = \text{argmin}_{\rho \in (0,1]} Q_{t,\rho}, \quad Q_{t,\rho} = \frac{1}{t_0} \sum_{s=t-t_0}^{t-1} \left(x_s - \hat{x}_{s|s-1,\rho} \right)^2.$$

The package utilizes the `optimize` function from the **stats** (R Core Team 2026) package, where we set for ρ `lower = 0.001` and `upper = 1` in the bounded optimization.

2.1. One-step adaptive forecasting procedure: Adapt

The typical adaptive forecast analysis proceeds as follows. It involves an out-of-sample forecast of x_{N+1} and the evaluation of adaptive forecasts over an evaluation period $(N-E, ..., N]$, where E

is a pre-selected length of the evaluation period.

1. Out-of-sample forecast of x_{N+1}:

For a given sample $x_1, ..., x_N$, the adaptive out-of-sample forecast $\hat{x}_{N+1|N,\hat{\rho}_{N+1}}$ is computed, and the tuning parameter $\hat{\rho}_{N+1}$ is recorded.

2. Forecasts over the evaluation period:

For each t in the evaluation period $t \in (N-E,\, N]$, the adaptive forecasts $\hat{x}_{t|t-1,\hat{\rho}_t}$ are computed, the tuning parameters $\hat{\rho}_t$ recorded, and the errors

$$e_t = x_t - \hat{x}_{t|t-1,\hat{\rho}_t} \tag{2}$$

are obtained.

3. Evaluation of forecast accuracy:

To evaluate forecast quality over $t \in (N-E,\, N+1]$, the mean squared forecast error (MSFE) for the adaptive forecast $\hat{x}_{t|t-1,\hat{\rho}_t}$ is computed as

$$MSFE_t = \frac{1}{n_v} \sum_{s=t-n_v}^{t-1} e_s^2, \tag{3}$$

where the computation for each t is based on the past n_v forecast errors e_s, with n_v fixed and preselected. This produces a series of MSFEs:

$$MSFE_t, \quad t \in (N-E,\, N+1].$$

The package provides both numerical and graphical outputs of $MSFE_t$, which allow comparison and evaluation of different forecasting methods over the evaluation period.

In the package, the methodology described above is referred to as "Adapt", denoted by the superscript $^{(ad)}$, i.e. $\hat{x}_{t|t-1,\hat{\rho}_t} = \hat{x}^{(ad)}_{t|t-1}$.

In sum, given the sample $x_1, ..., x_N$, the adaptive forecast produces:

- The forecast $\hat{x}^{(ad)}_{N+1|N}$,
- The tuning parameter $\hat{\rho}_{N+1}$, used to compute this forecast,
- The MSFE_{N+1}.

In addition, the package provides outputs (plots and numerical values) of these quantities and forecast errors e_t, for the entire evaluation period:

$$\hat{x}^{(ad)}_{t|t-1},\ \hat{\rho}_t,\ e_t,\ MSFE_t, \quad \text{for} \ \ t \in (N-E, N].$$

The adaptive forecast $\hat{x}^{(ad)}_{t|t-1}$ can be interpreted as *signal extraction.* It decomposes x_t into a persistent component (signal) and a noise component:

$$\begin{aligned} x_t &= \hat{x}^{(ad)}_{t|t-1} + e_t, \quad t \in (N-E,\, N], && (4)\\ &=: Adapt_t + u_t, && (5) \end{aligned}$$

where $Adapt_t = \hat{x}^{(ad)}_{t|t-1}$ is the signal, and the noise component $u_t = e_t$ is the forecast error.

In most of the cases, the noise component u_t is approximately stationary and weakly dependent. This series u_t can be further modelled and predicted using an AR process, see Section 2.3.

When the data series x_t exhibits abrupt changes (e.g., breaks in the mean), the persistency and non-stationarity might be not fully captured by $Adapt_t$. In such cases, it may be necessary to perform an additional adaptive forecast on the forecast errors e_t in (4), effectively applying a second-level adapt procedure.

2.2. Two-step adaptive forecasting procedure: Adapt^2

To assess whether a two-step adaptive forecasting procedure is required, the user applies a second adaptive forecast step to the adaptive forecast errors e_t in (4), with the aims to extract any remaining persistency and non-stationarity.

For each $t \in (N-E, N]$, this procedure produces the following outputs:

- the adaptive forecast of the errors, $\hat{e}^{(ad)}_{t|t-1}$,
- the resulting residual $u_t = e_t - \hat{e}^{(ad)}_{t|t-1}$,
- the $MSFE_t$,
- and the tuning parameter $\hat{\rho}_t$.

For $t = N+1$, the procedure provides the out-of-sample forecast $\hat{e}^{(ad)}_{N+1|N}$, along with $\hat{\rho}_{N+1}$ and $MSFE_{N+1}$.

Two step adaptive forecasting procedure allows the decomposition of the data x_t into a persistent signal and a noise component:

$$\begin{aligned} x_t &= \hat{x}^{(ad)}_{t|t-1} + \hat{e}^{(ad)}_{t|t-1} + u_t, \quad t \in (N-E,\, N] \\ &= Adapt^2_t + u_t, \end{aligned} \tag{6}$$

where the two-stage forecast adapt^2, denoted by

$$Adapt^2_t = \hat{x}^{(ad2)}_{t|t-1}, \quad \hat{x}^{(ad2)}_{t|t-1} = \hat{x}^{(ad)}_{t|t-1} + \hat{e}^{(ad)}_{t|t-1},$$

represents the signal, extracted by the two-step adaptive procedure, and u_t is the corresponding forecast error.

The package also provides an out-of-sample adaptive forecast of x_{N+1} given by

$$Adapt^2_{N+1} = \hat{x}^{(ad)}_{N+1|N} + \hat{e}^{(ad)}_{N+1|N},$$

together with the tuning parameter $\hat{\rho}_{N+1}$ and MSFE_{N+1}.

In most cases, the two-step adaptive procedure transforms data x_t into:

- a persistent (potentially non-stationary) signal $Adapt^2_t$, and
- a noise component u_t that is approximately stationary and weakly dependent.

This noise component u_t can then be further modeled using a stationary AR process; see Section 2.4.

In the package, the methodology described in (6) is referred to as "Adapt^2". To determine whether the second adaptive step is necessary, one compares the MSFEs from the one-step and two-step procedures over the evaluation period $(N-E,N]$:

- If, in general,
$$MSFE_t^{(ad2)} \geq MSFE_t^{(ad)},$$
where $MSFE_t^{(ad2)}$ and $MSFE_t^{(ad)}$ denote the MSFEs of the adapt^2 and adapt methods, respectively, then no additional persistence is extracted from e_t, and the second adaptive step is not needed.

- If, on the other hand,
$$MSFE_t^{(ad2)} < MSFE_t^{(ad)},$$
particularly for the out-of-sample forecast at $t = N+1$, then the two-step adaptive forecast should be preferred.

2.3. Combined forecasting procedure: Adapt+AR forecast

This method combines the Adapt and AR forecasting procedures. First, over the evaluation period the adaptive forecasts and corresponding forecast errors, denoted by Adapt$_t$ and u_t, for $t \in (N - E,\, N]$, are computed as in (5).

Next, for each $t \in (N-E,\, N+1]$, an AR(p), $p = 1, 2, ..., p_{max}$, forecast of u_t is obtained:

$$\begin{aligned} u_{t|t-1}^{(ar)} &= \hat{\phi}_0 + \hat{\phi}_1 u_{t-1} + ... + \hat{\phi}_p u_{t-p} \\ &=: \hat{\phi}_{0,t} + \hat{\phi}_{1,t} u_{t-1} + ... + \hat{\phi}_{p,t} u_{t-p} \end{aligned} \tag{7}$$

by fitting the AR(p) model

$$u_j = \phi_0 + \phi_1 u_{j-1} + ... + \phi_p u_{j-p} + \xi_j \tag{8}$$

to the subsample $u_j,\ j \in [t - n_{ar} + p,\, t-1]$, where the subsample size n_{ar} is fixed.

The estimates $\hat{\phi}_{0,t}, ..., \hat{\phi}_{p,t}$ may vary with t.

This leads to the decomposition

$$u_t = u_{t|t-1}^{(ar)} + \varepsilon_t, \quad t \in (N-E, N], \tag{9}$$

where $u_{t|t-1}^{(ar)}$ is the AR forecast and ε_t is the associated forecast error.

The combined forecast of x_t is then given by

$$\hat{x}_{t|t-1}^{(1,ar)} = Adapt_t + u_{t|t-1}^{(ar)}, \quad t \in (N-E, N+1].$$

Accordingly, over the evaluation period, the data can be decomposed as

$$x_t \quad = \quad \hat{x}^{(1,ar)}_{t|t-1} + \varepsilon_t, \quad t \in (N-E,\, N], \tag{10}$$

where ε_t is the forecast error of the combined method, as defined in (9).

The mean squared forecast error (MSFE) of the combined forecast is computed as

$$MSFE_t = \frac{1}{n_v} \sum_{s=t-n_v}^{t-1} \varepsilon_s^2, \quad t \in (N-E,\, N+1], \tag{11}$$

allowing comparison of forecasting performance across methods over the evaluation period.

The package provides outputs of:

- the combined forecast $\hat{x}^{(1,ar)}_{t|t-1}$,
- the forecast error ε_t,
- and the corresponding MSFEs $MSFE_t$,

for $t \in (N-E, N]$.

In addition, the package reports the out-of-sample forecast $\hat{x}^{(1,ar)}_{N+1|N}$ for x_{N+1}, together with its MSFE_{N+1}.

In the package, the methodology described through (10) is referred to as “Adapt+AR”.

2.4. Combined forecasting procedure: Adapt^2+AR forecast

This method combines the Adapt^2 and AR forecasting procedures.

First, the two-step adaptive forecasts $Adapt^2_t$ and the corresponding forecast errors u_t are computed over the evaluation period $t \in (N-E,\, N]$, as in (6).

Next, for each $t \in (N-E,\, N]$, an AR(p), $p = 1, 2, ..., p_{max}$ forecast $u^{(ar)}_{t|t-1}$ of u_t is obtained in the same way as in (7)–(8):

$$u^{(ar)}_{t|t-1} = \hat{\phi}_{0,t} + \hat{\phi}_{1,t} u_{t-1} + ... + \hat{\phi}_{p,t} u_{t-p}.$$

This yields decomposition

$$u_t = u^{(ar)}_{t|t-1} + \varepsilon_t, \quad t \in (N-E, ..., N], \tag{12}$$

where $u^{(ar)}_{t|t-1}$ is the AR forecast of u_t and ε_t is the corresponding forecast error.

The combined Adapt^2+AR forecast of x_t is then given by

$$\hat{x}^{(2,ar)}_{t|t-1} = Adapt^2_t + u^{(ar)}_{t|t-1}, \quad t \in (N-E, N+1].$$

Accordingly, over the evaluation period, the data can be decomposed as

$$x_t \quad = \quad \hat{x}^{(2,ar)}_{t|t-1} + \varepsilon_t, \quad t \in (N-E,\, N], \tag{13}$$

where ε_t is the forecast error of the combined method, as defined in (12).

The mean squared forecast error (MSFE) of the combined forecast is computed in the same way as in (11) for $t \in (N-E, N+1]$.

The package provides outputs of:

- the combined forecasts $\hat{x}^{(2,ar)}_{t|t-1}$,
- the forecast errors ε_t,
- and the corresponding MSFEs $MSFE_t$,

over the evaluation period $t \in (N-E, N]$.

In addition, for $t = N+1$, the package reports the out-of-sample forecast $\hat{x}^{(2,ar)}_{N+1|N}$ together with its MSFE_{N+1}.

In the package, the methodology described in (13) is referred to as "Adapt^2+AR".

2.5. Selection of the optimal forecast method

The Adapt +AR and Adapt^2+AR procedures for forecasting x_{N+1}, given the sample $x_N, ..., x_1$, provide a comprehensive framework for both prediction and model evaluation.

In particular, the package allows for:

- forecasting the next observation x_{N+1},
- conducting a historical analysis of forecasts over the evaluation period,
- producing numerical and graphical outputs of forecasts, forecast errors, tuning parameters, and MSFEs over the evaluation period $t \in (N-E, N]$,
- extracting and visualising the persistent component of the data, captured by the Adapt_t and Adapt^2_t forecasts,
- identifying the forecasting method that yields the lowest MSFE over the evaluation period and for the out-of-sample forecast at $t = N+1$,
- assessing the adequacy of the selected model by testing for the absence of autocorrelation in the forecast errors (i.e. whether they resemble zero-mean, possibly heteroskedastic, white noise).

The package permits selection of the optimal forecasting method based on the smallest MSFE, both over the evaluation period $(N-E, N]$ and for the out-of-sample forecast at $t = N+1$. This selection is further supported by diagnostic checks on the forecast errors, ensuring that they exhibit no significant autocorrelation over the period $(N - n_{ar}, N]$.

In summary, the package implements four adaptive forecasting methodologies: Adapt, Adapt^2, Adapt+AR, and Adapt^2+AR. The parameters t_0, n_v, and n_{ar} are preset but can also be specified by the user. When the sample size is limited, these parameters are automatically adjusted to their maximum feasible values.

The evaluation period of length E is chosen to be as large as possible given the data and parameter settings, that calculations are feasible adjusting the parameters t_0, n_v and n_{ar} when necessary. The user may visualise either the full evaluation period or selected subperiods, depending on the focus of the analysis.

The function used to compute adaptive forecasts for the time series $\{x_t\}$ is called `forAD`, and is of the form

```
forAD(x, p_max = 3, T0 = 50, n_v = 100, n_ar = 400,
      plots = TRUE, P = 50, PL = 500, p = 1, date_1 = NULL)
```

The input arguments are defined as follows.

The 1st argument, `x`, specifies the data $x_1, ..., x_N$. It can be a univariate numeric time series object (`ts`, `xts`, `zoo`), a numeric vector, or a numeric column of a data frame.

The 2nd argument, `p_max`, determines the maximum order of the AR(p) model, with $p = 1, ..., p_{\max}$. The default is `p_max = 3`, but it can be specified by the user. Setting `p_max = 0` disables AR forecasting.

The 3rd argument, `T0`, is the length t_0 of the training period used for selecting the tuning parameter. The default value is `T0 = 50`, or it can be specified by the user.

The 4th argument, `n_v`, defines the number n_v of past forecast errors used in the computation of the MSFE. The default is `n_v = 100`, or it can be specified by the user.

The 5th argument, `n_ar`, determines the number n_{ar} of observations used for estimating the AR(p) models and computing correlograms. The default value is `n_ar = 400`, or it can be specified by the user.

The 6th argument, `plots`, is a logical indicator controlling whether plots are produced (default: `TRUE`). When enabled, the following graphical outputs are generated over a period of length P:

1. The MSFEs, $MSFE_t$, for $t \in (N - P, N + 1]$, corresponding to the methods:

   adapt, adapt+AR(p), adapt^2, adapt^2+AR(p), for $p = 1, \ldots, p_{\max}$.

2. The tuning parameters $\hat{\rho}_t$, for $t \in (N - P, N + 1]$ for the adapt and adapt^2 methods.

3. Forecast plots over the period $t \in (N - P, N]$, including:

   - adapt forecasts $\hat{x}^{(ad)}_{t|t-1}$,
   - adapt^2 forecasts $\hat{x}^{(ad2)}_{t|t-1}$,
   - adapt+AR(p) forecasts $\hat{x}^{(1,ar)}_{t|t-1}$ for the selected p,
   - adapt^2+AR(p) forecasts $\hat{x}^{(2,ar)}_{t|t-1}$ for the selected p,
   - data x_t.

   These plots are also produced over a longer period of length PL.

4. Forecast errors e_t, for $t \in (N - P, N]$, for the adapt and adapt^2 methods, as well as for adapt+AR(p) and adapt^2+AR(p) for the selected p.

5. Correlograms of the forecast errors e_t, for $t \in (N-n_{ar}, N]$, for all methods (adapt, adapt+AR(p), adapt^2, adapt^2+AR(p)), computed up to $\min(10, n_{ar}-1)$ lags and for $p = 1, \ldots, p_{\max}$. These correlograms are based on the package **testcorr** (Dalla, Giraitis, and Phillips 2025).

The argument `plots` can also be specified as a logical vector of length 5, allowing the user to select which of the above plots to display. When `p_max = 0`, only plots corresponding to the adapt and adapt^2 methods are produced.

The 7th argument, `P`, determines the length of the period displayed in the plots (default: `P = 50`, or it can be specified by the user).

The 8th argument, `PL`, specifies the length of the longer period used in the forecasting plots (default: `PL = 500`), or it can be specified by the user.

The 9th argument, `p`, sets the order of the AR model used in the plots. The default is `p = 1`.

The 10th argument, `date_1`, defines the time index corresponding to the out-of-sample forecast $t = N + 1$. If `date_1 = NULL`, the index is set to `NROW(x) + 1`. For time series objects (`ts`, `xts`, `zoo`), the user should provide the appropriate date in the corresponding format.

An object of class '`forAD`' is returned as a list with the following components:

`for_1`	The one-step ahead forecast $\hat{x}^{(ad)}_{N+1\|N}$, $\hat{x}^{(1,ar)}_{N+1\|N}$, $\hat{x}^{(ad2)}_{N+1\|N}$, $\hat{x}^{(2,ar)}_{N+1\|N}$ for all methods (adapt, adapt+AR(p), adapt^2, adapt^2+AR(p), $p = 1, \ldots, p_{\max}$).
`rho`	The tuning parameters $\hat{\rho}_{N+1}$ for (adapt, adapt^2) methods.
`ar_coef_se`	The estimated coefficients $\hat{\phi}_{0,N}, \hat{\phi}_{1,N}, \ldots, \hat{\phi}_{p,N}$ and their standard errors for the adapt+AR(p), $p = 1, \ldots, p_{\max}$, methods.
`ar_coef_se_sq`	The estimated coefficients $\hat{\phi}_{0,N}, \hat{\phi}_{1,N}, \ldots, \hat{\phi}_{p,N}$ and their standard errors for the adapt^2+AR(p), $p = 1, \ldots, p_{\max}$, methods.
`MSFE`	The MSFE_{N+1} for all methods.
`for_in`	The rolling forecasts $\hat{x}^{(ad)}_{t\|t-1}$, $\hat{x}^{(1,ar)}_{t\|t-1}$, $\hat{x}^{(ad2)}_{t\|t-1}$, $\hat{x}^{(2,ar)}_{t\|t-1}$ over the evaluation period for all methods.
`rho_in`	The rolling tuning parameters $\hat{\rho}_t$ over the evaluation period for (adapt, adapt^2) methods.
`MSFE_in`	The rolling MSFEs over the evaluation period for all methods.
`err_in`	The rolling one-step ahead forecast errors over the evaluation period for all methods.
`data`	The original data $x_1, \ldots, x_N$.

The output of the function `forAD` consists of a collection of tables and plots presenting the aforementioned results.

The package permits the computation of forecasts for all methods (adapt, adapt^2, adapt+AR, adapt^2+AR) when the sample size satisfies $N \geq 25 + 2p_{\max}$. When the user sets $p_{\max} = 0$, forecasts (adapt and adapt^2) can be computed for $N \geq 21$.

Examples

We provide some examples to illustrate adaptive forecasting of a univariate time series $\{x_t\}$ using the function `forAD`.

Example 1

We simulate a time series $\{x_t\}$ of sample size $N = 500$ as

$$x_t = 2 + |\sin(4t/N)| + 0.5w_t,$$

where $\{w_t\}$ is an AR(1) series with parameter $\phi_1 = -0.5$ and i.i.d. N(0,1) innovations. This time series is characterized by a time-varying mean and dependent noise arising from AR(1) process.

We simulate the data for $\{x_t\}$:

```
R> N <- 500
R> set.seed(123)
R> w <- arima.sim(list(order = c(1, 0, 0), ar = -0.5), n = N)
R> x <- 2 + abs(sin(4 * seq.int(1, N) / N)) + 0.5 * w
```

We use the function `forAD` to evaluate adaptive forecasting for $\{x_t\}$:

```
R> library(forecastADAPT)
R> print(forAD(x))
```

The output of the function `forAD` is comprised of 2 sections of tables and a set of 28 plots.

The **1st section** of tables `Adaptive+AR one-step ahead forecast output` contains 3 tables.

The **1st table** presents the forecasts, MSFEs, and relative MSFEs at $N + 1$.

The results for the adapt and adapt+AR methods (for $p = 1, 2, 3$) are as follows:

- adapt forecast $\hat{x}^{(ad)}_{N+1|N} = 2.62$, $\text{MSFE}_{N+1} = 0.39$,
- adapt+AR(1) forecast $\hat{x}^{(1,ar)}_{N+1|N} = 2.70$, $\text{MSFE}_{N+1} = 0.29$,
- adapt+AR(2) forecast $\hat{x}^{(1,ar)}_{N+1|N} = 2.69$, $\text{MSFE}_{N+1} = 0.30$,
- adapt+AR(3) forecast $\hat{x}^{(1,ar)}_{N+1|N} = 2.68$, $\text{MSFE}_{N+1} = 0.30$.

The corresponding results for the adapt^2 and adapt^2+AR(p) are:

- adapt^2 forecast $\hat{x}^{(ad2)}_{N+1|N} = 2.68$, $\text{MSFE}_{N+1} = 0.40$,
- adapt^2+AR(1) forecast $\hat{x}^{(2,ar)}_{N+1|N} = 2.83$, $\text{MSFE}_{N+1} = 0.31$,
- adapt^2+AR(2) forecast $\hat{x}^{(2,ar)}_{N+1|N} = 2.80$, $\text{MSFE}_{N+1} = 0.32$,
- adapt^2+AR(3) forecast $\hat{x}^{(2,ar)}_{N+1|N} = 2.78$, $\text{MSFE}_{N+1} = 0.32$.

The relative MSFEs suggest that adapt+AR methods outperform alternatives, with the adapt+AR(1) providing the best forecasting performance.

The **2nd table** reports the estimated AR(p) coefficients $(\hat{\phi}_{0,N}, ..., \hat{\phi}_{p,N})$ and their standard errors (s.e.) for the adapt+AR(p) forecasts $\hat{x}^{(1,ar)}_{N+1|N}$, for $p = 1, 2, 3$:

- for $p = 1$: $\hat{\phi}_{0,N} = 0.007$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.493$ (s.e.=0.043),
- for $p = 2$: $\hat{\phi}_{0,N} = 0.006$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.493$ (s.e.=0.050), $\hat{\phi}_{2,N} = -0.003$ (s.e.=0.050),
- for $p = 3$: $\hat{\phi}_{0,N} = 0.007$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.490$ (s.e.=0.050), $\hat{\phi}_{2,N} = 0.021$ (s.e.=0.056), $\hat{\phi}_{3,N} = 0.052$ (s.e.=0.050).

The stars indicate significance levels *** (1%), ** (5%), and * (10%). The results show that only the AR(1) coefficient is statistically significant. Based on the significance of the coefficients, the adapt+AR(1) is preferred among the adapt^2+AR methods.

Based on the significance of the coefficients, the adapt+AR(1) is preferable among the adapt+AR methods.

The **3nd table** presents the estimated AR(p) coefficients ($\hat{\phi}_{0,N}, ..., \hat{\phi}_{p,N}$) and their standard errors (s.e.) for the adapt^2+AR(p) forecasts $\hat{x}^{(2,ar)}_{N+1|N}$, for $p = 1, 2, 3$:

- for $p = 1$: $\hat{\phi}_{0,N} = -0.031$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.515$ (s.e.=0.042),
- for $p = 2$: $\hat{\phi}_{0,N} = -0.034$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.540$ (s.e.=0.050), $\hat{\phi}_{2,N} = -0.049$ (s.e.=0.050),
- for $p = 3$: $\hat{\phi}_{0,N} = -0.032$ (s.e.=0.026), $\hat{\phi}_{1,N} = -0.537$ (s.e.=0.050), $\hat{\phi}_{2,N} = -0.049$ (s.e.=0.057), $\hat{\phi}_{3,N} = 0.003$ (s.e.=0.050).

Significance levels are indicated by *** (1%), ** (5%), and * (10%). The results once again show that only the AR(1) coefficient is statistically significant. Based on the significance of the coefficients, the adapt^2+AR(1) is preferred among the adapt^2+AR methods.

The **2nd section** of tables

`Adaptive+AR and adaptive²+AR($p$) one-step ahead forecast output x[N-k+1|N-k]`

contains 3 tables.

The **1st table** shows the relative MSFEs, $MSFE_t/MSFE^{(ad)}_t$, for $t = N - 9, ..., N, N + 1$ and $p = 1, 2, 3$. The first line consists of 1s, indicating that all MSFEs are measured relative to the adapt method. For all $t = N - 9, ..., N, N + 1$, the relative MSFEs for the adapt+AR and adapt^2+AR methods are below 1, while those for adapt^2 are slightly above one, suggesting that this method is not preferable. Overall, the relative MSFEs of adaptive+AR(1) tend to be lower than those of the other adapt+AR specifications ($p = 2, 3$) and adapt^2+AR methods ($p = 1, 2, 3$). Hence, adapt+AR(1) method is preferable over the most recent time period.

The **2nd table** provides the forecasts $\hat{x}^{(ad)}_{t|t-1}$, $\hat{x}^{(1,ar)}_{t|t-1}$, $\hat{x}^{(ad2)}_{t|t-1}$, $\hat{x}^{(2,ar)}_{t|t-1}$ for $p = 1, 2, 3$ and $t = N - 9, ..., N, N + 1$. The first row includes the observed data x_t for $t = N - 9, ..., N$ for comparison. Given that the relative MSFEs of the adapt+AR methods are similar across $p = 1, 2, 3$, it is expected that the corresponding forecasts $\hat{x}^{(1,ar)}_{t|t-1}$ are also very close across these specifications over the most recent time period.

The **3rd table** reports the tuning parameters $\hat{\rho}_t$, $t = N - 9, ..., N, N + 1$ for the adapt and adapt^2 methods. The values lie in the range 0.91-0.93 for the adapt method and 0.96-0.97 for the adapt^2 method during the most recent time period. The values of the tuning parameters of the adapt and adapt^2 methods do not differ much, suggesting that the second stage adapt^2 provides only limited additional adjustment.

The **plots** present graphically the results of the adaptive forecasting methods.

The first 8 plots in **Figure 1** display the MSFEs, $MSFE_t$, $t \in (N-P,\, N]$, for the methods adapt, adapt^2, adapt+AR, adapt^2+AR. The solid dot indicates the value at $t = N+1$:

- the blue line and dot (Adapt) correspond to the MSFEs of the adapt method,
- the red line and dot (Adapt^2) correspond to the MSFEs of adapt^2 method,
- the purple lines and dots (Adapt+AR(p)) show the MSFEs for adapt+AR with $p = 1, 2, 3$,
- the green lines and dots (Adapt^2+AR(p)) show the MSFEs for adapt^2+AR with $p = 1, 2, 3$.

The next 8 plots in **Figure 2** present the forecasts together with the observed data over two plotting horizons: a shorter period $t \in (N-P, N+1] = [451,\, 501]$, and a longer period $t \in (N-PL,\, N+1] = [29, 501]$. In all plots, the dashed black line (Data) represents the observed series x_t, while the blue line correspond to adapt, adapt+AR(p), adapt^2, Adapt^2+AR(p) forecasts (with the selected $p = 1$): $\hat{x}^{(ad)}_{t|t-1}$, $\hat{x}^{(1,ar)}_{t|t-1}$, $\hat{x}^{(ad2)}_{t|t-1}$, $\hat{x}^{(2,ar)}_{t|t-1}$. The red dot indicates the out-of-sample forecast at $t = N+1$.

The adapt forecasts capture the non-constant, sinusoidal mean and are very similar to the adapt^2 forecasts, suggesting that there is no need to apply the adapt procedure twice. However, since the data include an AR(1) component with a negative coefficient, incorporating AR(1) dynamics (adapt+AR(1)) leads to a evident improvement in forecasting performance.

These plots reaffirm the conclusion drawn from the tables, indicating that adapt+AR(1) is the preferred forecasting method.

Figure 3 displays the tuning parameters $\hat{\rho}_t$, for $t \in (N-P,\, N]$, for the adapt and adapt^2 methods, with the solid dot indicating the value at $t = N+1$. The light blue line (and dot) correspond to the adapt method, while the pink line (and dot) correspond to the adapt^2 method.

The next 3 plots in **Figure 4** present the forecast errors for the methods adapt, adapt+AR(p), adapt^2, adapt^2+AR(p) over the period $t \in (N-P,\, N] = (451,\, 500]$, using the selected value $p = 1$. The first panel shows the adapt forecast errors e_t (dashed black line) together with their adapt forecasts $e^{(ad)}_{t|t-1}$ (blue line). The adaptive forecasts of the errors e_t are rather close to zero, indicating that second stage of adapt is not necessary. The second and third panels display the forecast errors for the adapt+AR(1), and adapt^2+AR(1) methods. The two plots are very similar suggesting that a single stage of adaptive forecasting is sufficient.

The final 8 plots in **Figure 5** show the correlograms/autocorrelation functions (up to 10 lags) of the forecast errors for the adapt, adapt+AR(p), adapt^2, adapt^2+AR(p) methods, for $p = 1, 2, 3$. The forecast errors from the adapt and adapt^2 methods exhibit substantial autocorrelation at lag 1. However, when adaptive methods are combined with AR forecasting, the autocorrelation becomes negligible, providing further supporting evidence that incorporating AR(1) dynamics improves forecasting performance.

Overall, the adaptive forecasting method combined with AR dynamics performs very well, tracking the non-constant sinusoidal mean and the negative autocorrelation structure of the data.

We have the following outputs:

```
-----------------------------------------------------------
Adaptive+AR and adaptive²+AR one-step ahead forecast output
-----------------------------------------------------------

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

|             | Forecast| MSFE| Relative MSFE|
|:------------|--------:|----:|-------------:|
|Adapt        |     2.62| 0.39|             1|
|Adapt+AR(1)  |     2.70| 0.29|          0.75|
|Adapt+AR(2)  |     2.69| 0.30|          0.75|
|Adapt+AR(3)  |     2.68| 0.30|          0.75|
|------------ | --------| ----| -------------|
|Adapt²       |     2.68| 0.40|          1.02|
|Adapt²+AR(1) |     2.83| 0.31|          0.79|
|Adapt²+AR(2) |     2.80| 0.32|          0.80|
|Adapt²+AR(3) |     2.78| 0.32|          0.81|


AR coefficients and standard errors (adapt+AR)

|      |     |   const|       ar1|     ar2|     ar3|
|:-----|:----|-------:|---------:|-------:|-------:|
|AR(1) |     |   0.007| -0.493***|      NA|      NA|
|      |s.e. | (0.026)|   (0.043)|    (NA)|    (NA)|
|AR(2) |     |   0.006| -0.493***|  -0.003|      NA|
|      |s.e. | (0.026)|   (0.050)| (0.050)|    (NA)|
|AR(3) |     |   0.007| -0.490***|   0.021|   0.052|
|      |s.e. | (0.026)|   (0.050)| (0.056)| (0.050)|

AR coefficients and standard errors (adapt²+AR)

|      |     |   const|       ar1|     ar2|     ar3|
|:-----|:----|-------:|---------:|-------:|-------:|
|AR(1) |     |  -0.031| -0.515***|      NA|      NA|
|      |s.e. | (0.026)|   (0.042)|    (NA)|    (NA)|
|AR(2) |     |  -0.034| -0.540***|  -0.049|      NA|
|      |s.e. | (0.026)|   (0.050)| (0.050)|    (NA)|
|AR(3) |     |  -0.032| -0.537***|  -0.049|   0.003|
|      |s.e. | (0.026)|   (0.050)| (0.057)| (0.050)|
```

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	0.74	0.74	0.74	0.73	0.75	0.74	0.73	0.74	0.75	0.76	0.75
Adapt+AR(2)	0.74	0.74	0.74	0.75	0.74	0.76	0.73	0.75	0.75	0.75	0.75
Adapt+AR(3)	0.74	0.73	0.73	0.74	0.74	0.76	0.74	0.74	0.76	0.75	0.75
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	1.01	1.00	1.00	1.03	1.02	1.04	1.01	1.01	1.04	1.04	1.02
Adapt^2+AR(1)	0.77	0.76	0.76	0.78	0.78	0.79	0.77	0.78	0.80	0.80	0.79
Adapt^2+AR(2)	0.79	0.79	0.79	0.79	0.79	0.81	0.80	0.80	0.81	0.81	0.80
Adapt^2+AR(3)	0.78	0.79	0.80	0.78	0.79	0.80	0.80	0.81	0.81	0.81	0.81

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	2.32	2.40	3.38	2.76	1.95	3.07	2.12	2.01	3.18	2.49	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	2.69	2.66	2.64	2.69	2.70	2.64	2.68	2.63	2.58	2.63	2.62
Adapt+AR(1)	2.61	2.85	2.77	2.34	2.68	3.02	2.48	2.92	2.89	2.34	2.70
Adapt+AR(2)	2.61	2.85	2.77	2.34	2.68	3.02	2.48	2.92	2.89	2.34	2.69
Adapt+AR(3)	2.61	2.87	2.77	2.31	2.68	3.06	2.46	2.89	2.90	2.30	2.68
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	2.80	2.75	2.72	2.80	2.80	2.72	2.76	2.70	2.62	2.70	2.68
Adapt^2+AR(1)	2.65	3.15	3.00	2.07	2.76	3.50	2.35	3.30	3.25	2.08	2.83
Adapt^2+AR(2)	2.64	3.16	3.03	2.07	2.74	3.52	2.38	3.29	3.30	2.10	2.80
Adapt^2+AR(3)	2.65	3.18	3.03	2.04	2.74	3.56	2.38	3.27	3.30	2.05	2.78

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.92	0.92	0.92	0.92	0.92	0.93	0.92	0.92	0.91	0.92	0.91
Adapt^2	0.96	0.96	0.96	0.97	0.97	0.97	0.96	0.97	0.96	0.97	0.96

Selection of the best forecasting method

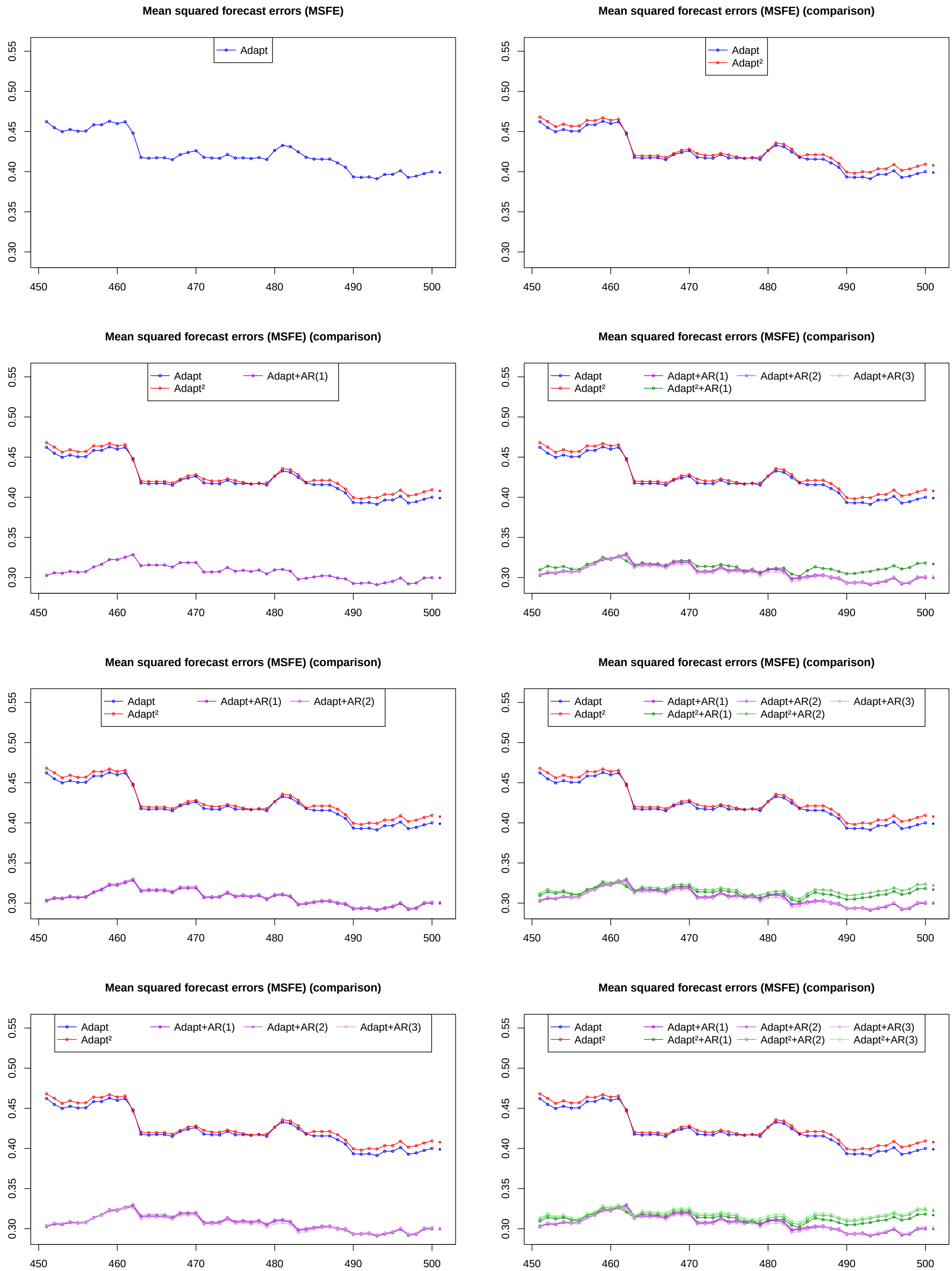

Figure 1: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

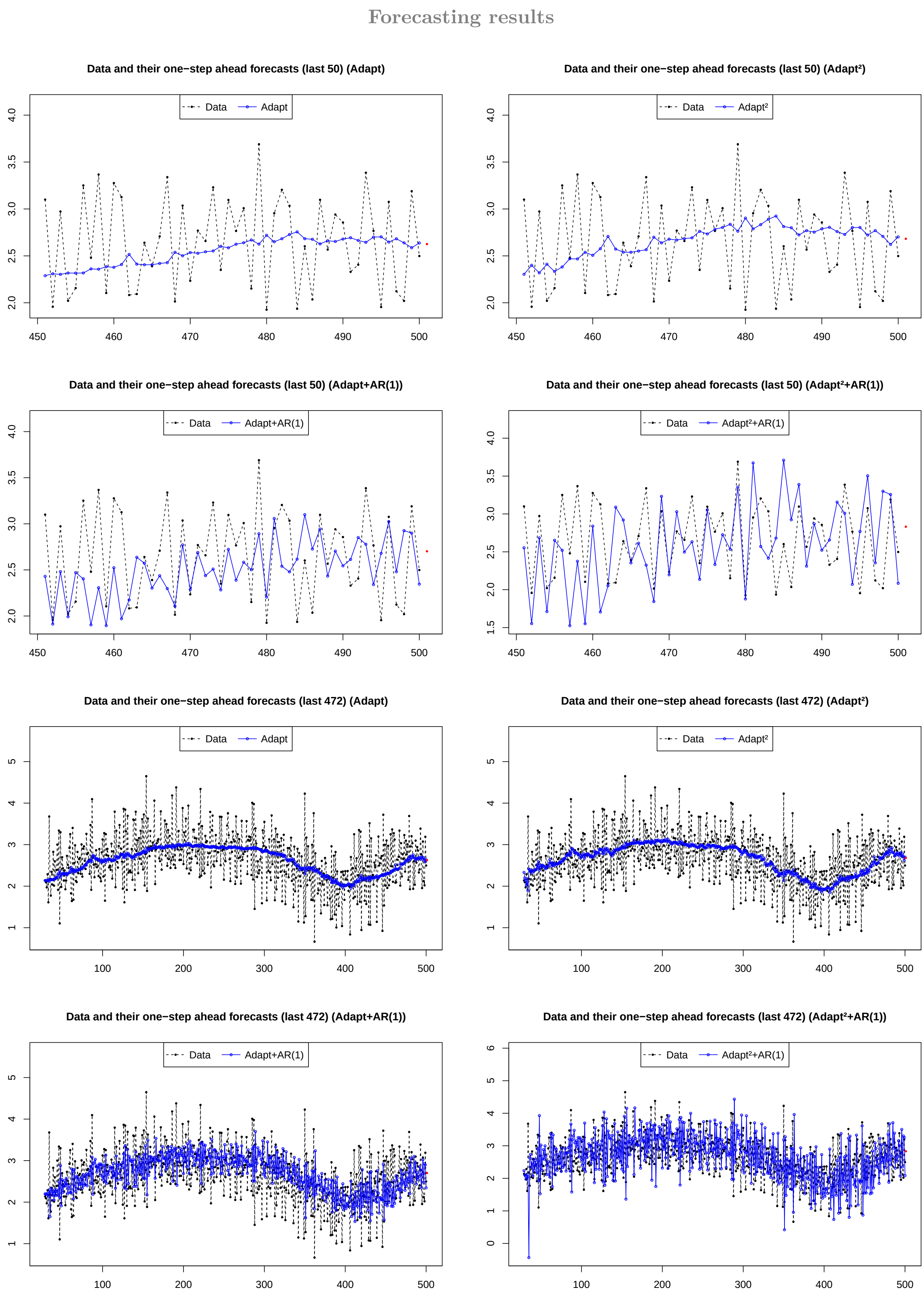

Figure 2: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

Selected tuning parameter

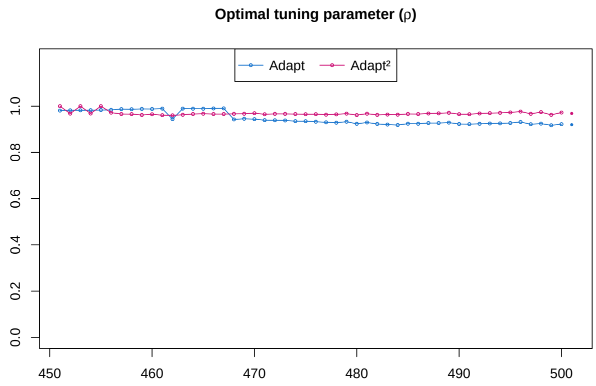

Figure 3: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

Evaluation of forecasting method

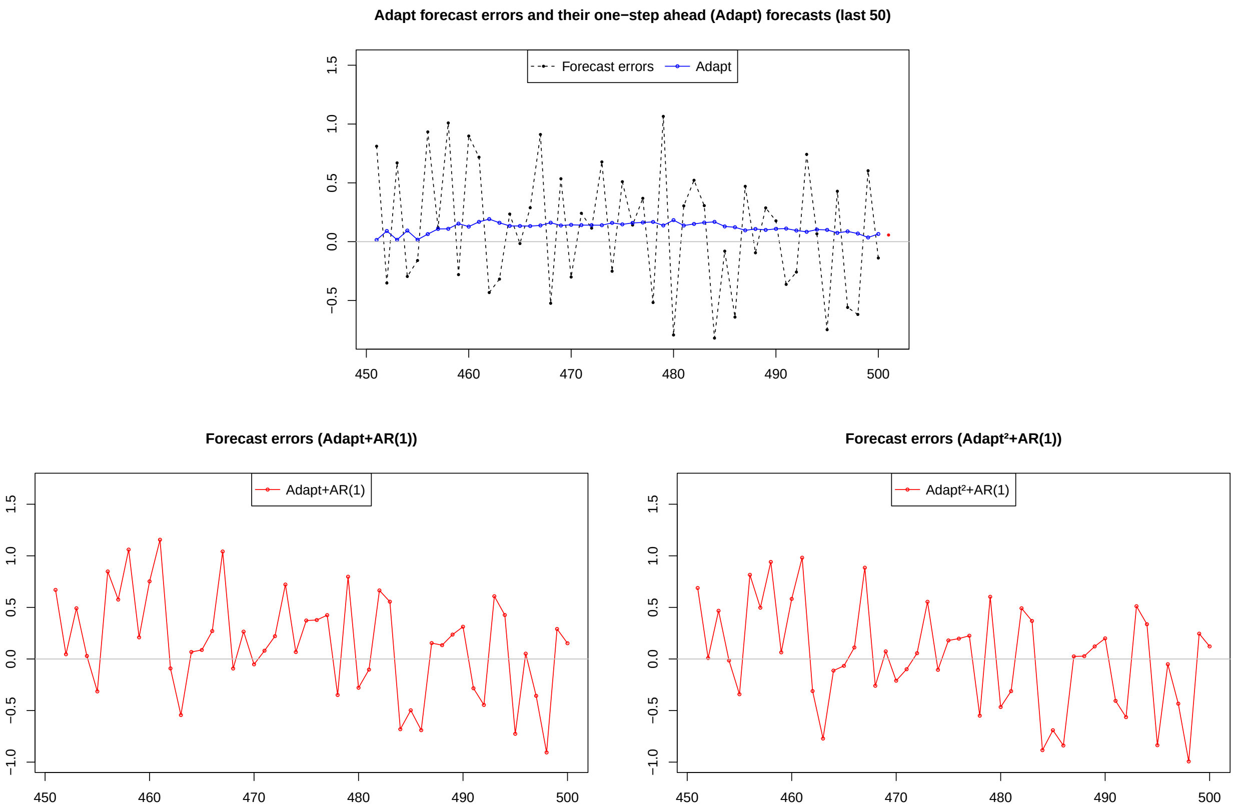

Figure 4: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

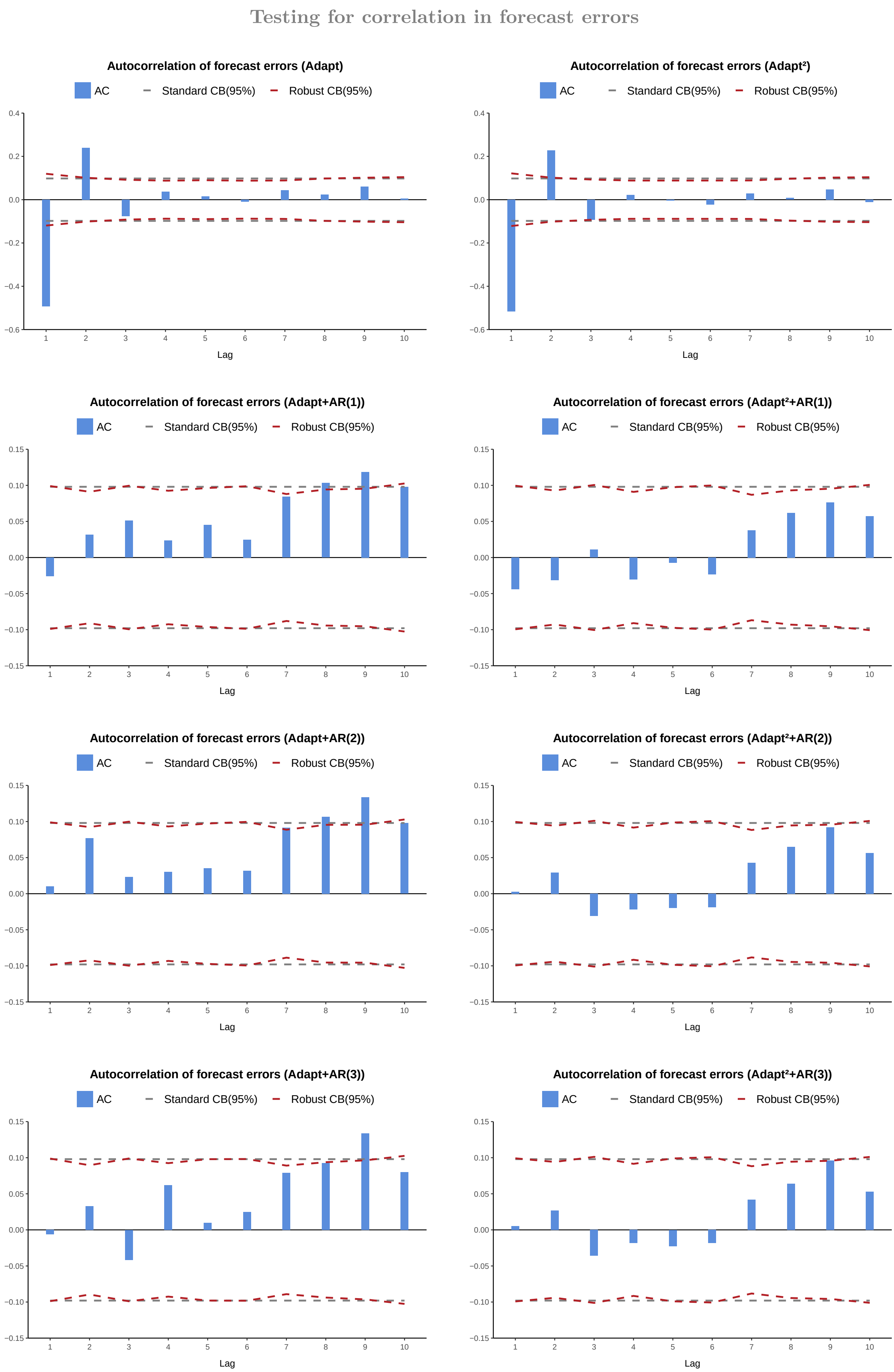

Figure 5: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Example 2

We simulate a time series $\{x_t\}$ of sample size $N = 500$ as

$$x_t = I(t > 0.5N) + I(t > 0.97N) + 0.3u_t,$$

where $I(\cdot)$ is the indicator function and $\{u_t\}$ is an i.i.d. N(0,1) series.

This time series exhibits two structural breaks, in the mean, occurring at $t = 0.5N$ and $t = 0.97N$, while otherwise the observations are independent.

We simulate the data for $\{x_t\}$:

```
R> N <- 500
R> set.seed(123)
R> u <- rnorm(N)
R> x <- c(rep(0, 0.5 * N), rep(1, 0.47 * N), rep(2, 0.03 * N)) + 0.3 * u
```

We use the function `forAD` to evaluate adaptive forecasting for $\{x_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(x))
```

We use this dataset to illustrate the performance of adaptive forecasting methods in the presence of abrupt changes in the mean of a series.

From the tables, we observe that the relative MSFEs (RMSFE) for the adapt+AR, adapt^2, adapt^2+AR forecasts are generally around 1. In some cases they are slightly above 1, and in others slightly below 1. From the plots, see Figure 6, it is clear that the MSFEs of the various methods are not substantially different from each other.

An exception occurs shortly after the break, where the adapt^2 method exhibits a lower MSFE. Nevertheless, the simple adapt method quickly adjusts, and after a few periods its MSFE becomes similar to that of the adapt^2 method. This indicates that adapt forecasting suffices for this series.

This observation is also evident from the tables and the plots, see Figure 7, where the forecast values are very close across the methods, except immediately after the breaks where adapt^2 performs slightly better. The impact of the break on the adapt and adapt^2 forecasts is further reflected in the selected tuning parameters, see Figure 8, which become unstable after the break points. Moreover, applying adapt to the adapt forecast errors has no effect before the break (see top plot in Figure 9), but does show some adjustment immediately after the break, highlighting why adapt^2 briefly outperforms adapt.

The correlograms, see Figure 10, suggest that there is no autocorrelation in the forecast errors for any method. This confirms that AR forecasting is not needed, which is also supported by the AR coefficient tables showing no significant parameters. User can set `p_max = 0`, i.e. run `print(forAD(x, p_max = 0))`, to view only the adapt and adapt^2 outputs.

Overall, the adaptive forecasting performs very well, tracking the non-constant breaking mean and the lack of autocorrelation of the data. Around the break points, the forecasts take few periods to adjust, but considering that the method does not use any break-point detection, the performance is highly satisfactory.

We have the following outputs:

```
---------------------------------------------------------
```

Adaptive+AR and adaptive^2+AR one-step ahead forecast output

```
---------------------------------------------------------
```

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

	Forecast	MSFE	Relative MSFE
Adapt	2.06	0.12	1
Adapt+AR(1)	2.08	0.12	0.99
Adapt+AR(2)	2.08	0.12	0.99
Adapt+AR(3)	2.08	0.12	1.00
------------	--------	----	-------------
Adapt^2	2.18	0.12	1.00
Adapt^2+AR(1)	2.21	0.12	1.01
Adapt^2+AR(2)	2.21	0.12	1.02
Adapt^2+AR(3)	2.21	0.13	1.03

AR coefficients and standard errors (adapt+AR)

		const	ar1	ar2	ar3
AR(1)		0.017	0.023	NA	NA
	s.e.	(0.016)	(0.050)	(NA)	(NA)
AR(2)		0.016	0.023	0.018	NA
	s.e.	(0.016)	(0.050)	(0.050)	(NA)
AR(3)		0.017	0.023	0.018	0.001
	s.e.	(0.016)	(0.050)	(0.050)	(0.050)

AR coefficients and standard errors (adapt^2+AR)

		const	ar1	ar2	ar3
AR(1)		0.003	0.011	NA	NA
	s.e.	(0.016)	(0.050)	(NA)	(NA)
AR(2)		0.003	0.011	-0.012	NA
	s.e.	(0.016)	(0.050)	(0.050)	(NA)
AR(3)		0.003	0.011	-0.013	-0.044
	s.e.	(0.017)	(0.050)	(0.050)	(0.050)

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	0.99	1.01	0.97	0.98	1.01	0.99	0.99	0.99	0.98	0.98	0.99
Adapt+AR(2)	0.99	0.99	0.98	0.97	1.01	1.00	1.01	1.01	0.98	0.98	0.99
Adapt+AR(3)	1.02	0.99	0.99	0.98	1.00	1.01	1.00	1.02	1.00	1.00	1.00
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	0.97	0.97	0.95	0.96	1.00	1.01	1.01	1.01	1.02	0.99	1.00
Adapt^2+AR(1)	0.97	0.97	0.96	0.96	1.00	1.02	1.01	1.03	1.03	1.00	1.01
Adapt^2+AR(2)	1.00	0.99	0.96	0.95	1.01	1.04	1.03	1.02	1.01	1.01	1.02
Adapt^2+AR(3)	1.01	1.00	0.97	0.95	1.03	1.05	1.04	1.03	1.02	1.03	1.03

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	1.96	2.42	2.38	1.67	1.73	1.59	2.05	2.04	2.10	2.16	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	1.97	1.97	2.14	2.23	2.03	1.92	1.80	1.89	1.94	2.00	2.06
Adapt+AR(1)	1.99	1.99	2.17	2.26	2.03	1.93	1.80	1.91	1.96	2.02	2.08
Adapt+AR(2)	2.01	1.99	2.17	2.27	2.04	1.93	1.80	1.91	1.97	2.03	2.08
Adapt+AR(3)	2.01	1.98	2.16	2.27	2.03	1.93	1.80	1.91	1.97	2.03	2.08
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	2.31	2.24	2.44	2.53	2.21	2.07	1.91	2.01	2.06	2.13	2.18
Adapt^2+AR(1)	2.34	2.26	2.47	2.56	2.22	2.07	1.91	2.03	2.09	2.15	2.21
Adapt^2+AR(2)	2.36	2.27	2.48	2.56	2.22	2.09	1.91	2.04	2.09	2.16	2.21
Adapt^2+AR(3)	2.34	2.23	2.51	2.58	2.21	2.09	1.95	2.06	2.11	2.15	2.21

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.66	0.65	0.63	0.62	0.65	0.65	0.63	0.65	0.65	0.63	0.64
Adapt^2	0.88	0.91	0.90	0.90	0.94	0.95	0.96	0.96	0.96	0.95	0.95

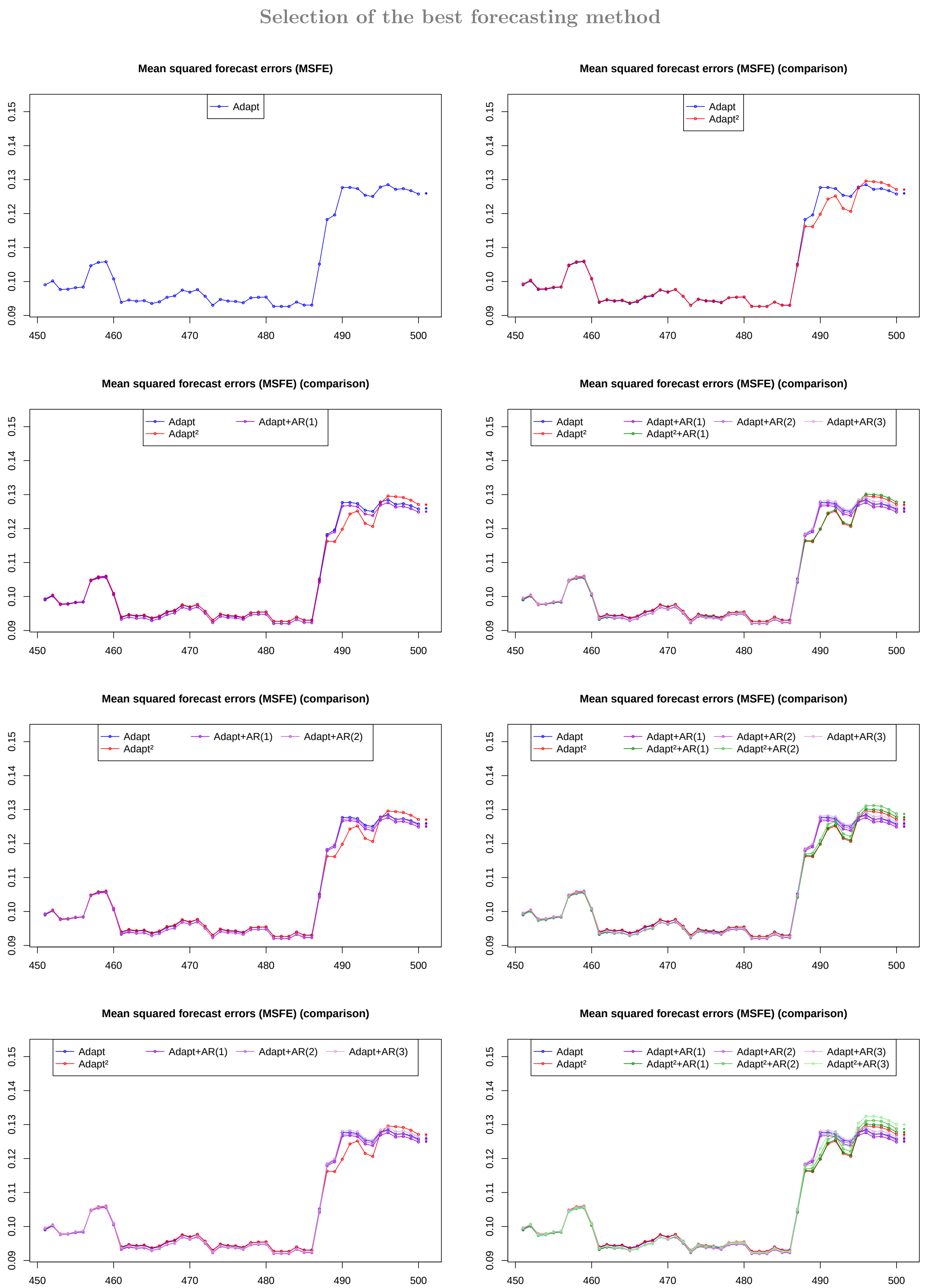

Figure 6: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Forecasting results

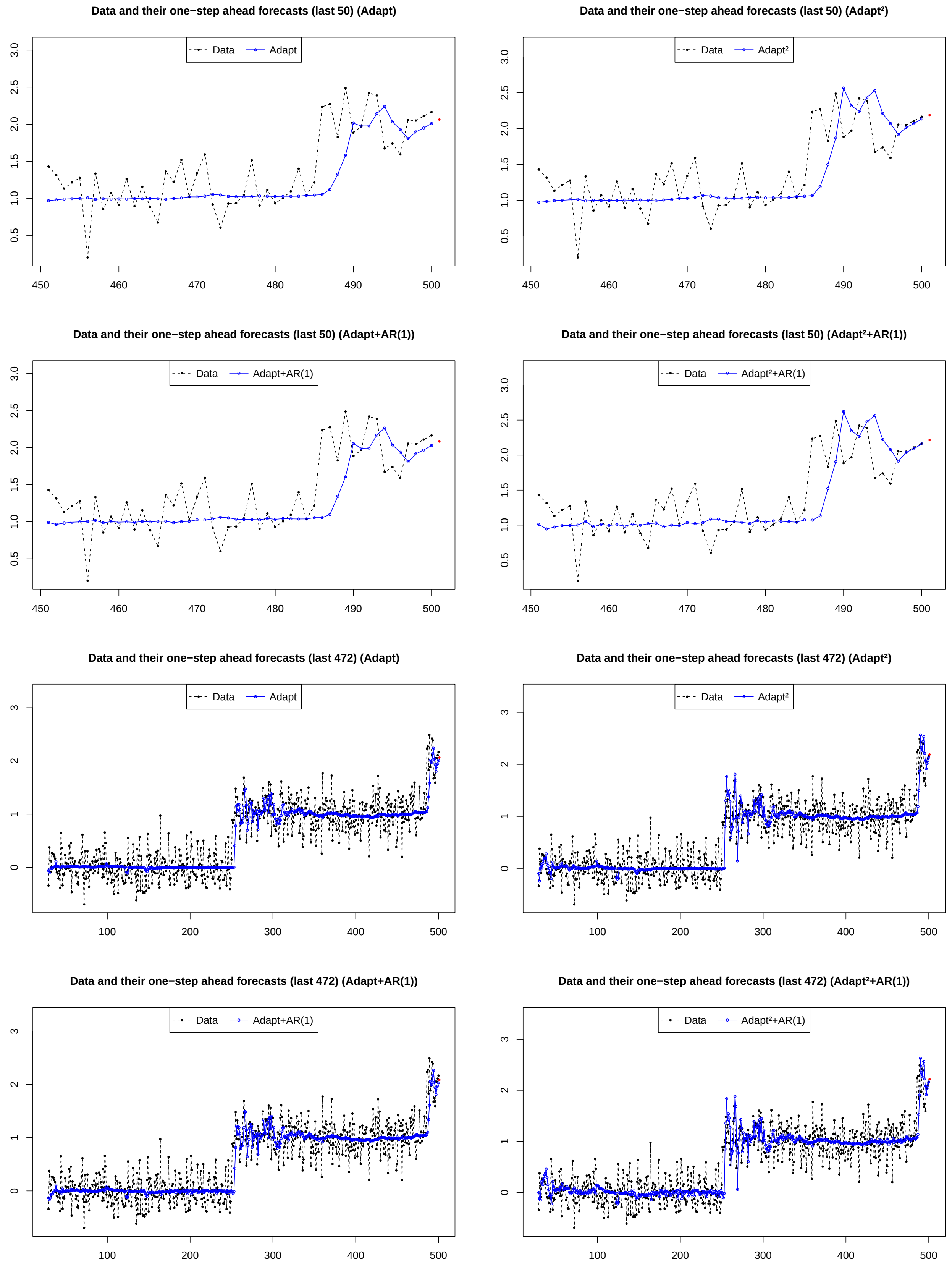

Figure 7: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

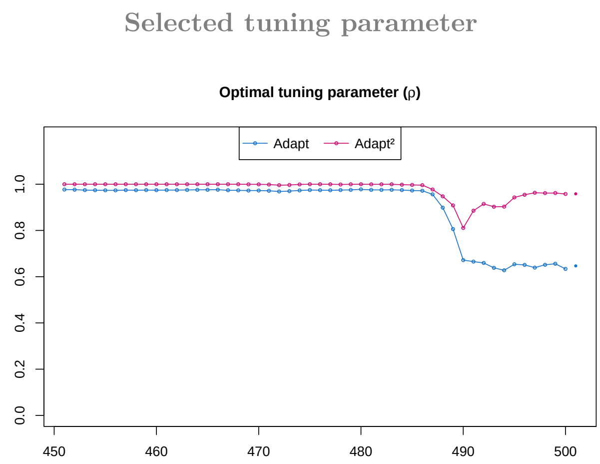

Figure 8: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

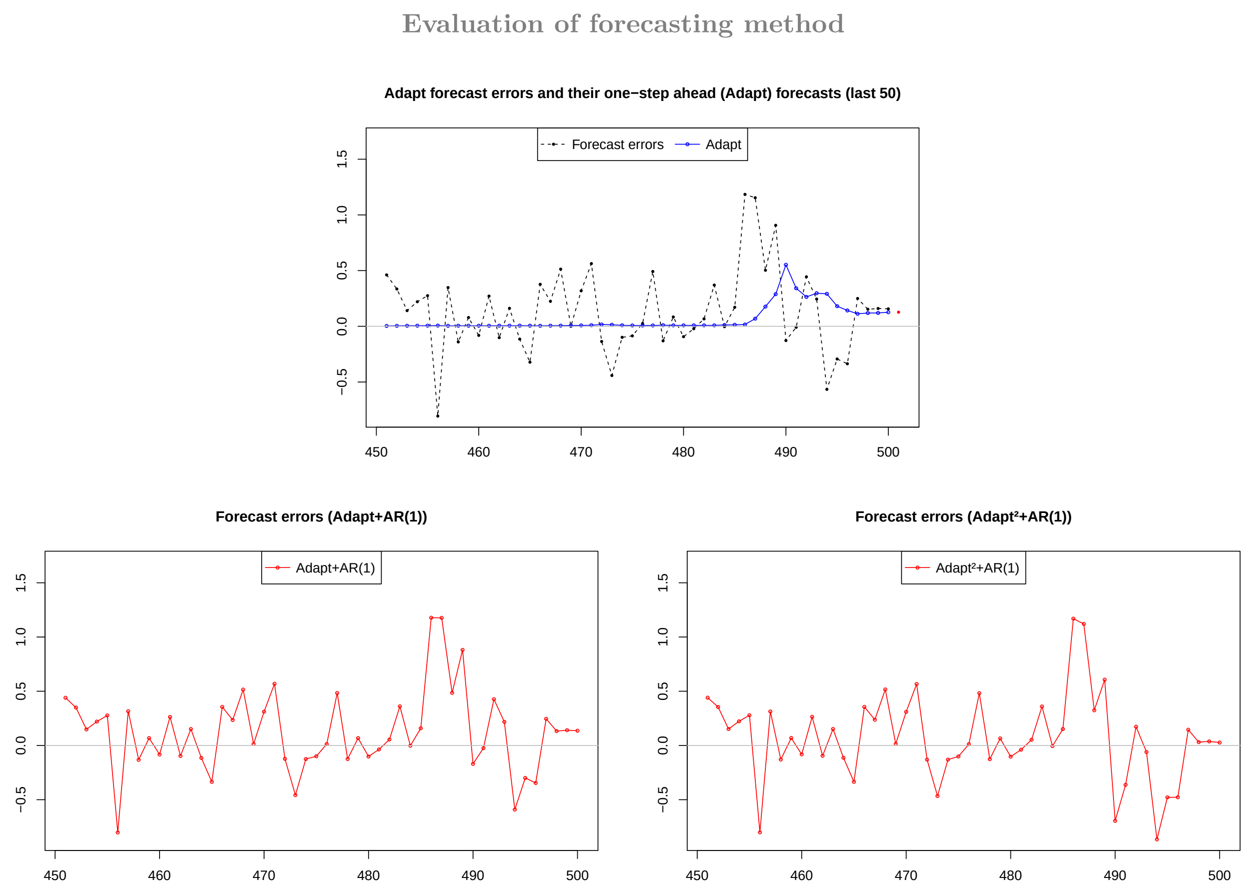

Figure 9: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

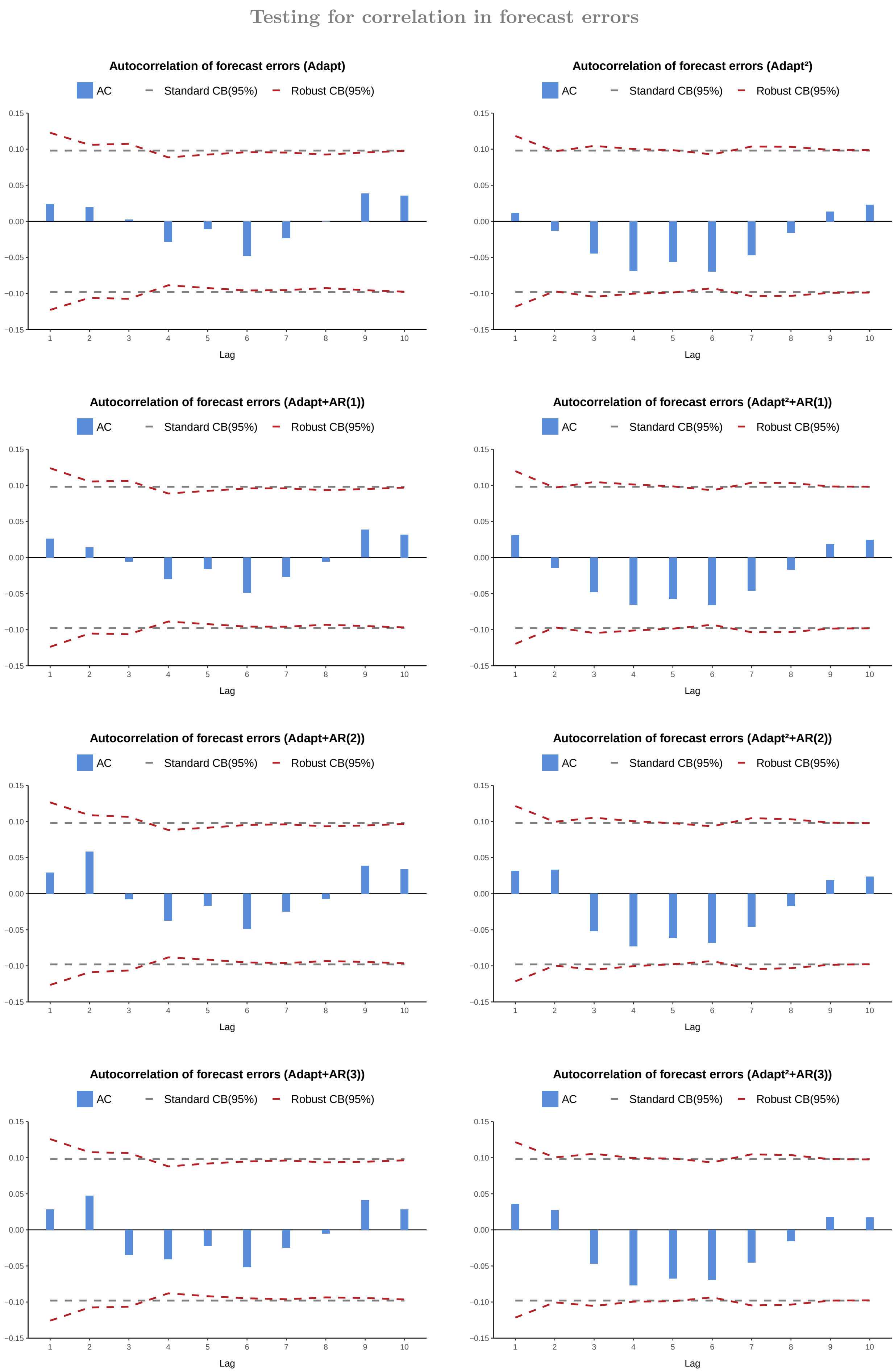

Figure 10: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Example 3

We simulate a time series $\{x_t\}$ of sample size $N = 500$ as

$$x_t = 0.1t + \sum_{s=1}^{t} u_s + w_t,$$

where $\{u_t\}$ is an i.i.d. $N(0,1)$ sequence, $\{w_t\}$ is an AR(1) process with parameter $\phi_1 = 0.5$ and i.i.d. $N(0,1)$ innovations, and $\{u_t\}, \{w_t\}$ are independent. This series combines a linear trend, a unit root component, and a stationary AR(1) process.

We simulate the data for $\{x_t\}$:

```
R> N <- 500
R> set.seed(1234)
R> u <- rnorm(N)
R> set.seed(123)
R> w <- arima.sim(list(order = c(1, 0, 0), ar = 0.5), n = N)
R> x <- 0.1 * seq.int(1, N) + cumsum(u) + w
```

We use the function `forAD` to evaluate adaptive forecasting for $\{x_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(x))
```

From the tables, we observe that the relative MSFEs for the adapt+AR, adapt^2, adapt^2+AR forecasts are generally around 1, in some cases slightly above or below. From the plots, see Figure 11, it is also clear that the MSFEs of the various methods are very similar. These results suggest that adapt forecasting alone suffices. This is further supported by the forecast tables and plots, see Figure 12, where the forecasts across methods are nearly identical.

Moreover, the adapt^2 method is unnecessary. The plot of the selected tuning parameters, see Figure 13, shows that the adapt^2 parameters are very close to 1, indicating that the second-level adapt is effectively just the mean. This is further confirmed when adapt is applied to the adapt forecast errors, see top plot in Figure 14.

The correlograms, see Figure 15, indicate negligible autocorrelation in the forecast errors for all methods, confirming that the AR forecasting is not required. A minor exception is a small autocorrelation at lag 2 in the adapt and adapt^2 forecast errors, corresponding to occasional significance of the AR(2) parameter. However, since the forecasts across methods are largely unchanged, this minor autocorrelation can be safely ignored.

The user can also try setting `p_max = 0`, i.e. running `print(forAD(x, p_max = 0))`, to generate outputs only for the adapt and adapt^2 methods.

Overall, the adaptive forecasting performs very well, tracking the non-constant trending mean, the stochastic trend (unit root), and the positive of autocorrelation present in the data. Although the data contains an AR(1) component with positive autocorrelation, the results suggest that combining adaptive forecasting with AR methods is unnecessary, since the adaptive method adequately tracks both the deterministic trend, stochastic trend and the autocorrelation structure.

We have the following outputs:

```
---------------------------------------------------------------
Adaptive+AR and adaptive²+AR one-step ahead forecast output
---------------------------------------------------------------
```

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

	Forecast	MSFE	Relative MSFE
Adapt	49.6	2.81	1
Adapt+AR(1)	49.8	2.81	1.00
Adapt+AR(2)	49.5	2.75	0.97
Adapt+AR(3)	49.5	2.78	0.98
------------	--------	----	-------------
Adapt^2	49.8	2.83	1.00
Adapt^2+AR(1)	49.9	2.85	1.01
Adapt^2+AR(2)	49.3	2.78	0.99
Adapt^2+AR(3)	49.4	2.81	1.00

AR coefficients and standard errors (adapt+AR)

		const	ar1	ar2	ar3
AR(1)		0.170**	-0.009	NA	NA
	s.e.	(0.080)	(0.050)	(NA)	(NA)
AR(2)		0.190**	-0.011	-0.129**	NA
	s.e.	(0.080)	(0.050)	(0.050)	(NA)
AR(3)		0.197**	-0.012	-0.129***	-0.013
	s.e.	(0.081)	(0.050)	(0.050)	(0.050)

AR coefficients and standard errors (adapt^2+AR)

		const	ar1	ar2	ar3
AR(1)		0.01	-0.01	NA	NA
	s.e.	(0.08)	(0.05)	(NA)	(NA)
AR(2)		0.01	-0.01	-0.12***	NA
	s.e.	(0.08)	(0.05)	(0.05)	(NA)
AR(3)		0.02	-0.01	-0.13***	-0.01
	s.e.	(0.08)	(0.05)	(0.05)	(0.05)

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	1.00	1.00	0.95	0.98	1.03	1.00	1.01	1.02	1.01	1.00	1.00
Adapt+AR(2)	0.99	0.99	0.92	0.96	0.97	0.98	0.97	1.03	0.97	1.00	0.97
Adapt+AR(3)	1.04	0.98	0.94	0.99	0.99	0.98	0.99	1.00	0.98	0.99	0.98
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	1.00	1.01	0.96	1.04	0.98	1.01	1.03	1.02	1.05	1.01	1.00
Adapt^2+AR(1)	1.03	1.00	0.95	1.00	0.98	1.03	1.01	1.01	1.02	1.01	1.01
Adapt^2+AR(2)	1.04	0.98	0.94	0.99	0.98	1.04	0.99	0.99	1.00	1.01	0.99
Adapt^2+AR(3)	1.01	0.99	0.96	0.98	0.97	1.01	0.99	1.02	0.99	1.00	1.00

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	46.6	44.4	48.4	47.2	44.2	44.7	46.4	45.5	47.9	50.2	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	46.2	46.6	44.8	47.5	47.3	45.0	44.8	46.0	45.7	47.3	49.6
Adapt+AR(1)	46.4	46.8	44.9	47.6	47.5	45.2	45.0	46.1	45.8	47.4	49.8
Adapt+AR(2)	46.3	46.5	44.9	47.9	47.1	45.3	45.4	46.2	45.6	47.5	49.5
Adapt+AR(3)	46.3	46.5	44.8	47.9	47.1	45.2	45.4	46.3	45.6	47.5	49.5
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	46.3	46.7	44.9	47.6	47.4	45.2	44.9	46.1	45.8	47.4	49.8
Adapt^2+AR(1)	46.5	46.9	45.1	47.7	47.6	45.4	45.1	46.3	46.0	47.5	49.9
Adapt^2+AR(2)	46.3	46.4	45.0	48.3	46.8	45.5	45.9	46.4	45.6	47.7	49.3
Adapt^2+AR(3)	46.3	46.3	44.9	48.3	47.0	45.4	45.9	46.5	45.6	47.6	49.4

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.17	0.08	0.15	0.24	0.25	0.26	0.27	0.25	0.25	0.25	0.20
Adapt^2	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Selection of the best forecasting method

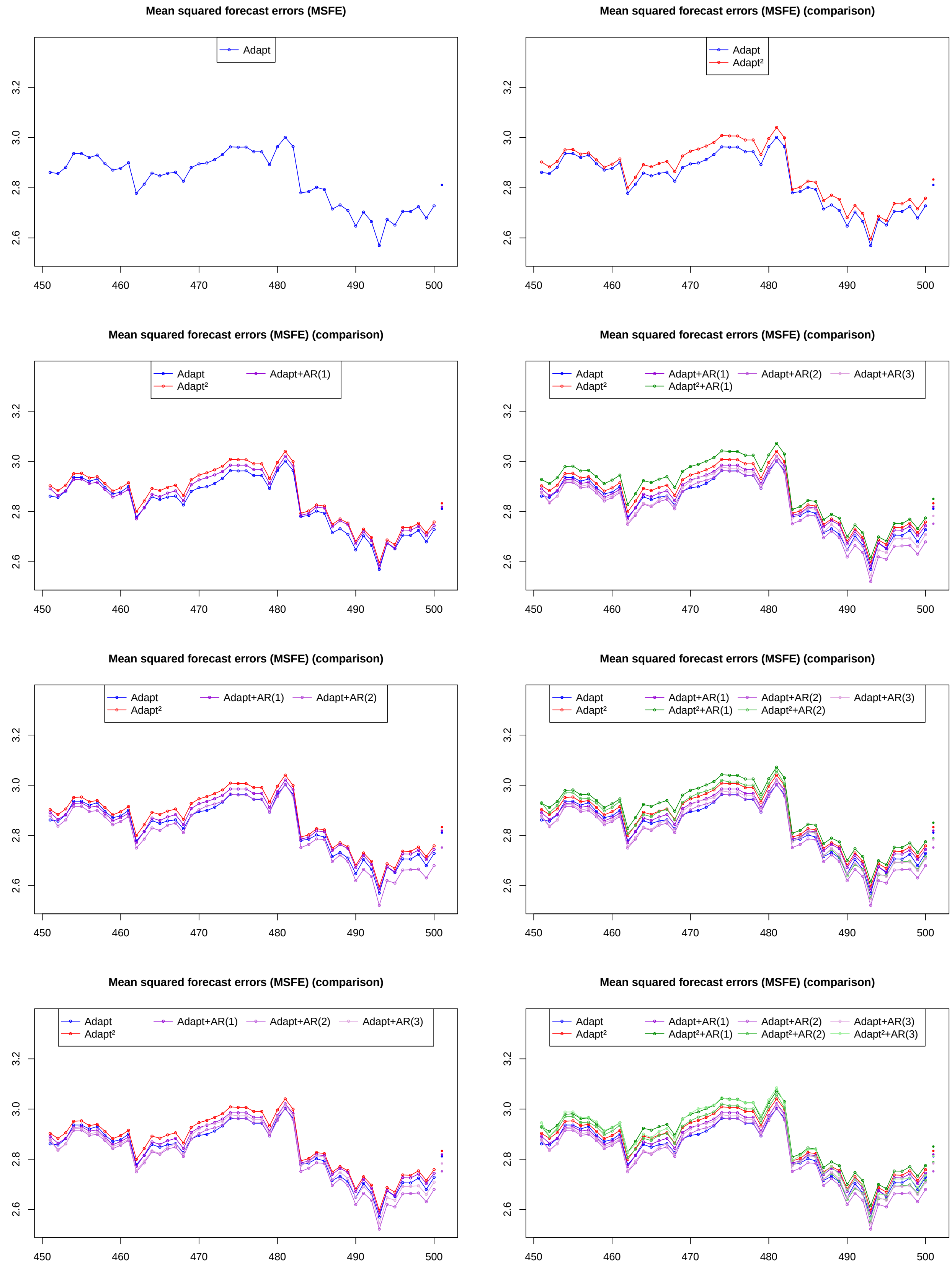

Figure 11: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

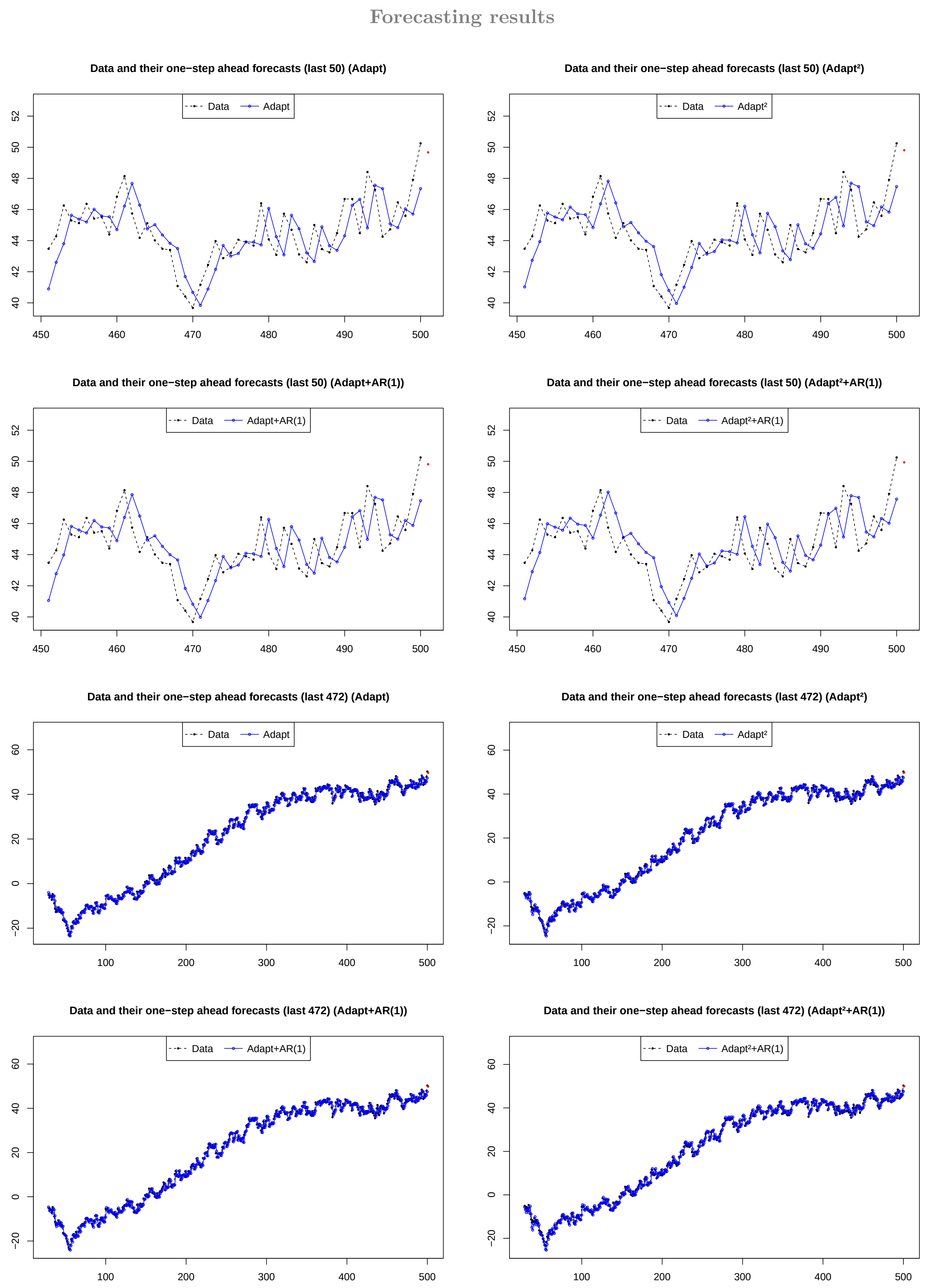

Figure 12: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

Selected tuning parameter

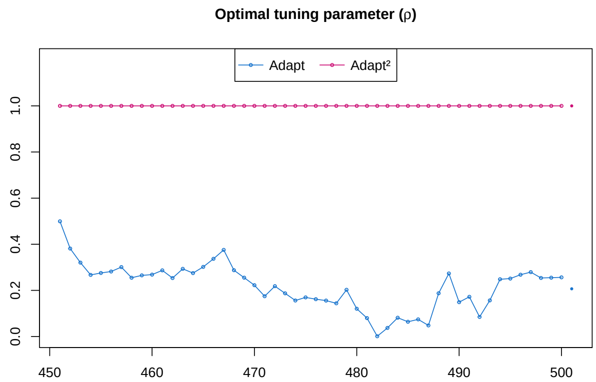

Figure 13: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

Evaluation of forecasting method

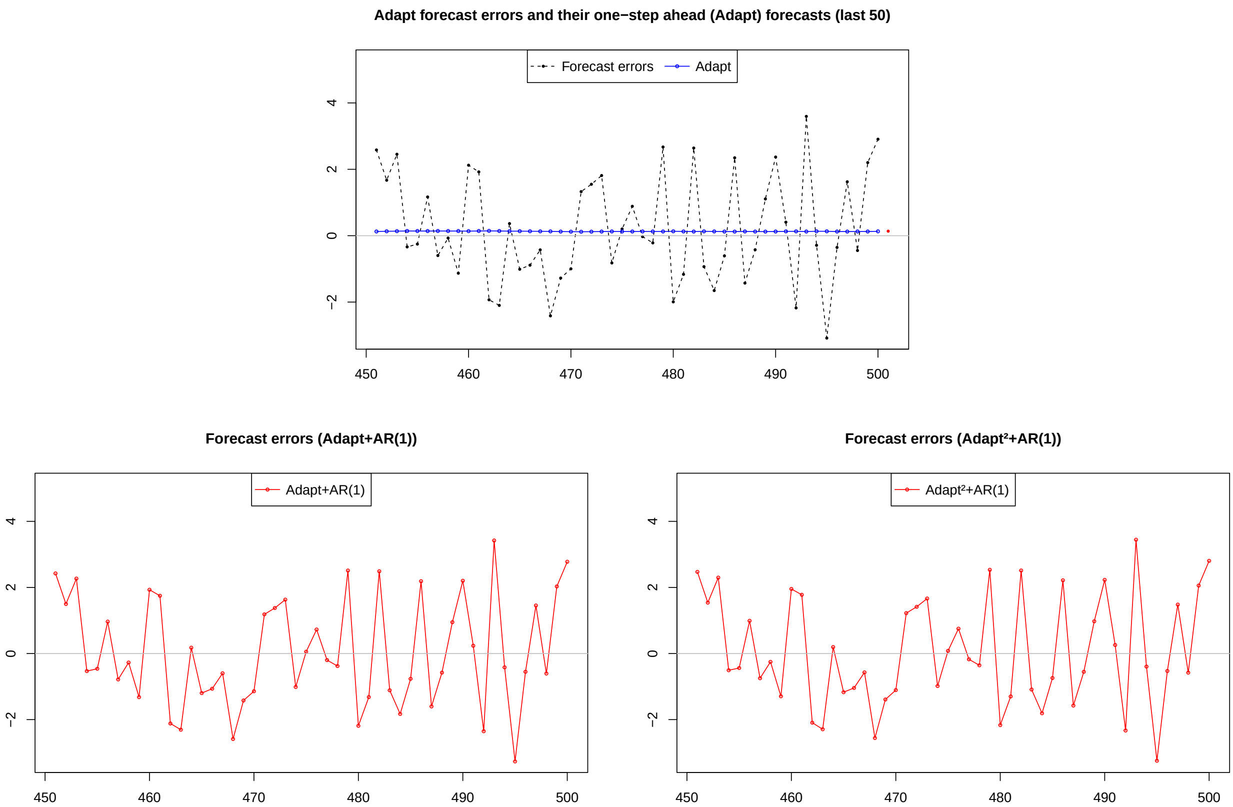

Figure 14: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

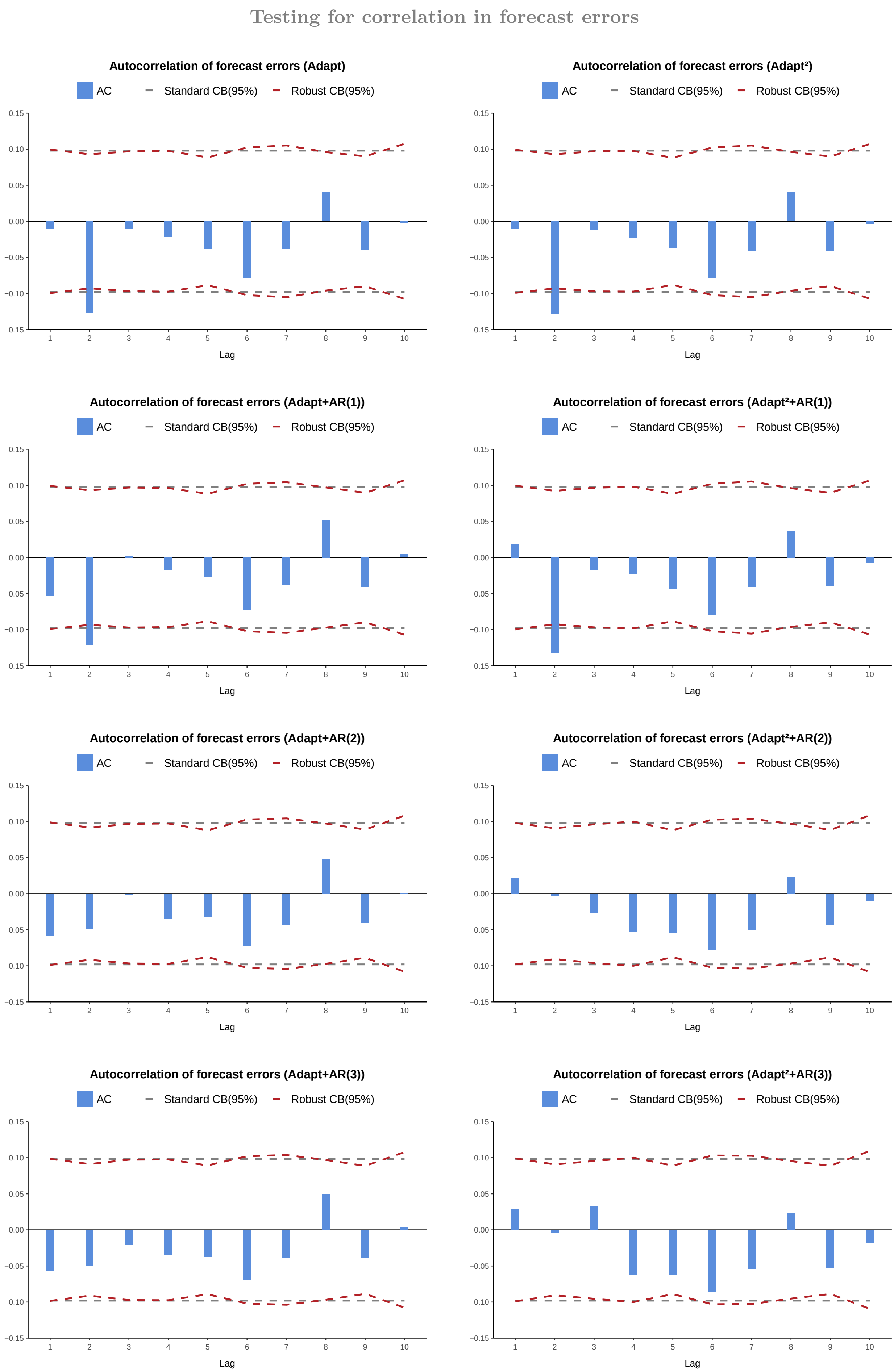

Figure 15: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Empirical example 1

We use the time series of the US monthly year-on-year inflation rate $\{INFL_t\}$ (in %), seasonally adjusted, for the period January 1950 – December 2024 with a sample size $N = 900$ (retrieved on 15/08/2025). We employ the package **quantmod** (Ryan and Ulrich 2025) to retrieve the data from FRED. This package depends on **xts** package (Ryan and Ulrich 2026) for handling time series data, which we also use to construct lagged variables.

```
R> library(quantmod)
R> getSymbols("CPIAUCSL", src = "FRED", return.class = "xts")
R> INFL <- 100 * (CPIAUCSL - lag(CPIAUCSL, k = 12)) / lag(CPIAUCSL, k = 12)
R> INFL <- window(INFL, start = "1950-01-01", end = "2024-12-01")
```

We use the function `forAD` to evaluate adaptive forecasting for $\{INFL_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(INFL, date_1 = "2025-01-01"))
```

From the tables, we see that the relative MSFE of the adapt+AR methods is well below 1. From the MSFE plots (Figure 16) it is clear that adapt+AR outperforms the other methods. Among these, adapt+AR(2) performs marginally better toward the end of the sample compared to adapt+AR(1) and adapt+AR(3). The table of AR coefficients further suggests the use of adapt+AR(2). Nevertheless, the differences among the adapt+AR forecasts are marginal. Based on the adapt+AR(2) method, the forecast of inflation for January 2025 is 2.92%, compared to the true value of 2.99%.

The correlograms (Figure 20) indicate almost no autocorrelation in the forecast errors of adapt+AR(2), while the forecast errors of the adapt method exhibit strong autocorrelation at lag 1, confirming the need for AR adjustment. The user may set `p = 2`, i.e. run `forAD(INFL, p = 2, date_1 = "2025-01-01")`, to visualise the forecasts from the adapt+AR(2) and adapt^2+AR(2) methods.

A second round of adaptive forecasting is not required, as seen from the MSFE table and plots (Figure 16). This is further supported by the forecast plots (Figure 17), where adapt^2 appears more volatile and does not improve upon the adapt forecasts. The tuning parameters for the adapt method (Figure 18) reach the lower bound of the optimisation, hence, the adapt forecast is the last observed value. For the adapt^2 method, there is some deviation of the tuning parameter from the lower bound in the final two years. However, this does not seem to improve the forecast, as the second stage of adapt forecasting applied to the forecast errors (Figure 19) gives values very close to zero in the period of the last two years.

Overall, the adaptive forecasting method performs very well, tracking both the non-constant mean and the autocorrelation structure of the data.

We have the following outputs:

```
------------------------------------------------------------
Adaptive+AR and adaptive²+AR one-step ahead forecast output
------------------------------------------------------------

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

|             | Forecast| MSFE| Relative MSFE|
|:------------|--------:|----:|-------------:|
|Adapt        |     2.87| 0.16|             1|
|Adapt+AR(1)  |     2.93| 0.12|          0.75|
|Adapt+AR(2)  |     2.92| 0.12|          0.74|
|Adapt+AR(3)  |     2.92| 0.12|          0.74|
|------------ | --------| ----| -------------|
|Adapt²       |     2.93| 0.17|          1.01|
|Adapt²+AR(1) |     3.03| 0.17|          1.02|
|Adapt²+AR(2) |     2.95| 0.15|          0.90|
|Adapt²+AR(3) |     2.93| 0.15|          0.93|


AR coefficients and standard errors (adapt+AR)

|      |     |    const|       ar1|        ar2|      ar3|
|:-----|:----|--------:|---------:|----------:|--------:|
|AR(1) |     |  -0.0003| 0.4088***|         NA|       NA|
|      |s.e. | (0.0186)|  (0.0457)|       (NA)|     (NA)|
|AR(2) |     |   0.0001| 0.4891***| -0.2017***|       NA|
|      |s.e. | (0.0183)|  (0.0492)|   (0.0491)|     (NA)|
|AR(3) |     |  -0.0006| 0.5077***| -0.2386***|   0.0776|
|      |s.e. | (0.0182)|  (0.0502)|   (0.0549)| (0.0501)|

AR coefficients and standard errors (adapt²+AR)

|      |     |   const|      ar1|       ar2|      ar3|
|:-----|:----|-------:|--------:|---------:|--------:|
|AR(1) |     |   0.009|  0.127**|        NA|       NA|
|      |s.e. | (0.021)|  (0.049)|      (NA)|     (NA)|
|AR(2) |     |   0.012| 0.167***| -0.310***|       NA|
|      |s.e. | (0.020)|  (0.047)|   (0.047)|     (NA)|
|AR(3) |     |   0.011| 0.135***| -0.292***| -0.108**|
|      |s.e. | (0.020)|  (0.049)|   (0.048)|  (0.049)|
```

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	0.77	0.79	0.79	0.77	0.74	0.75	0.75	0.73	0.76	0.75	0.75
Adapt+AR(2)	0.77	0.79	0.79	0.76	0.75	0.75	0.75	0.73	0.75	0.75	0.74
Adapt+AR(3)	0.77	0.79	0.78	0.75	0.75	0.74	0.73	0.74	0.74	0.74	0.74
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	1.02	1.05	1.05	1.00	1.01	0.98	1.01	1.00	0.99	1.01	1.01
Adapt^2+AR(1)	1.04	1.07	1.07	1.05	1.02	1.00	1.02	1.02	1.02	1.02	1.02
Adapt^2+AR(2)	0.93	0.95	0.93	0.93	0.90	0.88	0.91	0.88	0.91	0.90	0.90
Adapt^2+AR(3)	0.97	0.97	0.95	0.95	0.93	0.92	0.93	0.90	0.93	0.93	0.93

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	3.46	3.35	3.23	2.97	2.93	2.61	2.43	2.57	2.71	2.87	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	3.16	3.46	3.35	3.23	2.97	2.93	2.61	2.43	2.57	2.71	2.87
Adapt+AR(1)	3.18	3.58	3.30	3.18	2.85	2.92	2.47	2.35	2.62	2.77	2.93
Adapt+AR(2)	3.23	3.60	3.23	3.20	2.85	2.97	2.45	2.40	2.67	2.75	2.92
Adapt+AR(3)	3.25	3.59	3.22	3.22	2.84	2.97	2.42	2.41	2.65	2.73	2.92
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	3.12	3.55	3.36	3.20	2.86	2.85	2.42	2.25	2.50	2.72	2.93
Adapt^2+AR(1)	3.17	3.72	3.29	3.14	2.73	2.85	2.26	2.18	2.60	2.81	3.03
Adapt^2+AR(2)	3.27	3.71	3.11	3.22	2.76	2.98	2.21	2.31	2.66	2.71	2.95
Adapt^2+AR(3)	3.25	3.71	3.10	3.20	2.78	2.99	2.23	2.31	2.67	2.69	2.93

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Adapt^2	0.63	0.62	0.61	0.64	0.72	0.67	0.61	0.63	0.65	0.65	0.63

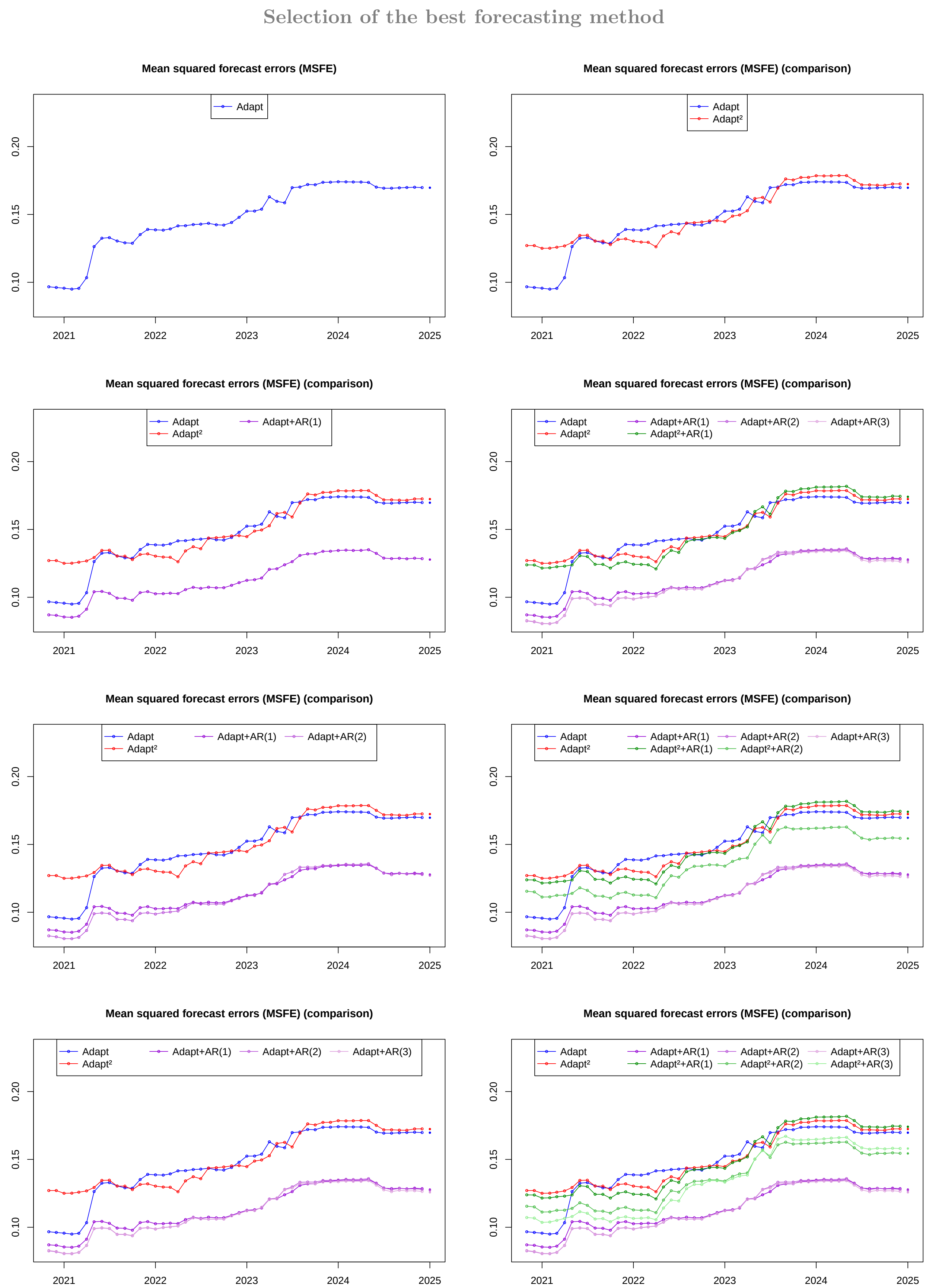

Figure 16: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Forecasting results

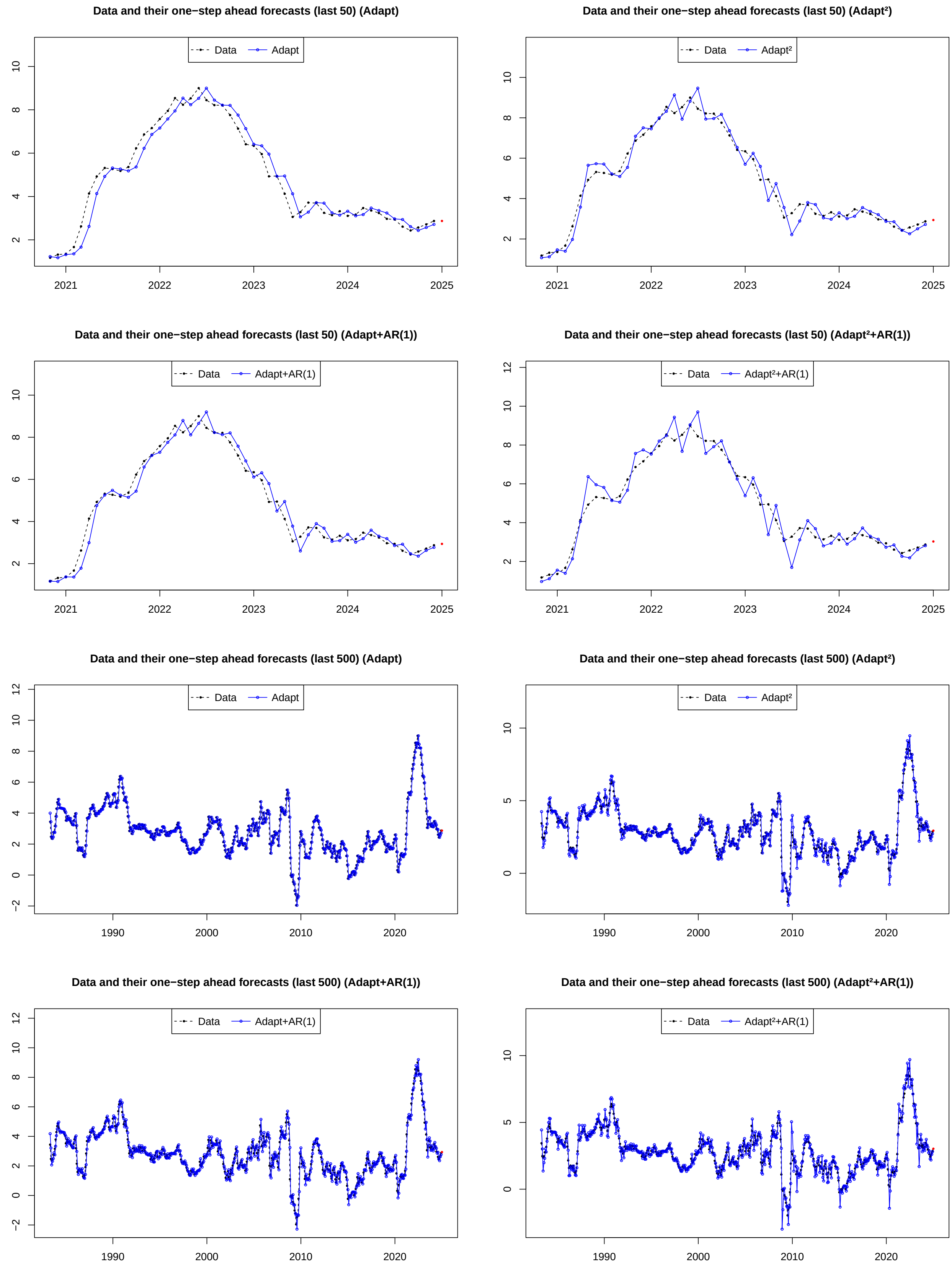

Figure 17: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

Selected tuning parameter

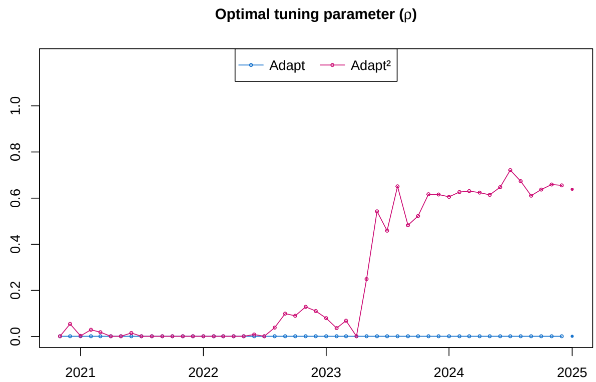

Figure 18: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

Evaluation of forecasting method

Figure 19: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

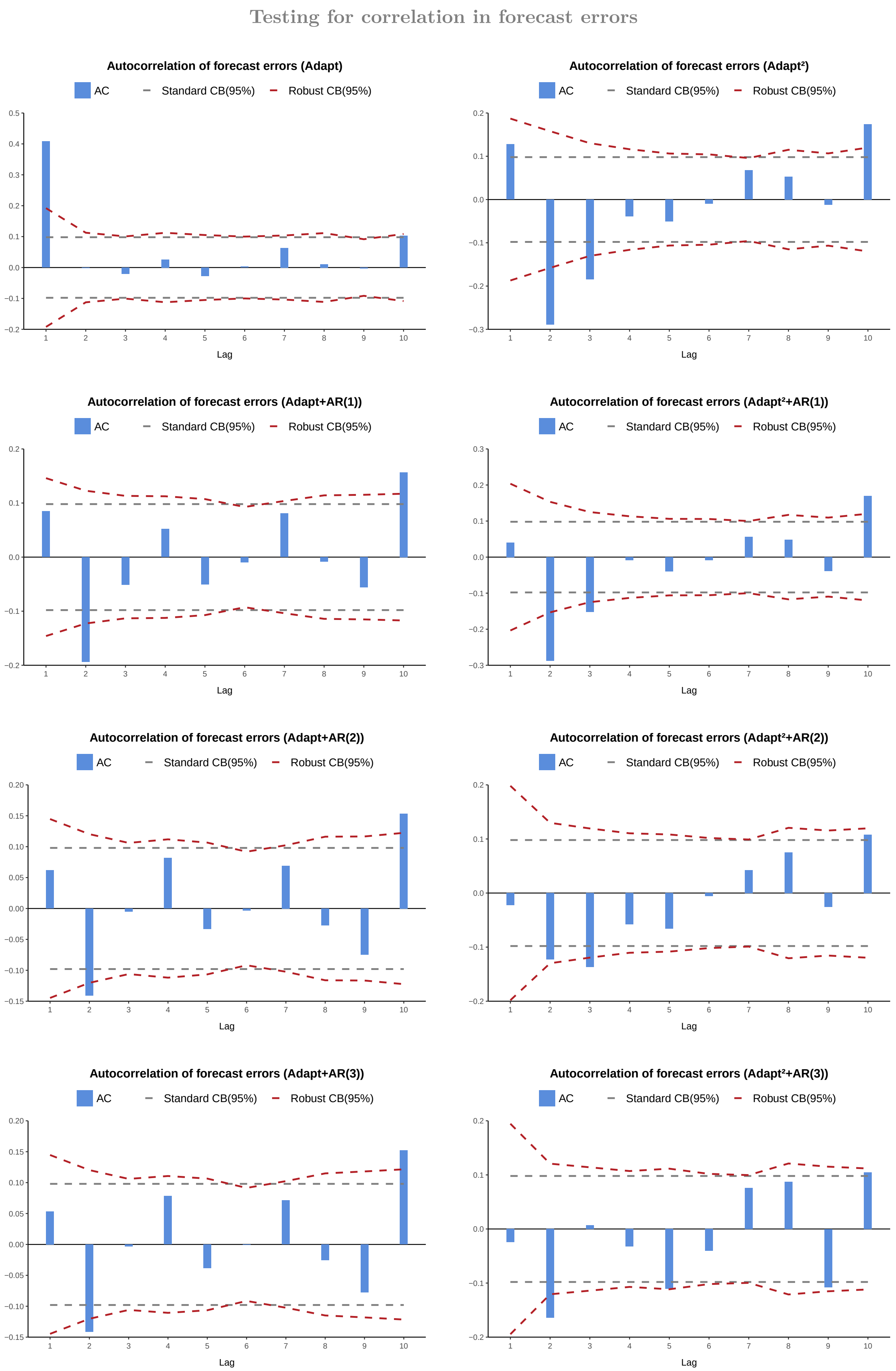

Figure 20: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Empirical example 2

We use the time series of the US monthly unemployment rate $\{UNRATE_t\}$ (in %), seasonally adjusted, over the period January 1950 – December 2024, with sample size $N = 900$ (retrieved on 01/09/2025). Note that, due to the COVID-19 pandemic, the time series exhibits a sharp spike in April 2020, which took more than a year to revert back to pre-pandemic levels. For exposition, we do not remove any observations. To see the effect of this episode on the adaptive forecasting, we present plots for the most recent 6 years, that is we set `P = 72`. The data are retrieved from FRED using the **quantmod** package (Ryan and Ulrich 2025), which depends on the **xts** package (Ryan and Ulrich 2026) for handling time series data.

```
R> library(quantmod)
R> getSymbols("UNRATE", src = "FRED", return.class = "xts")
R> UNRATE <- window(UNRATE, start = "1950-01-01", end = "2024-12-01")
```

We use the function `forAD` to evaluate adaptive forecasting for $\{UNRATE_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(UNRATE, P = 72, date_1 = "2025-01-01"))
```

From the tables, we observe that the relative MSFE of the adapt+AR, adapt^2, adapt^2+AR forecasts are well above 1. From the MSFEs plots (Figure 21), it is also clear that the adapt method yields the lowest forecast errors.

This result is driven by the extreme values of the unemployment rate during 2020. The resulting forecast errors for that period are large (Figure 24), which distorts the AR estimation and the second-stage adapt^2 forecasting, as well as the correlograms of the forecast errors (Figure 25).

The adaptive method down-weights these large forecast errors and, in many cases, assigns substantial weight to the most recent observation. This is seen from the tuning parameter values for the adapt method (Figure 23), which frequently attain the lower bound of the optimization range. Consequently, the adapt method behaves similarly to a last-observation-based forecast.

Based on the adapt method, the forecast of the unemployment rate for January 2025 is 4.1%, whereas the observed value was 4%.

The user could set `p_max = 0`, i.e. run `print(forAD(UNRATE, p_max = 0, P = 72, date_1 = "2025-01-01"))`, to focus on the outputs of the adapt and adapt^2 methods only.

Overall, the adaptive forecasting performs rather well, keeping in mind that no pre-treatment of the data was applied around the COVID-19 period.

We have the following outputs:

Adaptive+AR and adaptive^2+AR one-step ahead forecast output

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

	Forecast	MSFE	Relative MSFE
Adapt	4.10	1.30	1
Adapt+AR(1)	4.09	3.05	2.35
Adapt+AR(2)	4.07	3.13	2.40
Adapt+AR(3)	4.07	3.34	2.56
------------	--------	----	-------------
Adapt^2	4.11	2.78	2.14
Adapt^2+AR(1)	4.11	8.65	6.65
Adapt^2+AR(2)	4.08	10.50	8.07
Adapt^2+AR(3)	4.08	14.14	10.87

AR coefficients and standard errors (adapt+AR)

		const	ar1	ar2	ar3
AR(1)		-0.006	0.025	NA	NA
	s.e.	(0.029)	(0.050)	(NA)	(NA)
AR(2)		-0.007	0.028	-0.116**	NA
	s.e.	(0.029)	(0.049)	(0.049)	(NA)
AR(3)		-0.007	0.025	-0.115**	-0.020
	s.e.	(0.029)	(0.050)	(0.050)	(0.050)

AR coefficients and standard errors (adapt^2+AR)

		const	ar1	ar2	ar3
AR(1)		-0.02	-0.31***	NA	NA
	s.e.	(0.04)	(0.04)	(NA)	(NA)
AR(2)		-0.03	-0.36***	-0.17***	NA
	s.e.	(0.03)	(0.04)	(0.04)	(NA)
AR(3)		-0.03	-0.36***	-0.18***	-0.02
	s.e.	(0.04)	(0.05)	(0.05)	(0.05)

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	2.35	2.35	2.35	2.35	2.35	2.35	2.35	2.34	2.34	2.35	2.35
Adapt+AR(2)	2.40	2.40	2.41	2.40	2.40	2.40	2.41	2.40	2.40	2.40	2.40
Adapt+AR(3)	2.56	2.56	2.57	2.56	2.56	2.56	2.56	2.56	2.56	2.56	2.56
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	2.13	2.14	2.14	2.13	2.13	2.13	2.13	2.14	2.13	2.13	2.14
Adapt^2+AR(1)	6.64	6.64	6.65	6.64	6.65	6.64	6.64	6.65	6.64	6.65	6.65
Adapt^2+AR(2)	8.06	8.07	8.06	8.07	8.07	8.06	8.07	8.06	8.06	8.07	8.07
Adapt^2+AR(3)	10.86	10.87	10.86	10.86	10.87	10.86	10.87	10.86	10.86	10.87	10.87

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	3.9	3.9	4	4.1	4.2	4.2	4.1	4.1	4.2	4.1	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	3.89	3.89	3.89	3.99	4.09	4.19	4.19	4.10	4.10	4.19	4.10
Adapt+AR(1)	3.89	3.89	3.89	3.99	4.08	4.19	4.19	4.09	4.09	4.19	4.09
Adapt+AR(2)	3.91	3.87	3.89	3.99	4.07	4.18	4.18	4.09	4.10	4.19	4.07
Adapt+AR(3)	3.90	3.87	3.88	3.99	4.07	4.18	4.17	4.08	4.10	4.19	4.07
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	3.90	3.90	3.90	4.00	4.09	4.30	4.23	4.04	4.10	4.23	4.11
Adapt^2+AR(1)	3.80	3.86	3.86	3.93	4.02	4.24	4.23	4.04	4.05	4.17	4.11
Adapt^2+AR(2)	3.81	3.79	3.85	3.92	3.98	4.20	4.20	4.06	4.07	4.15	4.08
Adapt^2+AR(3)	3.81	3.80	3.84	3.92	3.98	4.19	4.19	4.06	4.08	4.15	4.08

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.001	0.001	0.001	0.001	0.086	0.001	0.001	0.001	0.001	0.001	0.001
Adapt^2	0.999	0.999	0.999	0.999	0.999	0.001	0.349	0.342	0.699	0.755	0.840

Selection of the best forecasting method

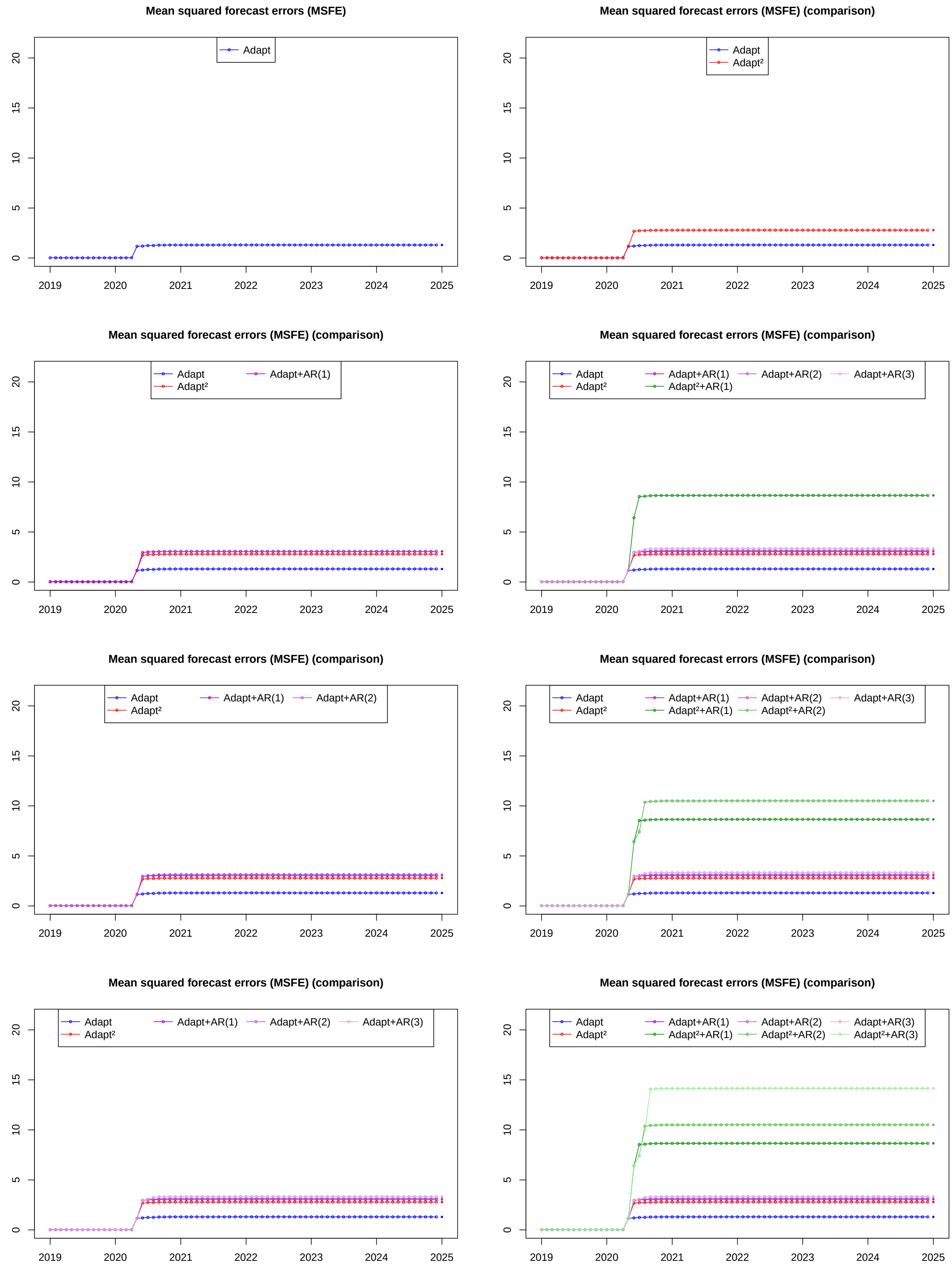

Figure 21: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

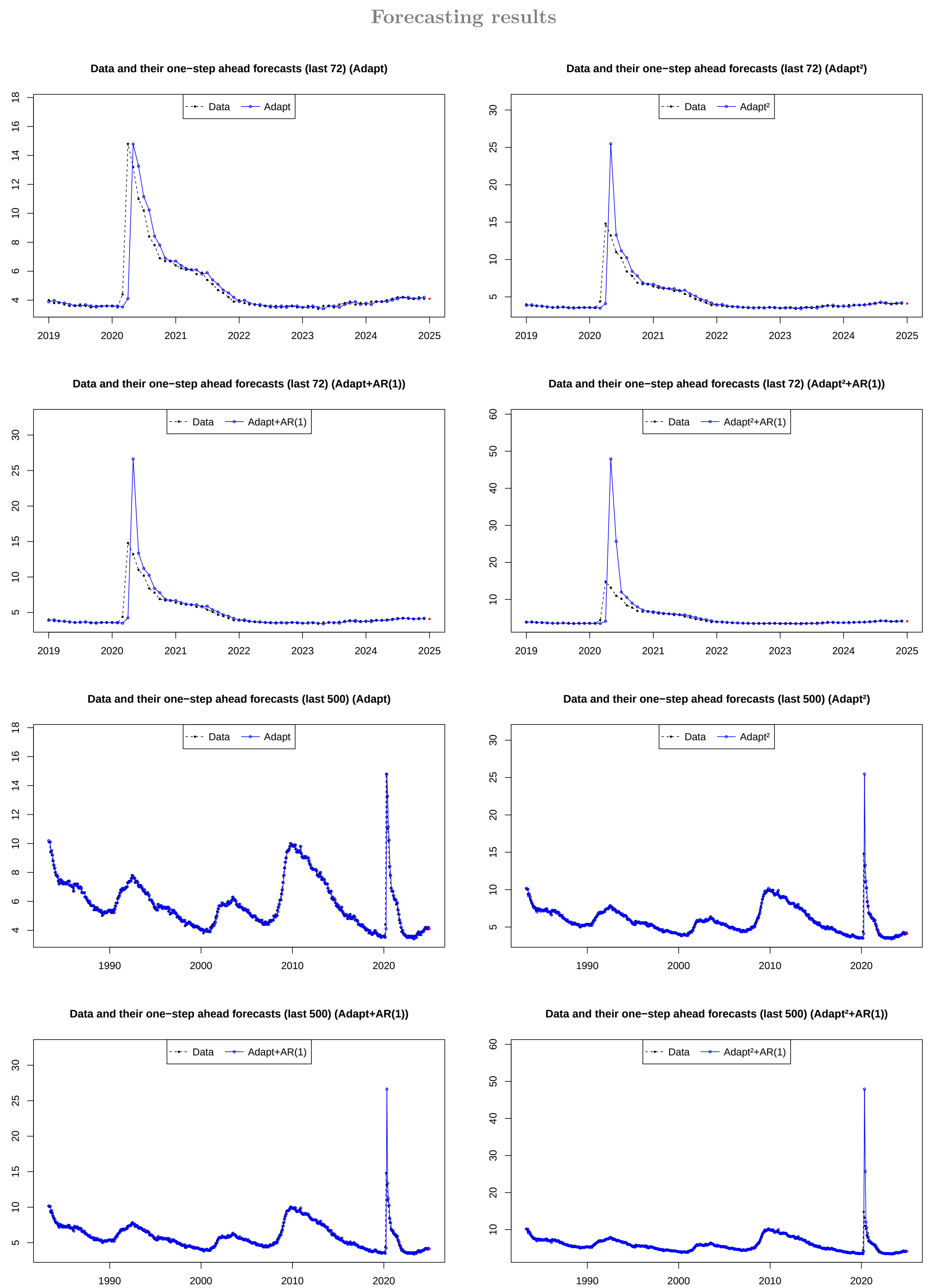

Figure 22: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

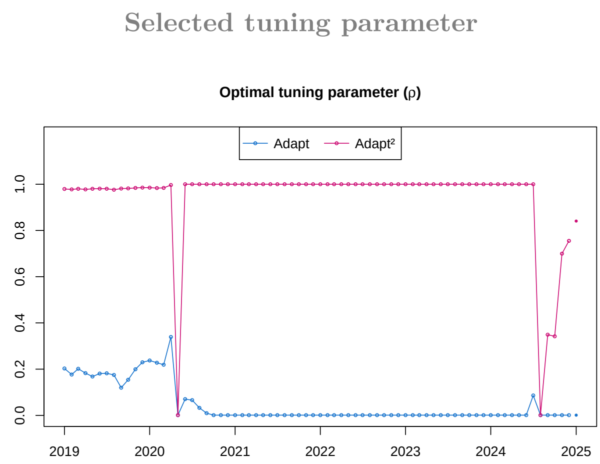

Figure 23: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

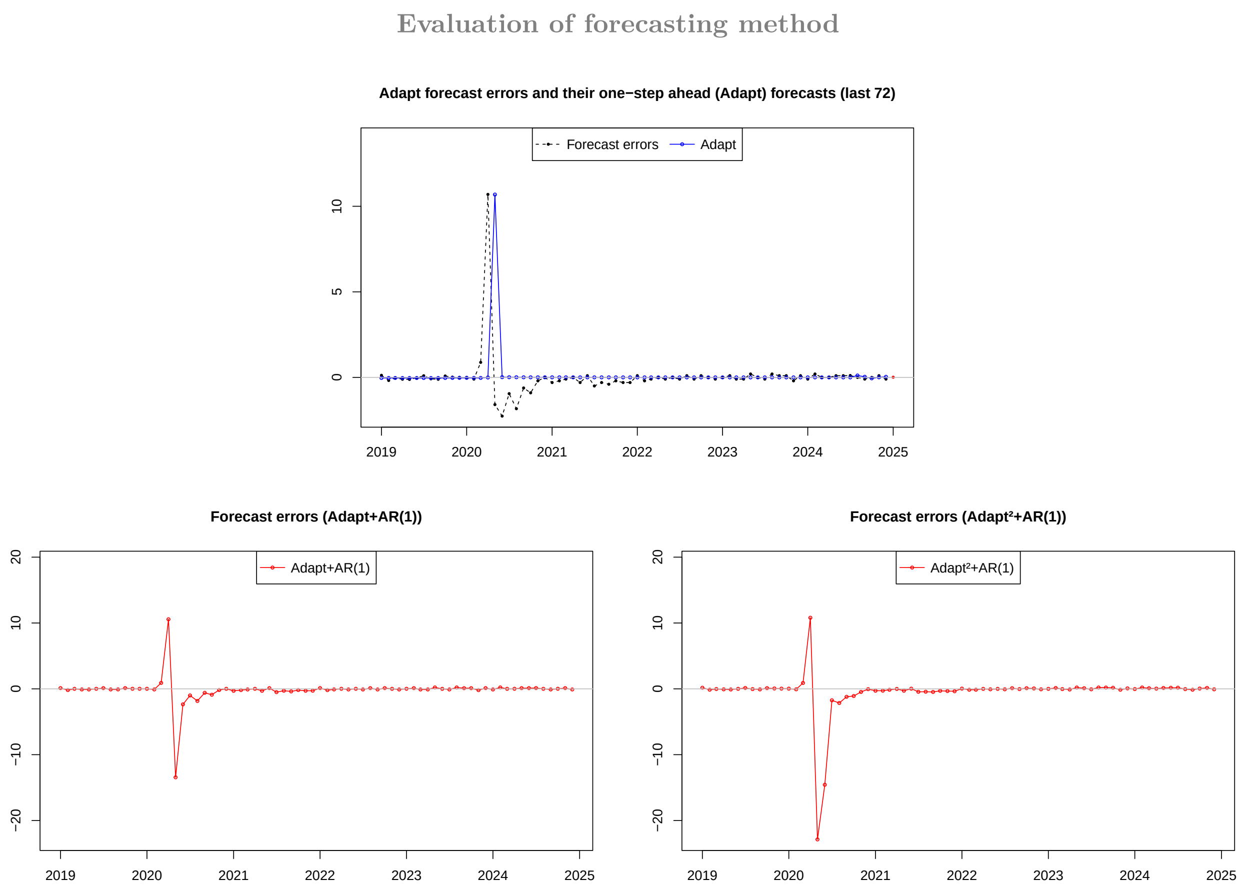

Figure 24: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

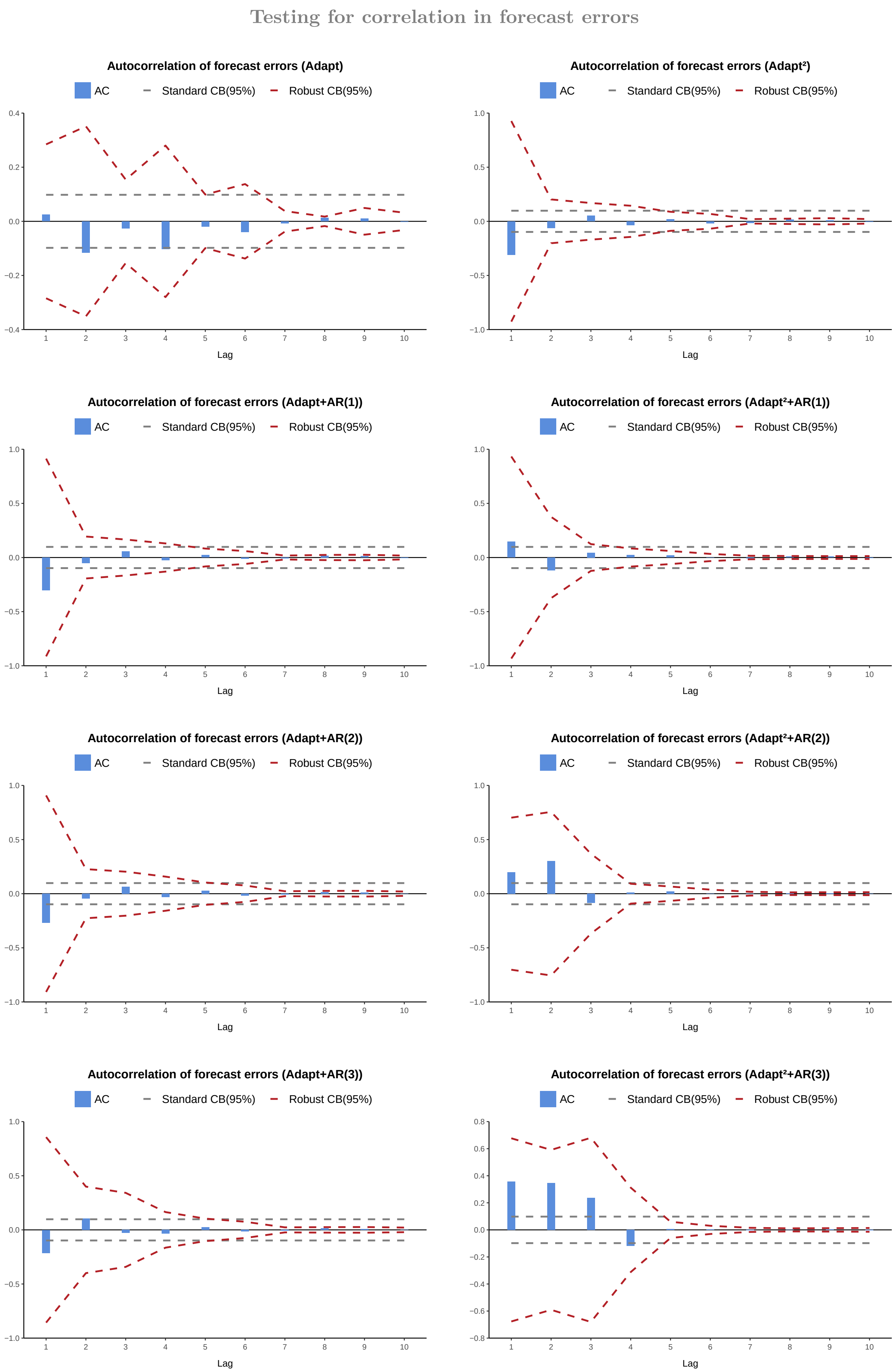

Figure 25: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Empirical example 3

We use the time series of the US quarterly growth rate $\{G_t\}$ (in %), seasonally adjusted, over the period 1950Q1 – 2024Q4 with sample size $N = 297$ (retrieved on 21/09/2025). We employ the package **quantmod** (Ryan and Ulrich 2025) to retrieve the data from FRED. This package depends on the **xts** package (Ryan and Ulrich 2026) for handling the time series data. We exclude the quarters 2020Q1-Q3, to mitigate the impact of the COVID-19 pandemic, reducing the sample size to $N = 294$.

```
R> library(quantmod)
R> getSymbols("GDP", src = "FRED", return.class = "xts")
R> G <- 100 * (GDP - lag(GDP, k = 1)) / lag(GDP, k = 1)
R> G <- window(G, start = "1950-01-01", end = "2024-10-01")
R> G <- G[!(index(G) %in% as.Date(c("2020-01-01", "2020-04-01", "2020-07-01")))]
```

We use the function `forAD` to evaluate adaptive forecasting for $\{G_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(G, date_1 = "2025-01-01"))
```

From the tables, we observe that the relative MSFE of the adapt^2, adapt^2+AR forecasts exceeds 1, while those of the adapt+AR are only slightly over 1. Together with the MSFE plots (Figure 26), this suggest that the simple adapt method is sufficient. Given the small differences in MSFE, the forecasts across the various methods are very similar, as seen in the tables and forecast plots (Figure 27).

The tables show that the estimated AR coefficients are not statistically significant (with one exception), indicating that AR adjustments are not required. Moreover, the second-stage adapt^2 procedure is not needed either: this is evident from the selected tuning parameters (Figure 28) which are equal to 1, implying that the second-stage adapt just reduces to the sample mean. This is further supported by the forecast error plot (see Figure 29), where applying adaptive forecasting to the adapt forecast errors yields values very close to zero.

Furthermore, the correlograms (Figure 30) show no substantial autocorrelation in forecast errors across all methods, further supporting the use of the simple adapt approach. Based on the adapt method, the forecast of growth rate of GDP for 2025Q1 is evaluated at 1.24%, compared to the observed value 0.8%, reflecting a slowdown of the economy.

The user could set `p_max = 0`, i.e. run `print(forAD(G, p_max = 0, date_1 = "2025-01-01"))`, to focus on the outputs of the adapt and adapt^2 methods only.

Overall, the adaptive forecasting performs rather well, particularly given that only minimal pretreatment of the data was applied around the COVID-19 period.

We have the following outputs:

```
------------------------------------------------------------
Adaptive+AR and adaptive²+AR one-step ahead forecast output
------------------------------------------------------------

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

|             | Forecast| MSFE| Relative MSFE|
|:------------|--------:|----:|-------------:|
|Adapt        |     1.24| 0.42|             1|
|Adapt+AR(1)  |     1.20| 0.42|          0.99|
|Adapt+AR(2)  |     1.19| 0.41|          0.99|
|Adapt+AR(3)  |     1.19| 0.42|          1.01|
|------------ | --------| ----| -------------|
|Adapt²       |     1.23| 0.44|          1.06|
|Adapt²+AR(1) |     1.18| 0.45|          1.07|
|Adapt²+AR(2) |     1.17| 0.45|          1.07|
|Adapt²+AR(3) |     1.17| 0.46|          1.09|


AR coefficients and standard errors (adapt+AR)

|      |     |  const|    ar1|    ar2|     ar3|
|:-----|:----|------:|------:|------:|-------:|
|AR(1) |     |  -0.03|   0.08|       |        |
|      |s.e. | (0.05)| (0.05)|   (NA)|    (NA)|
|AR(2) |     |  -0.03|   0.08|   0.05|        |
|      |s.e. | (0.05)| (0.05)| (0.05)|    (NA)|
|AR(3) |     |  -0.02|   0.09|   0.06| -0.12**|
|      |s.e. | (0.05)| (0.05)| (0.05)|  (0.05)|

AR coefficients and standard errors (adapt²+AR)

|      |     |   const|     ar1|     ar2|     ar3|
|:-----|:----|-------:|-------:|-------:|-------:|
|AR(1) |     |  -0.005|   0.013|        |        |
|      |s.e. | (0.054)| (0.060)|    (NA)|    (NA)|
|AR(2) |     |  -0.003|   0.011|   0.047|        |
|      |s.e. | (0.054)| (0.060)| (0.060)|    (NA)|
|AR(3) |     |  -0.001|   0.015|   0.047|  -0.096|
|      |s.e. | (0.054)| (0.060)| (0.060)| (0.060)|
```

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	1.00	1.00	0.98	0.99	1.00	0.99	1.00	1.01	1.00	0.99	0.99
Adapt+AR(2)	0.99	0.98	0.98	1.00	1.00	0.99	1.00	1.01	0.99	0.98	0.99
Adapt+AR(3)	1.00	1.00	1.00	1.01	1.01	1.00	1.03	1.03	1.01	1.01	1.01
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	1.07	1.05	1.06	1.06	1.06	1.08	1.07	1.07	1.08	1.06	1.06
Adapt^2+AR(1)	1.07	1.06	1.07	1.07	1.07	1.09	1.08	1.08	1.09	1.06	1.07
Adapt^2+AR(2)	1.07	1.06	1.09	1.06	1.07	1.08	1.07	1.10	1.07	1.07	1.07
Adapt^2+AR(3)	1.08	1.07	1.09	1.08	1.09	1.10	1.09	1.11	1.09	1.09	1.09

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	1.80	1.75	1.60	1.06	1.87	1.17	1.15	1.37	1.23	1.18	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	2.38	2.11	1.91	1.79	1.45	1.66	1.47	1.34	1.34	1.29	1.24
Adapt+AR(1)	2.35	2.04	1.85	1.74	1.35	1.67	1.40	1.28	1.31	1.25	1.20
Adapt+AR(2)	2.27	2.03	1.82	1.71	1.34	1.62	1.42	1.25	1.29	1.25	1.19
Adapt+AR(3)	2.13	2.21	1.82	1.78	1.37	1.67	1.51	1.20	1.36	1.29	1.19
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	2.37	2.11	1.90	1.78	1.44	1.65	1.46	1.33	1.33	1.28	1.23
Adapt^2+AR(1)	2.35	2.03	1.84	1.72	1.33	1.67	1.38	1.26	1.30	1.23	1.18
Adapt^2+AR(2)	2.21	2.02	1.79	1.69	1.30	1.59	1.43	1.21	1.27	1.24	1.17
Adapt^2+AR(3)	1.97	2.35	1.80	1.82	1.37	1.67	1.59	1.12	1.38	1.31	1.17

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.66	0.53	0.48	0.53	0.53	0.59	0.59	0.59	0.57	0.55	0.55
Adapt^2	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

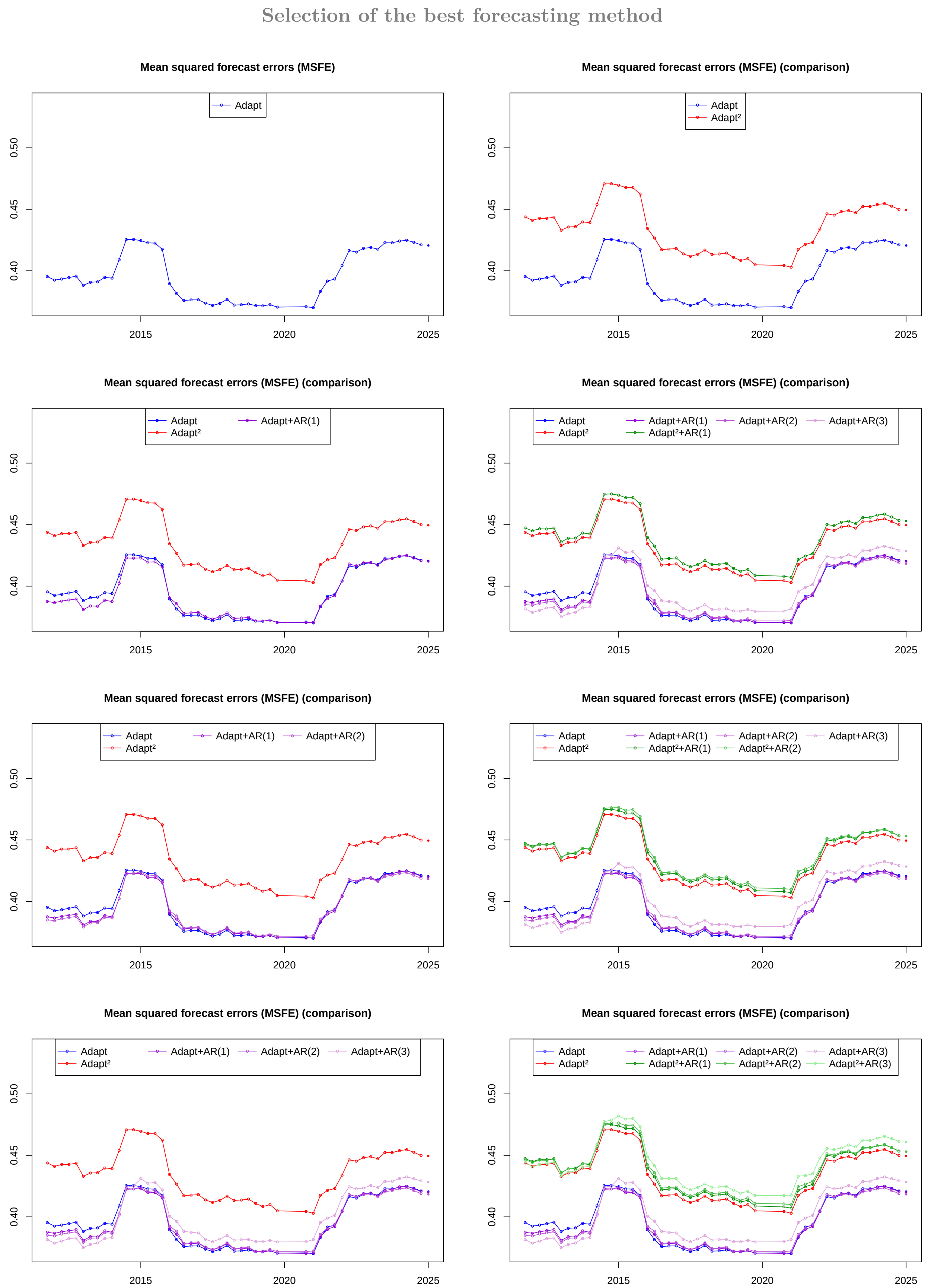

Figure 26: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Forecasting results

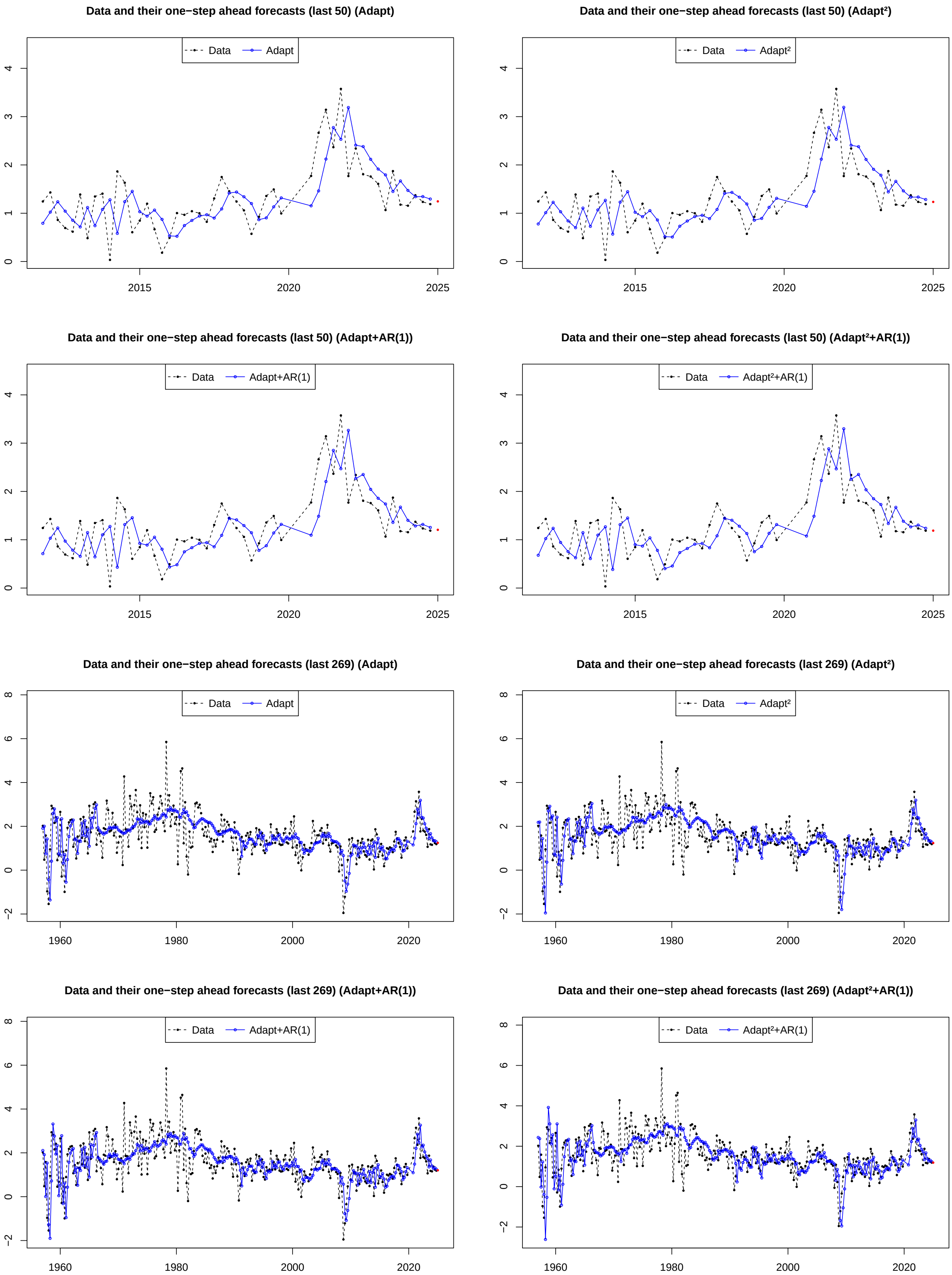

Figure 27: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

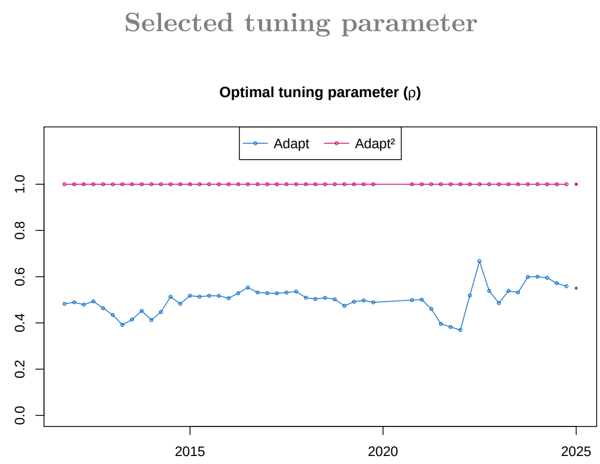

Figure 28: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

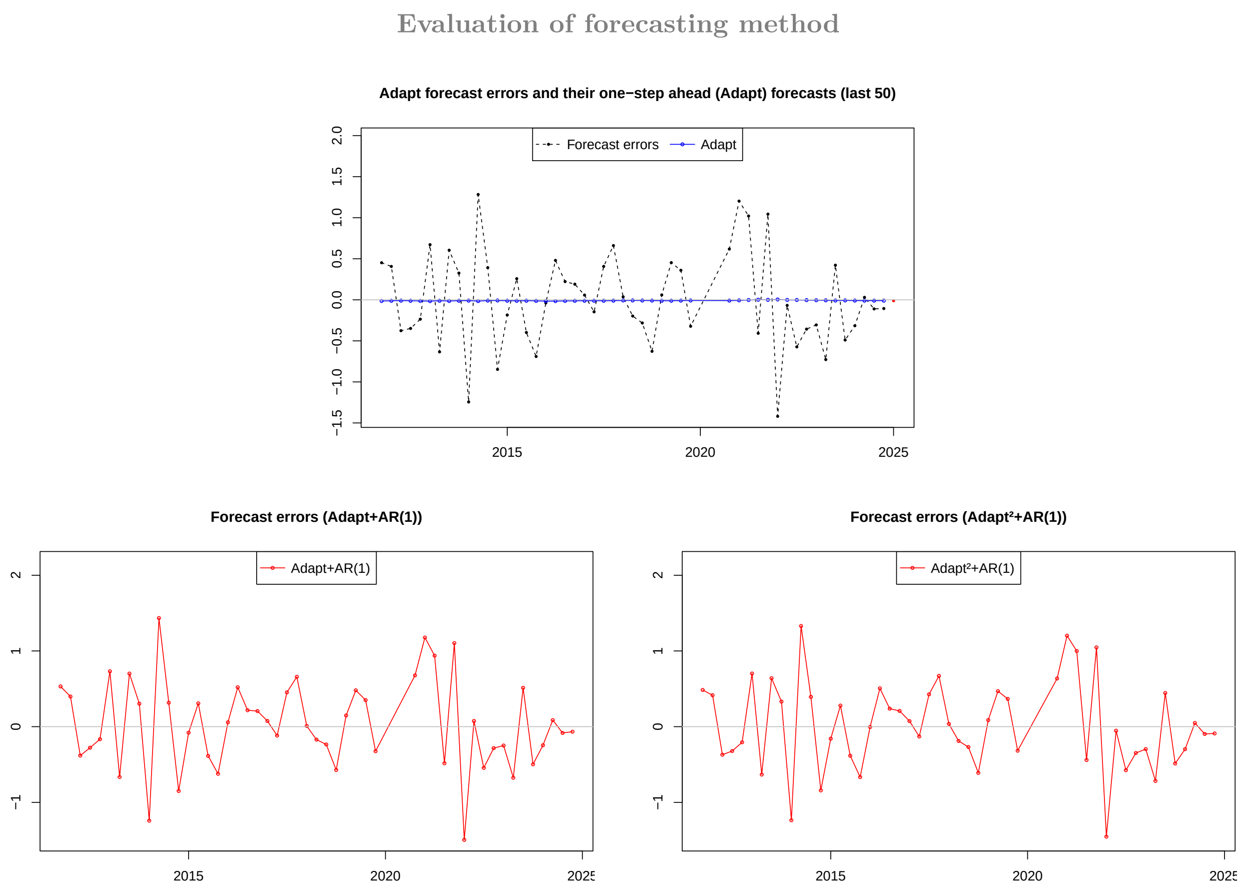

Figure 29: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

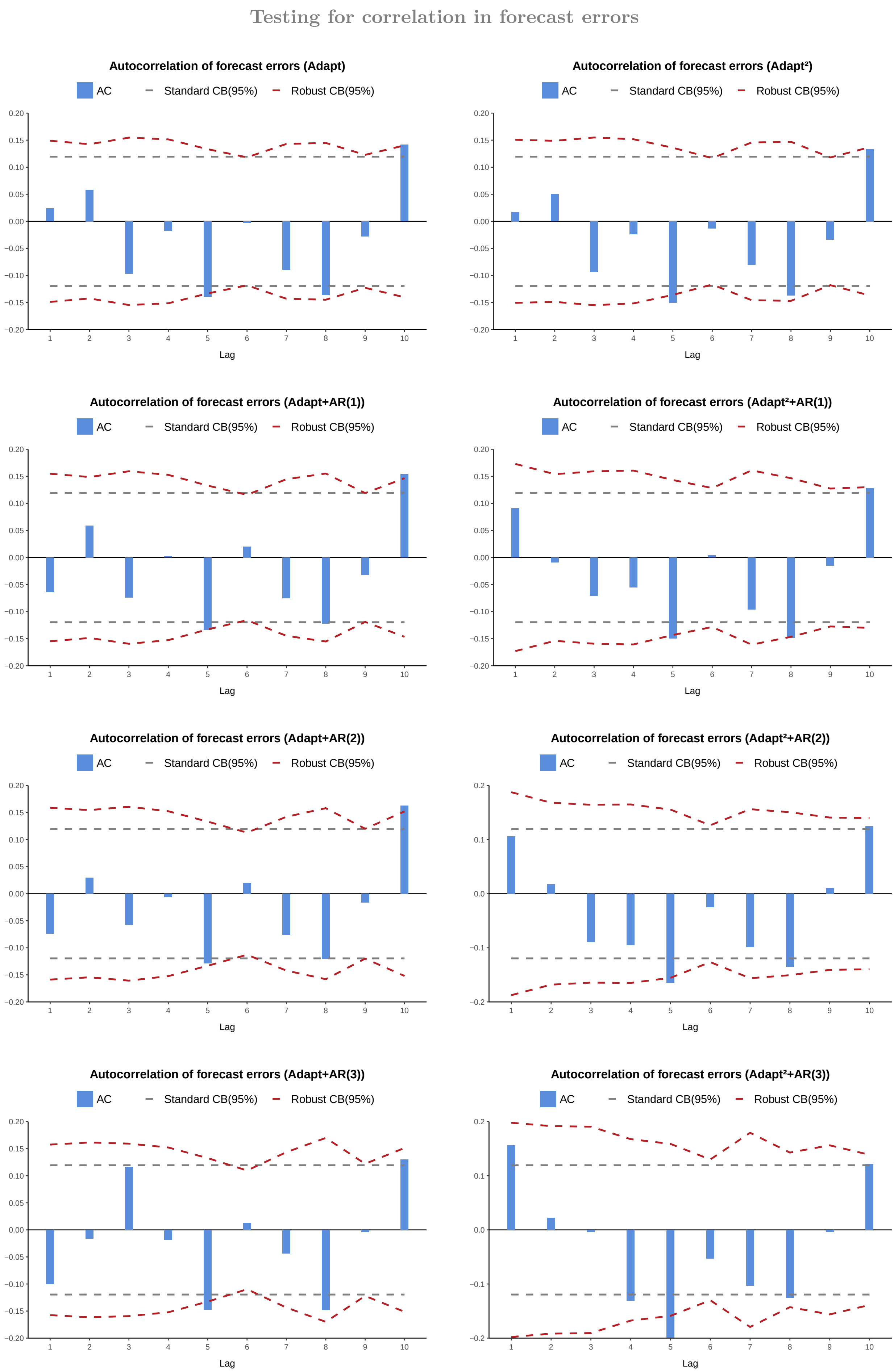

Figure 30: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

Empirical example 4

We consider the US GDP time series $\{GDP_t\}$ (in billions of USD), seasonally adjusted at an annual rate, over the period 1950Q1 – 2024Q4, with sample size $N = 297$ (retrieved on 21/09/2025). The data are retrieved from FRED using the **quantmod** package (Ryan and Ulrich 2025), which depends on the **xts** package (Ryan and Ulrich 2026) for handling time series data. To mitigate the effect of extreme values (outliers) associated with the COVID-19 period, we exclude the quarters 2020Q1–2020Q3. This reduces the sample size to $N = 294$.

```
R> getSymbols("GDP", src = "FRED", return.class = "xts")
R> GDP <- window(GDP, start = "1950-10-01", end = "2024-10-01")
R> GDP <- GDP[!(index(GDP) %in% as.Date(c("2020-01-01", "2020-04-01", "2020-07-01"
+          )))]
```

We use the function `forAD` to evaluate adaptive forecasting for $\{GDP_t\}$.

```
R> library(forecastADAPT)
R> print(forAD(GDP, date_1 = "2025-01-01"))
```

From the tables, we see that the relative MSFE of the adapt+AR, adapt^2, adapt^2+AR forecasts are well below 1. Together with the MSFE plots (Figure 31), this suggest that the simple adapt method is not sufficient. Among the alternative methods, the adapt^2 achieves a slightly lower MSFE. The rather weak performance of the adapt forecasts is also evident from the tables and forecast plots (Figure 32). Implementing AR(1) dynamics and the second-stage adapt^2 forecast procedure, visibly improves the forecasting performance of the adapt method.

The tables show that the estimated AR coefficients are not significant for the adapt^2+AR method, while they are significant for the adapt+AR method. The second level adapt^2 adjustment is required, as the selected tuning parameters (Figure 33) for the adapt^2 method lie away from the upper bound of 1. This is also seen from the forecast error plots (Figure 34), where applying adaptive forecasting to the adapt forecast errors, tracks a non-zero time-varying mean. In addition, the correlograms (Figure 35) show slow-decaying autocorrelation in the adapt forecast errors, reinforcing the need for a second round of adaptive adjustment. Using the adapt^2 method, the forecast of GDP for 2025Q1 is 30,083, compared to the observed value of 29,962.

The user could set `p_max = 0`, i.e. run `print(forAD(GDP, p_max = 0, date_1 = "2025-01-01"))`, to focus on the outputs of the adapt and adapt^2 methods only.

Overall, adaptive forecasting performs very well, particularly given that only minimal pre-treatment of the data was applied around the COVID-19 period.

We have the following outputs:

Adaptive+AR and adaptive^2+AR one-step ahead forecast output

Forecasts x[N+1|N], MSFE[N+1|N], RMSFE[N+1|N]

	Forecast	MSFE	Relative MSFE
Adapt	29723	67576	1
Adapt+AR(1)	30029	14430	0.21
Adapt+AR(2)	30049	13235	0.19
Adapt+AR(3)	30062	13320	0.19
------------	--------	----	-------------
Adapt^2	30083	12971	0.19
Adapt^2+AR(1)	30391	13543	0.20
Adapt^2+AR(2)	30411	13944	0.20
Adapt^2+AR(3)	30423	14803	0.21

AR coefficients and standard errors (adapt+AR)

		const	ar1	ar2	ar3
AR(1)		18.98***	0.82***		
	s.e.	(5.46)	(0.03)	(NA)	(NA)
AR(2)		12.90**	0.54***	0.34***	
	s.e.	(5.25)	(0.05)	(0.05)	(NA)
AR(3)		11.03**	0.48***	0.25***	0.16***
	s.e.	(5.26)	(0.05)	(0.06)	(0.05)

AR coefficients and standard errors (adapt^2+AR)

		const	ar1	ar2	ar3
AR(1)		2.724	0.013		
	s.e.	(4.394)	(0.060)	(NA)	(NA)
AR(2)		2.689	0.013	0.016	
	s.e.	(4.422)	(0.061)	(0.061)	(NA)
AR(3)		2.664	0.013	0.016	0.007
	s.e.	(4.450)	(0.061)	(0.061)	(0.061)

Adaptive+AR and adaptive^2+AR one-step ahead forecast output x[N-k+1|N-k]

Relative mean squared forecast errors RMSFE[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	1	1	1	1	1	1	1	1	1	1	1
Adapt+AR(1)	0.25	0.22	0.21	0.20	0.23	0.22	0.21	0.25	0.24	0.22	0.21
Adapt+AR(2)	0.22	0.21	0.19	0.23	0.21	0.20	0.19	0.23	0.21	0.20	0.19
Adapt+AR(3)	0.22	0.20	0.19	0.23	0.21	0.20	0.24	0.22	0.21	0.20	0.19
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	0.22	0.20	0.18	0.21	0.20	0.24	0.21	0.19	0.22	0.20	0.19
Adapt^2+AR(1)	0.22	0.20	0.19	0.21	0.20	0.24	0.21	0.20	0.22	0.21	0.20
Adapt^2+AR(2)	0.22	0.20	0.24	0.22	0.20	0.23	0.22	0.26	0.23	0.21	0.20
Adapt^2+AR(3)	0.23	0.21	0.25	0.22	0.21	0.24	0.23	0.26	0.23	0.22	0.21

Forecasts x[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Data	26272	26734	27164	27453	27967	28296	28624	29016	29374	29723	
-----------	----	----	----	----	----	----	----	----	----	----	----
Adapt	25805	26271	26733	27163	27453	27967	28296	28623	29016	29374	29723
Adapt+AR(1)	26303	26667	27128	27534	27707	28408	28583	28909	29357	29687	30029
Adapt+AR(2)	26293	26737	27157	27568	27765	28360	28665	28926	29354	29717	30049
Adapt+AR(3)	26386	26734	27187	27580	27783	28376	28649	28961	29356	29715	30062
------------	----	----	----	----	----	----	----	----	----	----	----
Adapt^2	26385	26798	27227	27629	27831	28412	28691	28988	29393	29742	30083
Adapt^2+AR(1)	26888	27195	27624	28000	28081	28860	28980	29275	29737	30058	30391
Adapt^2+AR(2)	26888	27266	27652	28034	28138	28809	29065	29290	29733	30088	30411
Adapt^2+AR(3)	26977	27276	27683	28049	28157	28827	29050	29325	29735	30086	30423

Tuning parameters rho[N-k+1|N-k]

	k=10	k=9	k=8	k=7	k=6	k=5	k=4	k=3	k=2	k=1	k=0
Adapt	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Adapt^2	0.55	0.52	0.50	0.52	0.50	0.56	0.56	0.55	0.55	0.54	0.54

Selection of the best forecasting method

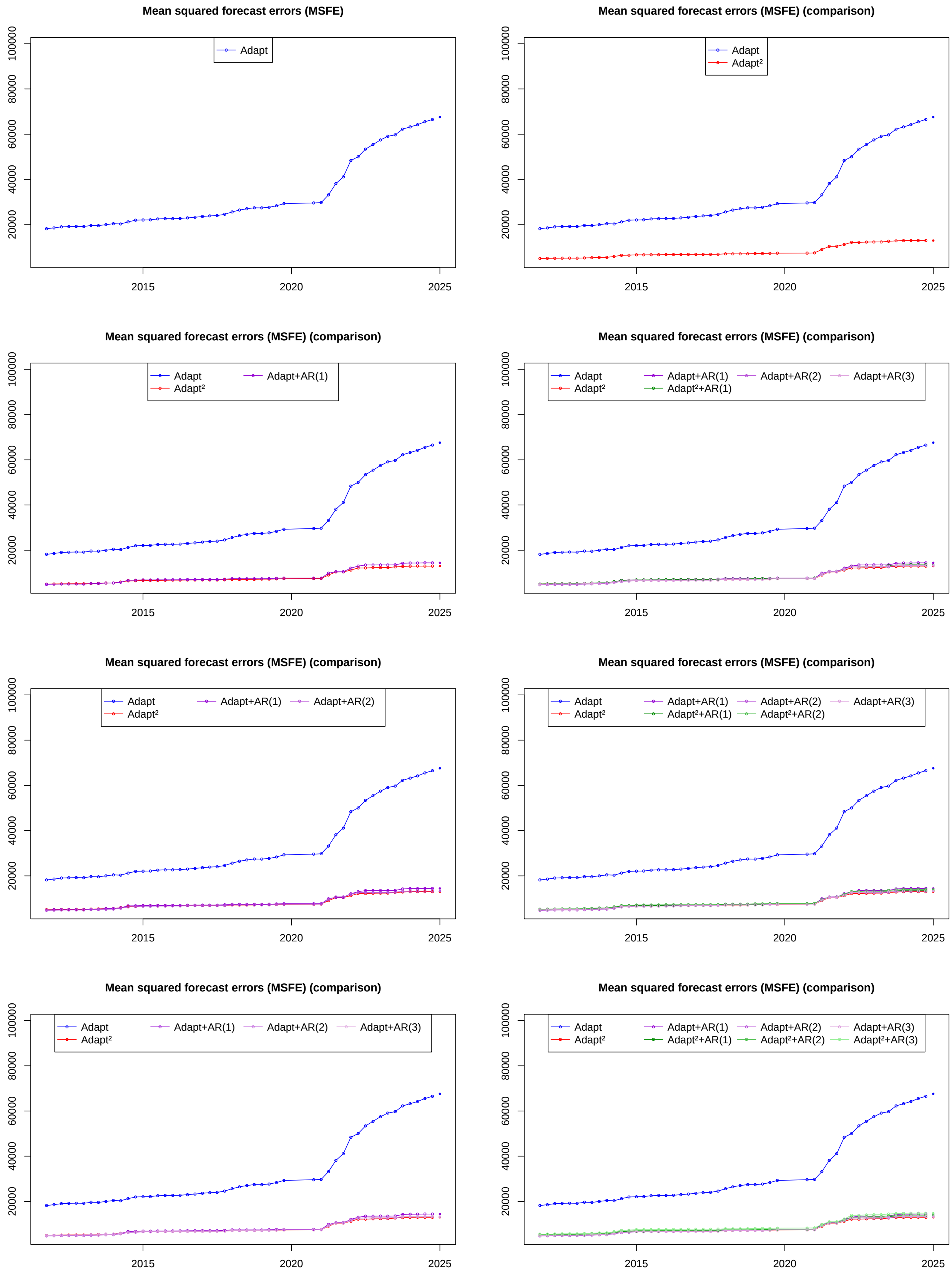

Figure 31: Comparison of the mean squared forecasts errors (MSFE) for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

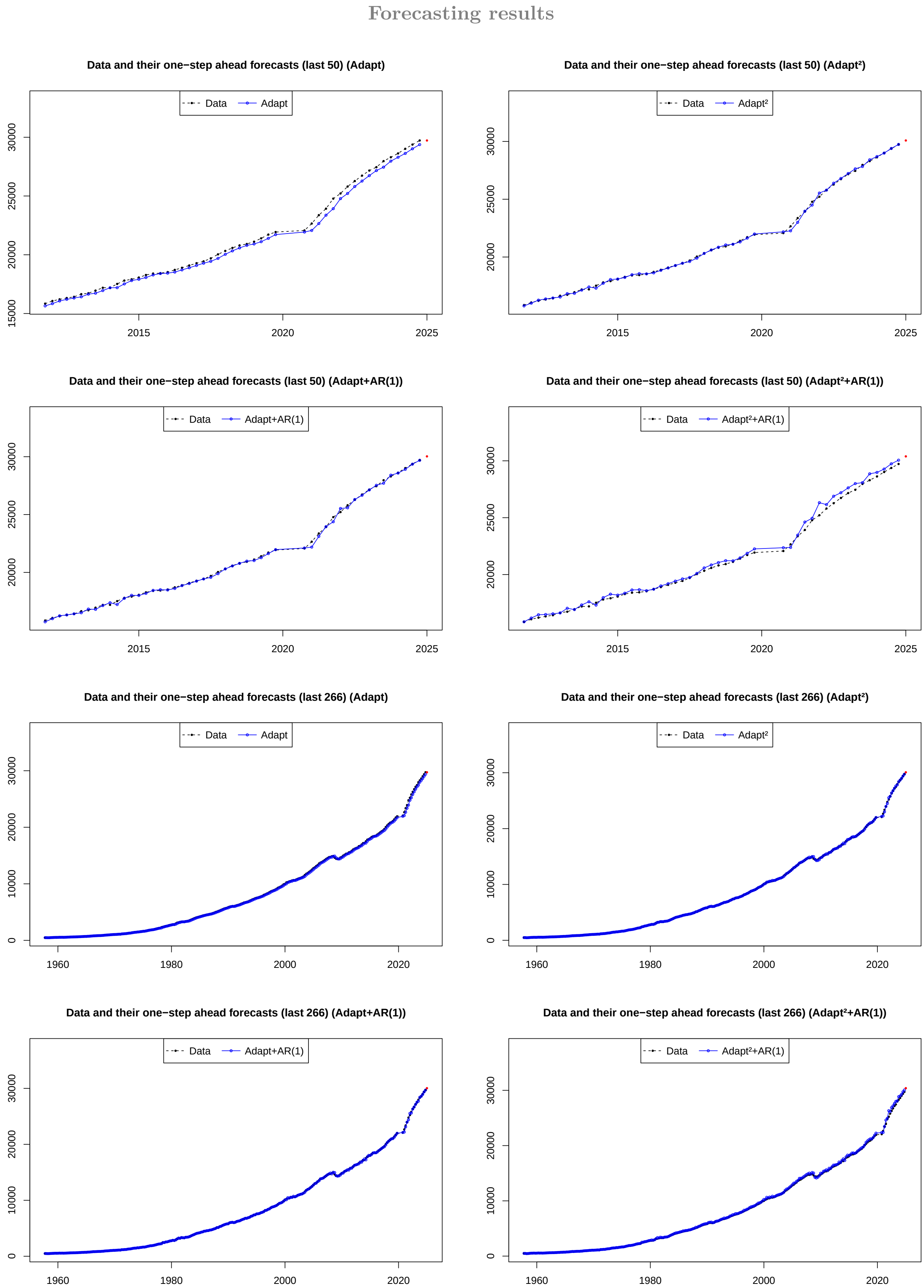

Figure 32: Data and their forecasts for the forecasting methods Adapt, Adapt+AR(1), Adapt^2, Adapt^2+AR(1).

Selected tuning parameter

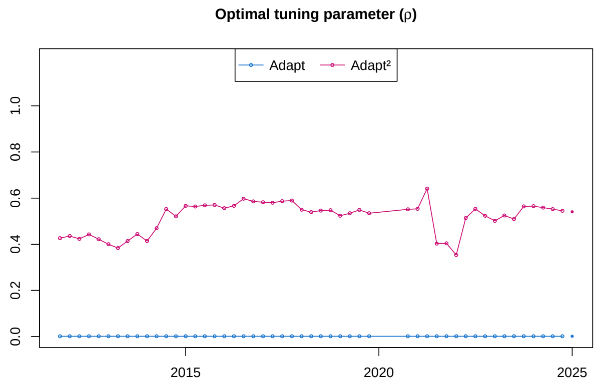

Figure 33: Selected tuning parameter for the forecasting methods Adapt, Adapt^2.

Evaluation of forecasting method

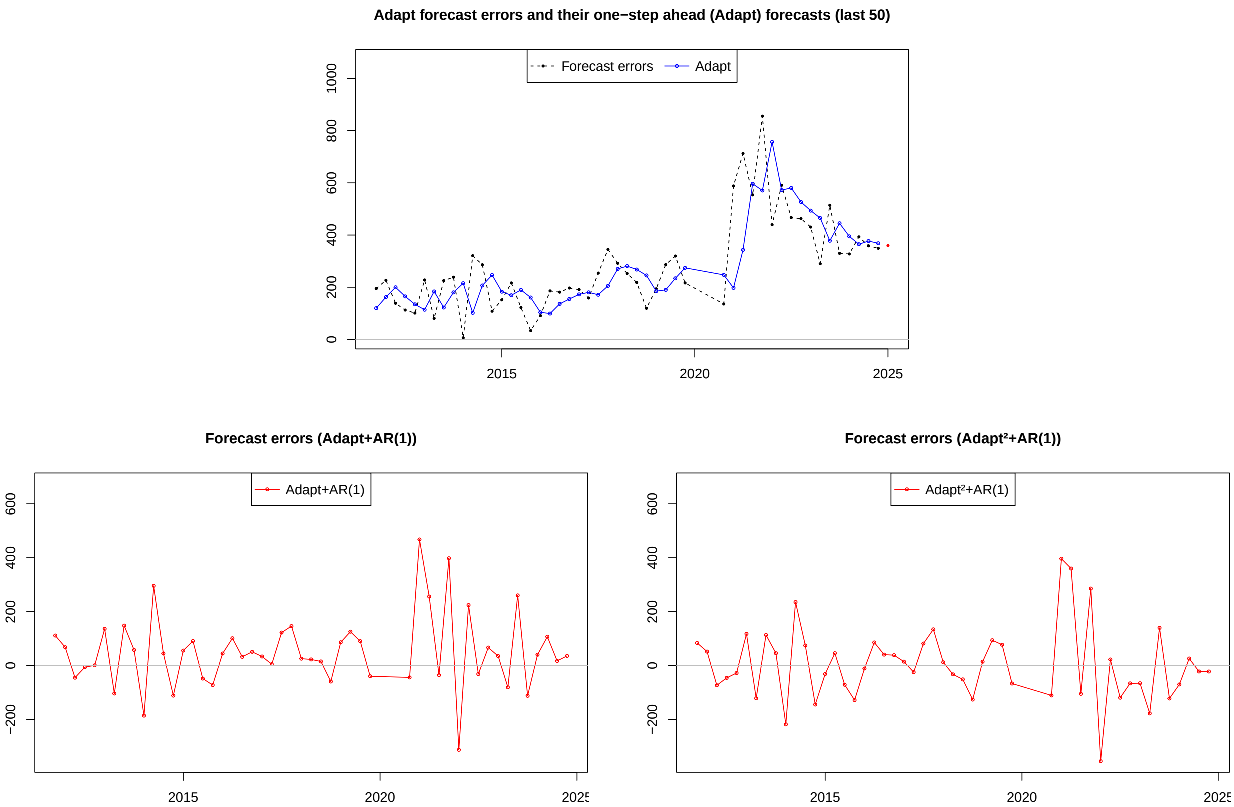

Figure 34: Forecasts errors for the forecasting methods Adapt+AR(1), Adapt^2+AR(1).

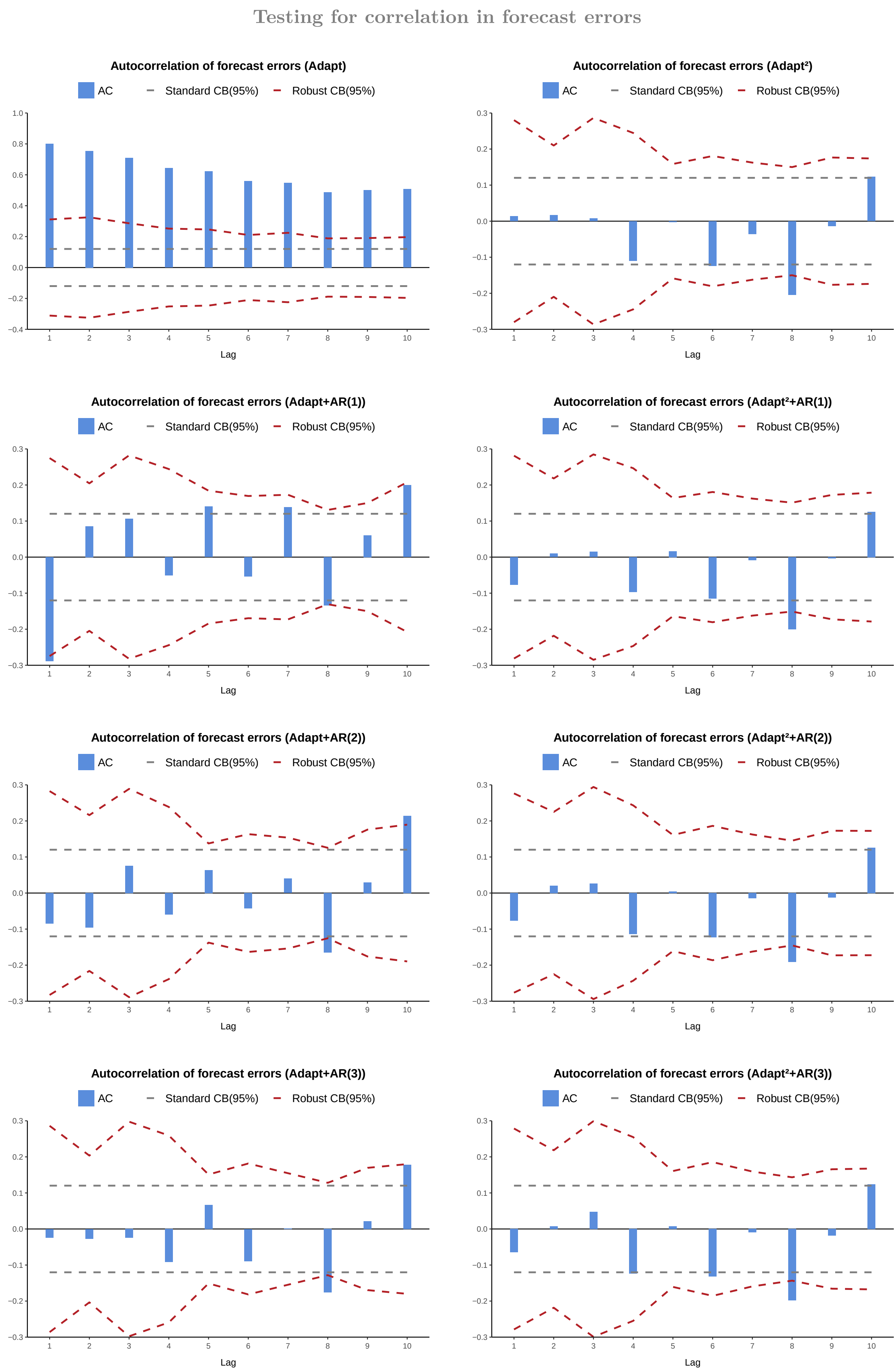

Figure 35: Correlogram of forecasts errors for the forecasting methods Adapt, Adapt+AR, Adapt^2, Adapt^2+AR.

References

Dalla V, Giraitis L, Phillips PCB (2025). *testcorr: Testing Zero Correlation.* R package version 0.3.0, URL https://CRAN.R-project.org/package=testcorr.

Giraitis L, Kapetanios G, Mansur M, Price S (2015). "Forecasting Under Structural Change." In J Beran, Y Feng, H Hebbel (eds.), *Empirical Economic and Financial Research: Theory, Methods and Practice*, pp. 401–419. Springer International Publishing, Cham. doi:10.1007/978-3-319-03122-4_25.

Giraitis L, Kapetanios G, Price S (2013). "Adaptive Forecasting in the Presence of Recent and Ongoing Structural Change." *Journal of Econometrics*, **177**(2), 153–170. doi:10.1016/j.jeconom.2013.04.003.

Hyndman RJ, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, Kuroptev K, Mücke M, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2026). *forecast: Forecasting functions for time series and linear models.* R package version 9.0.2, URL https://CRAN.R-project.org/package=forecast.

Hyndman RJ, Khandakar Y (2008). "Automatic Time Series Forecasting: The forecast Package for R." *Journal of Statistical Software*, **27**(3), 1–22. doi:10.18637/jss.v027.i03.

R Core Team (2026). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ryan JA, Ulrich JM (2025). *quantmod: Quantitative Financial Modelling Framework.* R package version 0.4.28, URL https://CRAN.R-project.org/package=quantmod.

Ryan JA, Ulrich JM (2026). *xts: eXtensible Time Series.* R package version 0.14.2, URL https://CRAN.R-project.org/package=xts.

Svetunkov I (2026). *smooth: Forecasting Using State Space Models.* R package version 4.4.0, URL https://CRAN.R-project.org/package=smooth.

Affiliation:

Violetta Dalla
Department of Economics
National and Kapodistrian University of Athens
Athens 10559, Greece
E-mail: vidalla@econ.uoa.gr
URL: https://en.econ.uoa.gr/index.php?id=31606

Liudas Giraitis
School of Economics and Finance
Queen Mary University of London

London E1 4NS, UK
E-mail: l.giraitis@qmul.ac.uk
URL: https://www.qmul.ac.uk/sef/staff/liudasgiraitis.html

George Kapetanios
Business School
King's College London
London WC2R 2LS, UK
E-mail: george.kapetanios@kcl.ac.uk
URL: https://www.kcl.ac.uk/people/george-kapetanios