UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Center For International Economics

Working Paper Series

Working Paper No. 2026-01

Diagnosing the trend and bootstrapping the forecasting intervals using a semiparametric ARMA

Dominik Schulz, Yuanhua Feng, Thomas Gries, Marlon Fritz and Sebastian Letmathe

März 2026

CIE Center For International Economics

Diagnosing the trend and bootstrapping the forecasting intervals using a semiparametric ARMA

Dominik Schulz, Yuanhua Feng, Thomas Gries,
Marlon Fritz and Sebastian Letmathe

Faculty of Business Administration and Economics, Paderborn University

March 25, 2026

Abstract

In this paper, an updated version of the R package `smoots` (smoothing time series) is introduced, which now includes functionalities for testing the linearity of trend functions and the stationarity of time series, which follow an additive component model, graphically and for forecasting as well as backtesting data-driven Semi-ARMA models. At first, the ideas for these new methods are developed. Furthermore, the Semi-Log-GARCH and the Semi-Log-ACD models are summarized as important model variants. Subsequently, the package is applied to test the trend functions in the monthly temperature changes of the Northern Hemisphere and the US Log-GDP for linearity. Moreover, the stationarity of the daily DAX returns and the VIX closing prices is assessed. Employing the `smoots` package, point forecasts and respective forecasting intervals are obtained for a variety of fitted Semi-ARMA models and backtests of these models are conducted. The forecasting quality of the Semi-ARMA is also compared to that of other methods.

Keywords: Asymptotics, linearity test, bootstrapping, interval forecasting, R

JEL Codes: C14, C51

1 Introduction

Within this paper, version 1.1.4 of the `smoots` package (smoothing time series, Feng et al., 2021) and its added functionalities are introduced. For a detailed discussion of package version 1.0.1 and its fundamental applications for data-driven local polynomial smoothing of trend stationary time series with short-memory errors, we refer the reader to Feng et al. (2020b) and the explanations therein. Further information on the IPI (iterative plug-in, Gasser et al., 1991) algorithm within `smoots` can be found in Feng et al. (2020a). In the updated package, three linearity tests for a time series' trend function are implemented. As will be shown, the first method can also be considered to conduct a stationarity test. Moreover, following the preliminary work of Fritz et al. (2018), who propose a method for calculating point predictions and respective prediction intervals with regard to semiparametric extensions of autoregressive moving-average (Semi-ARMA) models with Gaussian innovations, the updated `smoots` package has a built-in forecasting function for fitted Semi-ARMA models. Nonetheless, since the assumption of Gaussianity is violated in some cases (see e.g. Nelson and Granger, 1979, for a selection of non-normal economic data), the package also provides bootstrapped forecasting intervals. Related research on forecasting intervals for ARMA predictions can for example be found in Alonso et al. (2002), who consider a sieve bootstrap by Bühlmann (1997). In both the Gaussian and non-Gaussian case, we suggest to ignore the error in the predicted trend and to derive the prediction intervals exclusively from the detrended series. As a third contribution, the package includes functions, which allow for backtesting Semi-ARMA models using one-step-ahead rolling forecasts with forecasting intervals and scaled forecasting accuracy criteria (Hyndman and Koehler, 2006).

In Section 2, the general Semi-ARMA model with a nonparametric trend function will be presented alongside the data-driven trend estimation algorithm by Feng et al. (2020a) and some important model variants for financial econometrics. Furthermore, the linearity and stationarity tests will be discussed in Section 3, while the forecasting and backtesting techniques for Semi-ARMA models will be inspected in Section 4. A brief description of the new functions of the `smoots` package will be given in Section 5, whereas the package is applied to real data in Sections 6 and 7. Furthermore, also in Section 7, the forecasting performance of the Semi-ARMA model is compared to that of other established models.

2 The Semi-ARMA model and its estimation

In Section 2.1, the Semi-ARMA model and a general fitting approach will be described, while local polynomial regression and the general smoothing algorithm within the `smoots` package will be discussed briefly in Section 2.2. Moreover, notable variants of the Semi-ARMA model are the topic of Section 2.3.

2.1 The common Semi-ARMA model

Consider an observed time series y_t and define the time index as $t = 1, 2 \ldots, n$ with n being the number of observations. Assume an additive component model for y_t that consists of a smooth nonparametric trend function $m(x_t)$, where $x_t = t/n$ is the rescaled time on the interval $[0, 1]$, and a zero mean stationary rest ξ_t with short-memory. The model is formally given by

$$y_t = m(x_t) + \xi_t. \tag{1}$$

A Semi-ARMA model is defined by (1) and the additional assumption that the rest term ξ_t follows an autoregressive-moving-average (ARMA) model with autoregressive and moving-average orders p and q, respectively, i.e.,

$$\phi(B)\xi_t = \theta(B)u_t, \tag{2}$$

where the innovations u_t follow an independently and identically distributed (i.i.d.) white noise process with zero mean and constant variance σ_u^2, which will be denoted by $u_t \sim \text{IID}(0, \sigma_u^2)$. Moreover, $\phi(B) = 1 - \sum_{i=1}^{P} B^i \phi_i$ and $\theta(B) = \sum_{j=0}^{Q} B^j \theta_j$ are the characteristic polynomials with θ_j as well as ϕ_i being real valued coefficients and B being the backshift operator so that for $r = 0, 1, 2, \ldots,$ $B^r \xi_t = \xi_{t-r}$ holds.

A common procedure for model fitting consists of two separate estimation steps. At first, the values of the nonparametric trend function are estimated. For this purpose, we recommend the use of local polynomial regression (Ruppert and Wand, 1994; Cleveland and Loader, 1996; Fan and Gijbels, 1996) in this paper. This method includes automatic boundary correction (Fan and Gijbels, 1992; Hastie and Loader, 1993), which is an essential attribute of a nonparametric trend estimator in time series analysis, since it allows for more suitable predictions. Secondly, we propose to regard the detrended series as a realization of an ARMA process and thus, well-established methods, e.g. QMLE (quasi maximum likelihood estimation), can be applied to fit a parametric model to the residuals.

Different variants of the initial decomposition of the observed time series can be implemented as well. Another common approach is the multiplicative decomposition of time series. In this case, let $y_t^* > 0$ be the observed time series and $m^*(x_t)$ be the nonparametric trend function and ξ_t^* the multiplicative error with $E\left(\xi_t^*\right) = 1$ in y_t^*, respectively. Then

$$y_t^* = m^*(x_t) \times \xi_t^* \tag{3}$$

states the multiplicative component model. If we define $y_t = \ln(y_t^*)$, $m(x_t) = \ln[m^*(x_t)] + C_\xi$ and $\xi_t = \ln(\xi_t^*) - C_\xi$, where $C_\xi = E\left[\ln(\xi_t^*)\right]$, the logarithm of the observed series can be modeled according to a Semi-ARMA model (1) with (2).

2.2 Automatic trend estimation with local polynomials

The main smoothing method within `smoots` is local polynomial regression (Fan and Gijbels, 1996), because the boundary problem in nonparametric smoothing approaches is partially solved. The local polynomial estimator of the ν-th derivative of m, $m^{(\nu)}$, is regarded within this paper. For this purpose, define p as the order of the polynomial considered in the local regression. Let m be at least $(p+1)$-times differentiable at a point $x_0 \in X$ with $X = \{x_1, x_2, \ldots, x_n\}$. If $\nu \leq p$ and if $p - \nu$ is odd, solving the locally weighted least squares problem

$$Q(x_0) = \sum_{t=1}^{n} \left\{ y_t - \sum_{j=0}^{p} \beta_j (x_t - x_0)^j \right\}^2 W\left(\frac{x_t - x_0}{h}\right) \Rightarrow \min \tag{4}$$

yields the local polynomial estimator of $m^{(\nu)}$ at a point x_0 with h being the bandwidth and W being a weighting function on $[-1, 1]$, often a second order kernel function. Following this idea, $\hat{m}^{(\nu)}(x_0) = \nu!\hat{\beta}_\nu$ is an estimator of $m^{(\nu)}(x_0)$.

To quantify the quality of the local polynomial estimator, the mean integrated squared error (MISE)

$$\text{MISE}(h) = \int_{c_b}^{d_b} E\left\{\left[\hat{m}^{(\nu)}(x_t) - m^{(\nu)}(x_t)\right]^2\right\} \mathrm{d}x_t \tag{5}$$

is commonly evaluated, where $0 \leq c_b < d_b \leq 1$ are boundary cutoffs to reduce the effect of the boundary within the bandwidth selection.

Let $K_{(\nu,k)}(u)$ be the k-th order equivalent kernel that is realized for the estimation of $m^{(\nu)}$ at an interior point with $k = p + 1$. Following Feng et al. (2020a), the asymptotic MISE (AMISE) of the local polynomial estimator is then given by

$$\mathrm{AMISE}(h) = h^{2(k-\nu)} \frac{I[m^{(k)}]\beta^2_{(\nu,k)}}{(k!)^2} + \frac{2\pi c_f (d_b - c_b) R(K)}{nh^{2\nu+1}}, \tag{6}$$

where $c_f = (2\pi)^{-1} \sum_{l=-\infty}^{\infty} E\left(\xi_t \xi_{t+l}\right)$, $I[m^{(k)}] = \int_{c_b}^{d_b} [m^{(k)}(x)]^2 \mathrm{d}x$, $\beta_{(\nu,k)} = \int_{-1}^{1} u^k K(u) \mathrm{d}u$ and $R(K) = \int_{-1}^{1} K^2(u)\mathrm{d}u$. The asymptotically optimal bandwidth is then defined as

$$h_A = C_A n^{-1/(2k+1)} \tag{7}$$

with

$$C_A = \left[\frac{2\nu + 1}{2(k-\nu)} \frac{2\pi c_f (k!)^2 (d_b - c_b) R(K)}{I[m^{(k)}]\beta^2_{(\nu,k)}} \right]^{1/(2k+1)}. \tag{8}$$

Within the `smoots` package, a plug-in algorithm for automated bandwidth selection is implemented. Since only c_f and $I[m^{(k)}]$ are unknown in (8), a selected bandwidth can be acquired by inserting suitable estimates of these two quantities. As was stated by Feng et al. (2020a), the existence of error correlation does not have an impact on $\hat{I}[m^{(k)}]$. Therefore, approaches usually applied to independent error models will also yield a suitable estimate of $I[m^{(k)}]$ in the case of dependent errors. For the estimation of c_f, however, a nonparametric lag-window estimator is applied following the idea of Feng et al. (2020a) that corresponds to the algorithm by Bühlmann (1996) with minor adjustments. Given a starting bandwidth h_0 and a number of maximum iterations J, where $j = 1, 2, \ldots, J$ is the current iteration, $\hat{c}_f$ is obtained from the detrended values following a bandwidth $\mathrm{CF}h_{j-1}$ with CF being a suitable correction factor as displayed in Table 1 in Feng and Heiler (2009). Furthermore, an estimate of $I[m^{(k)}]$ is computed from estimates $\hat{m}^{(k)}$ of $m^{(k)}$ using a bandwidth $h_{d,j} = h^{\alpha}_{j-1}$, where α is a suitable inflation factor, and a local polynomial of order $k+1$. Ultimately, $\hat{c}_f$ and $\hat{I}[m^{(k)}]$ are inserted into (7) and the process is repeated multiple times for the respectively subsequent iteration until either convergence or the maximum number of iterations J is reached. A more detailed explanation of the underlying algorithms, especially the algorithm for the data-driven estimation of $m^{(1)}(x)$ and $m^{(2)}(x)$, which was omitted at this point, can be found in Feng et al. (2020a).

2.3 Some important variants

Both the Semi-ARMA model (1) with (2) described in Section 2.1 and the bandwidth selection algorithm for local polynomial regression presented in Section 2.2 are also relevant when modeling volatilites, durations or means on the financial market (see e.g. Feng et al., 2020b). As shown by Feng et al. (2020b), a variety of models, most notably the Semi-Log-GARCH (Feng et al., 2020b) and the Semi-Log-ACD (Forstinger, 2018; Feng et al., 2020b) model, admit a Semi-ARMA representation. Therefore, in the following, a brief summary of the Semi-Log-GARCH as well as the Semi-Log-ACD model will be provided.

To model conditional volatility, Engle (1982) and Bollerslev (1986) introduced the ARCH (autoregressive conditional heteroskedasticity) and GARCH (generalized ARCH) models, respectively, of which the latter has gained many extensions during the past decades. After the introduction of the Log-GARCH model (Pantula, 1986; Geweke, 1986; Milhøj, 1987), it has been established that the model has an ARMA representation that allows for consistent and asymptotically normal estimators of all parameters of the Log-GARCH representation except for the volatility intercept (see e.g. Francq and Zakoïan, 2006). It were then Sucarrat et al. (2016) who introduced a consistent and asymptotically normal estimator of the volatility intercept via established ARMA estimation approaches. In combination with the release of the `smoots` 1.0.1 package, Feng et al. (2020b) introduced a semiparametric extension of the Log-GARCH model (Semi-Log-GARCH), which will be described hereinafter. Let r_t be the (demeaned) log-returns. The Semi-Log-GARCH model is then defined by

$$\begin{aligned}
&r_t = \sqrt{\nu(x_t)}\epsilon_t, \quad \epsilon_t = \sqrt{h_t}\eta_t \;\text{ with }\; \eta_t \sim \mathrm{IID}(0,1) \;\text{ and} \\
&\ln(h_t) = \omega + \sum_{i=1}^{P} \alpha_i \ln(\epsilon_{t-i}^2) + \sum_{j=1}^{Q} \beta_j \ln(h_{t-j}),
\end{aligned} \tag{9}$$

where $t = 1, 2, \ldots, n$, $\mathrm{E}(\epsilon_t^2) = 1$, $\Pr(\eta_t = 0) = 0$, $\sqrt{\nu(x_t)} > 0$ is a smooth nonparametric (deterministic) scale function that represents the slowly changing unconditional standard deviation and $\sqrt{h_t} > 0$ are the conditional standard deviations. Feng et al. (2020b) show that the log-transformed series $\ln(r_t^2) = \ln[\nu(x_t)] + \mu_{l\epsilon} + \ln(\epsilon^2) - \mu_{l\epsilon}$ with $\mu_{l\epsilon} = \mathrm{E}[\ln(\epsilon_t^2)]$, corresponds to a Semi-ARMA model (1) and (2) with $y_t = \ln(r_t^2)$, $m(x_t) = \ln[\nu(x_t)] + \mu_{l\epsilon}$ and $\xi_t = \ln(\epsilon_t^2) - \mu_{l\epsilon}$, where the ξ_t follow an ARMA(P^*,Q) model with

$P^* = \max(P, Q)$. For $P \geq Q$ and using the data-driven local polynomial estimator described in Section 2.2 and well-established ARMA estimation approaches, $\sqrt{\nu(x_t)h_t}$ can be estimated consistently from the Semi-ARMA representation of a Semi-Log-GARCH model (Feng et al., 2020b).

Define $\phi(z) = 1 - \sum_{i=1}^{P^*}(\alpha_i + \beta_i)z^i$, where $\alpha_i = 0$ for $i > P$ and $\beta_i = 0$ for $i > Q$, and $A(z) = \sum_{i=1}^{P} \alpha_i z^i$. Following Francq and Zakoïan (2019, p. 87), $\ln(h_t) = c + \sum_{i=1}^{\infty} \delta_i \ln(\eta_{t-i}^2)$ with $c = \omega/[1 - \sum_{i=1}^{P^*}(\alpha_i + \beta_i)]$ and $\sum_{i=1}^{\infty} \delta_i z^i = \phi^{-1}(z)A(z)$ holds, if all roots of $\phi(z)$ lie outside of the unit circle. By defining $\mu_\eta(s) := \mathrm{E}[(\eta_t^2)^s]$, the moments of order $2m$, $m = 1, 2, ...,$ can be stated by

$$\mathrm{E}(\epsilon_t^{2m}) = \mu_\eta(m)\exp(mc)\prod_{i=1}^{\infty}\mu_\eta(m\delta_i) \tag{10}$$

as an extension of the general moments for Log-GARCH(1,1) processes provided by Sucarrat et al. (2016) and the first and second order moments of squared Log-GARCH(P,Q) processes mentioned by Francq and Zakoïan (2019, p. 88). Under a conditional normal distribution, $\mathrm{E}(\epsilon_t^{2m})$ exists, if and only if $\delta_i > -1/(2m)$ for all $i \in \mathbb{N}^+$. In contrast, under a conditional t-distribution with d_f degrees of freedom, $\mathrm{E}(\epsilon_t^{2m})$ exists, if and only if $-1/(2m) < \delta_i < d_f/(2m)$ for all $i \in \mathbb{N}^+$. These moment conditions can be derived, because the infinite sum in (10) converges under a conditional normal or a conditional t-distribution due to $\ln[\mu_\eta(s)] = O(s)$ as $s \to 0$ (Francq and Zakoïan, 2019, p. 88). Given (10) and since in a Semi-Log-GARCH, the parametric part is with $\mathrm{Var}(\epsilon_t) = 1$, we conclude the following.

Proposition 1. *Under consideration of the assumption $Var(\epsilon_t) = 1$ and following the definition of $E(\epsilon_t^2)$ in accordance with equation* (10),

$$\omega = -\ln\left(\prod_{i=1}^{\infty}\mu_\eta(\delta_i)\right)\left(1 - \sum_{j=1}^{P^*}(\alpha_j + \beta_j)\right) \tag{11}$$

has to be satisfied in a Semi-Log-GARCH model and thus, the log-volatility intercept ω is restrained by the coefficients α_i and β_j.

Proof. Proof see appendix A.

Another important model with respect to the analysis of non-negative financial time series is the ACD (autoregressive conditional duration, Engle and Russell, 1998) model, which admits a logarithmic extension (see the Type I model in Bauwens and Giot, 2000), called a Log-ACD, that corresponds to a squared Log-GARCH process. Moreover, Forstinger (2018) provided the definition of and examples with applications to suitable financial data for a semiparametric extension of the Log-ACD (Semi-Log-ACD) model. The Semi-Log-ACD model is analogous to the previously described Semi-Log-GARCH model. Define the observed values F_t by $F_t = g(x_t)\lambda_t e_t$, where $g(x_t) > 0$ is a nonparametric mean function, $\lambda_t > 0$ are the conditional means and $e_t \geq 0$ are i.i.d. innovations that satisfy $\Pr(e_t = 0) = 0$ and $\mathrm{E}(e_t) = 1$. Moreover, $\mathrm{E}(\lambda_t) = 1$ is assumed. Now F_t is equivalent to r_t^2, $g(x_t)$ to $\nu(x_t)$, λ_t to h_t and e_t to η_t^2 of the Semi-Log-GARCH model.

3 Graphical tools for testing stationarity and linearity

In this section, methods for testing a trend function for linearity or for testing the stationarity of a time series will be introduced. First, asymptotically unbiased local polynomial estimators and respective confidence bounds will be presented in Section 3.1. Subsequently, the exact methodology for conducting the linearity and stationarity tests will be discussed in Section 3.2.

3.1 Asymptotically unbiased estimators and confidence bounds

For computing confidence bounds, an asymptotically (relatively) unbiased local polynomial estimate $\hat{m}_{\mathrm{ub}}^{(\nu)}$ of $m^{(\nu)}$ can be calculated with a bandwidth $h_{\mathrm{ub}} = o[n^{-1/(2k+1)}]$, whose order of magnitude is smaller than that of the optimal bandwidth h_A. The variance of the local polynomial estimator is now the dominating part in the error in $\hat{m}_{\mathrm{ub}}^{(\nu)}$, while the bias is asymptotically insignificant. Thus,

$$\sqrt{nh^{2\nu+1}}[m^{(\nu)}(x) - \hat{m}_{\mathrm{ub}}^{(\nu)}(x)] \xrightarrow{\mathcal{D}} N[0, 2\pi c_f R^*(x)] \tag{12}$$

holds (Feng et al., 2019), where $R^*(x)$ is the constant $R(K)$ at interior points and it is adjusted correspondingly for x being a boundary point, which affects the equivalent kernel and the integration domain. The asymptotic variance follows from the second line of Equation (A.1) in the appendix of Beran and Feng (2002); the asymptotic normality is for example discussed in Theorem 6.1 in Fan and Gijbels (1996) and will still hold under general short-memory errors due to the asymptotic variance only being affected, so that the variance of the errors is being replaced by the sum of autocovariances of the error process, while convergence rates and other factors remain fully unaffected for the estimator. Since there is no rule for selecting h_{ub}, we propose $h = (h_{\mathrm{A}})^{(2k+1)/(2k)} = O[n^{-1/(2k)}]$ as a reasonable choice with the bias in $\hat{m}_{\mathrm{ub}}^{(\nu)}$ achieving the parametric convergence rate $O(n^{-1/2})$. Consequently, for the trend m, we derive $h = O(n^{-1/4})$ and $h = O(n^{-1/8})$ for a local linear and a local cubic estimator, respectively. The use of a bandwidth of an even smaller order is no longer necessary. Therefore, we suggest to calculate asymptotically unbiased estimates and respective confidence intervals using $\hat{h}_{\mathrm{ub}} = (\hat{h})^{(2k+1)/(2k)}$. Similarly, asymptotically unbiased estimates of m' and m'' and their confidence intervals can be obtained under consideration of analogously adjusted bandwidths. Thus, $\hat{h}_{1,\mathrm{ub}}^{d} = (\hat{h}_1^d)^{7/6}$ is a suitable adjustment for estimating m', whereas $\hat{h}_{2,\mathrm{ub}}^{d} = (\hat{h}_2^d)^{9/8}$ should be considered in the estimation of m''. If the weights for conducting a local polynomial regression are defined as $w_{i,t}$, where t describes the current time point, at which $\hat{m}_{\mathrm{ub}}^{(\nu)}$ is computed, and where i indicates that the respective weight is assigned to the i-th observation, the $100\alpha\%$-confidence interval for $m^{(\nu)}$ is denoted by

$$\left[\hat{m}_{\mathrm{ub}}^{(\nu)}(x_t) - \Phi\left(1-\frac{\alpha^*}{2}\right)\left(2\pi c_f \sum_{i=1}^{n} w_{i,t}^2\right)^{\frac{1}{2}}, \hat{m}_{\mathrm{ub}}^{(\nu)}(x_t) + \Phi\left(1-\frac{\alpha^*}{2}\right)\left(2\pi c_f \sum_{i=1}^{n} w_{i,t}^2\right)^{\frac{1}{2}}\right], \quad (13)$$

where $\Phi(\cdot)$ is the inverse standard normal cumulative distribution function and with $\alpha^* = 1-\alpha$. The standard errors implemented using the realized weights are asymptotically valid by plugging the relationship in Equation (3.6) in Beran and Feng (2002) into the first line of their Equation (A.1). This result also holds analogously at boundary points.

3.2 The methodology

In accordance with Jones (2002), a linear trend function is an adequate starting point in analyzing growth curves. In consequence, we propose the use of a linear trend as an indicator for the trend's suitability in different subperiods of a given time series. To test the trend for linearity graphically, we suggest to consider the confidence bounds of the asymptotically unbiased estimates of $m^{(\nu)}$ that were introduced in Section 3.1. We refer to Hjellvik et al. (1998) as an example for related research on nonparametric tests of linearity. As proposed by Fritz et al. (2019a),

$$H_0: y_t = \beta_0 + \beta_1 x_t + \xi_t, \text{ against } H_1: \text{“}\textit{A linear trend is not suitable.}\text{”}$$

are the null and alternative hypotheses H_0 and H_1 considered here, respectively. We propose three approaches to test H_0 with the confidence bounds obtained for $m^{(\nu)}$ for $\nu = 0, 1, 2$. Firstly, the confidence bounds in accordance with an unbiased local estimator $\hat{m}_{\mathrm{ub}}$ at given confidence level α can be computed under consideration of the bandwidth adjustment proposed in Section 3.1. Note that both bias-corrected local linear or local cubic estimates can be taken into account. Furthermore, a linear trend $\hat{m}_{\mathrm{L}}$, which is unbiased and $\sqrt{n}$-consistent under H_0, can be fitted to the data in contrast to the non-parametric estimates and the corresponding confidence bounds. Displaying all results together in a single figure allows for the graphical examination and the comparison of the estimates. If $\hat{m}_{\mathrm{L}}$ lies clearly outside the estimated confidence bounds at least at some observation points, H_0 should be rejected at the significance level $\alpha^* = 1 - \alpha$, because the probability for that those phenomena can happen is asymptotically less than α^*. As an important indicator, the fitted linear regression provides an ideal benchmark of the economic development, where usually log-linear growth theories are assumed; such a figure hence indicates where the economic development is beyond and where it is below an average level of the growth path of advanced economies.

Secondly, under H_0, β_1 is expected to be the constant first derivative of m, which is why we propose to illustrate the confidence bounds of m' together with the coefficient $\hat{\beta}_1$ of the estimated linear trend $\hat{m}_{\mathrm{L}}$. Thirdly, since m'' should be equal to zero, if H_0 holds, an asymptotically unbiased estimate of m'', $\hat{m}''_{\mathrm{ub}}$ and the corresponding confidence bounds can be depicted together with a horizontal line at level zero. For the second and third test, respectively, if clearly more than $100\alpha^*\%$ of the horizontal line is outside of the confidence intervals, H_0 should be rejected at the significance level α^*. In addition to the described tests, more information on the growth rate and the curvature of the considered time series can be drawn from the results of the second and third linearity test. Moreover, note that with the first linearity test the (trend-)stationarity of a time series can be assessed when $\hat{m}_{\mathrm{ub}}$ and the correspondingly computed confidence intervals of m are compared to the sample mean of the observations.

4 Semiparametric forecasting approaches

Since a parametric representation of the trend is not available when fitting Semi-ARMA models, point forecasts of the trend cannot be computed as intuitively. Therefore, a methodology for obtaining point predictions and corresponding forecasting intervals for Semi-ARMA models will be presented in Section 4.1 for cases, in which the ARMA innovations are (approximately) Gaussian. In non-Gaussian cases, a bootstrapping method as explained in Section 4.2 can be used to derive prediction intervals. A backtesting approach for Semi-ARMA models is discussed in Section 4.3.

4.1 Point and interval forecasting under normal errors

We once more assume a given process to follow a Semi-ARMA model defined by equations (1) and (2). The forecast of y_t for a time point $n+\tau$ with $\tau = 1, 2, \ldots$ is defined as $\hat{y}_{n+\tau}$. Consequently, suitable forecasting approaches based on both the estimated nonparametric trend function $\hat{m}(x_t)$ and the corresponding residuals $\hat{\xi}_t = y_t - \hat{m}(x_t)$ must be implemented.

Define D as a dummy variable that is either equal to zero or to one. Fritz et al. (2018) proposed

$$\hat{m}(x_{n+\tau}) = \hat{m}(x_n) + D \times \tau \times \Delta m \tag{14}$$

with $\Delta m = \hat{m}(x_n) - \hat{m}(x_{n-1})$ as a forcasting approach for the nonparametric trend function. If $D = 1$, equation (14) represents a linear extrapolation, whereas for $D = 0$ the forecasts are constant.

The best linear predictions in ARMA models are well-established (see e.g. Section 3.3 in Brockwell and Davis, 2016). Fritz et al. (2018) propose to obtain (approximately) best linear predictions from an ARMA model fitted to $\hat{\xi}_t$, because Feng and Zhou (2015) showed that additional errors within the best linear predictions due to errors in the estimated trend function are asymptotically negligible. Therefore, the (approximately) best linear predicitions $\hat{\xi}_{n+k}$ are given by

$$\hat{\xi}_{n+\tau} = \sum_{i=1}^{P} \hat{\phi}_i \hat{\xi}_{n+\tau-i} + \sum_{j=1}^{Q} \hat{\theta}_j \hat{u}_{n+\tau-j}, \tag{15}$$

where $\hat{\xi}_{n+\tau-i}$ are forecasted values, however for $k-i \leq 0$, they are replaced by the actual values of the detrended series. Similarly, $\hat{u}_{n+\tau-j}$ are the fitted innovations for the training set time points that are unkown and therefore replaced by $\mathrm{E}(u_t) = 0$ for $\tau - j \geq 1$. In accordance with Fritz et al. (2018), the total point forecasts of a Semi-ARMA model are then given by $\hat{y}_{n+\tau} = \hat{m}(x_{n+\tau}) + \hat{\xi}_{n+\tau}$.

The error in $\hat{m}(x_{n+\tau})$ is omitted for simplicity as proposed by Fritz et al. (2018). This is in analogy to Chapter 10.3 in Hyndman and Athanasopoulos (2021), where the authors produce forecasting intervals conditional on the forecasted future predictor values. If the process described in equation (2) is stationary, it admits an MA(∞) representation $\xi_t = \sum_{i=0}^{\infty} c_i u_{t-i}$ with $c_0 = 1$ and $\sum_{i=0}^{\infty} |c_i| < \infty$. An individual future observation derived from this representation is then approximately given by $\xi_{n+\tau} = \hat{\xi}_{n+\tau} + \sum_{i=0}^{\tau-1} c_i u_{n+j-i}$ and consequently, the approximate variance of the prediction error $\xi_{n+\tau} - \hat{\xi}_{n+\tau}$ is $\tilde{\sigma}_\tau^2 = \sigma_u^2 \sum_{i=0}^{\tau-1} c_i^2$ (see e.g. equation (3.3.19) in Brockwell and Davis, 2016).

As proposed by Fritz et al. (2018), under the normality assumption, i.e. $u_t \sim N(0, \sigma_u^2)$, suitable forecasting intervals for $\hat{y}_{n+\tau}$ at the confidence level α can therefore be obtained by considering only the error in $\hat{\xi}_t$, i.e.,

$$[\hat{y}_{n+\tau} - \Phi(1-\alpha^*/2)\tilde{\sigma}_\tau, \hat{y}_{n+\tau} + \Phi(1-\alpha^*/2)\tilde{\sigma}_\tau] \tag{16}$$

are the prediction intervals for $\hat{y}_{n+\tau}$, where $\alpha^* = 1-\alpha$ and $\Phi(i)$ is the $(i \times 100)$%-quantile of the standard normal distribution.

4.2 Bootstrapping the forecasting intervals

If the error term does not follow a Gaussian white noise or if the error distribution is unknown, forecasting intervals can also be computed with use of standard statistical software. An approach for the estimation of forecasting intervals via a forward bootstrap in ARIMA processes was first introduced by Pascual et al. (2004). Pan and Politis (2016) proposed a forward bootstrap for autoregressive time series models of order P, where the bootstrapped point forecasts are obtained based on the original series rather than the simulated data. Subsequently, Lu and Wang (2020) expanded the algorithm to ARMA(P, Q) processes to obtain forecasting intervals for the future time point $n+1$.

The *B-ARMARoots* algorithm (Lu and Wang, 2020) is conducted under the assumption that the orders P and Q of the underlying ARMA(P, Q) process (2) are known. We propose to expand the algorithm by Lu and Wang (2020) to multiple future time points $n+\tau$ with $\tau = 1, 2, \dots, H$. Moreover, for simplicity, we fit the ARMA models by MLE only. Let n be the number of observations, ξ_t the observed series and $s = 1, 2, \dots, M$ the number of iteration with M being the total number of iterations.

1. Estimate the coefficients θ_j and ϕ_i of the underlying ARMA process. Obtain the point forecasts $\hat{\xi}_{n+1}, \hat{\xi}_{n+2}, \dots, \hat{\xi}_{n+H}$. Furthermore, calculate the residuals $\hat{u}_t$ and define $\hat{F}_u$ as the empirical distribution of the demeaned residuals. Set $s = 1$.
2. Draw a sample $u^*_{t,s}$ from $\hat{F}_u$ and simulate a series $\xi^*_{1,s}, \xi^*_{2,s}, \dots, \xi^*_{n,s}$ based on the sampled residuals and the estimated values $\hat{\theta}_j$ and $\hat{\phi}_i$.
3. Obtain the estimates $\hat{\theta}^*_{j,s}$ and $\hat{\phi}^*_{i,s}$ for the simulated data and calculate the residual series $\hat{u}^{**}_{t,s}$ from the original process ξ_t based on these coefficients.
4. Obtain the point forecasts $\hat{\xi}^*_{n+1,s}, \hat{\xi}^*_{n+2,s}, \dots, \hat{\xi}^*_{n+H,s}$ according to (15) using ξ_t, $\hat{u}^{**}_{t,s}$, $\hat{\theta}^*_{j,s}$ and $\hat{\phi}^*_{i,s}$. Calculate the future bootstrap observations $\xi^*_{n+\tau,s}$ under consideration of ξ_t, $\hat{u}_t$, $\hat{\phi}_i$, $\hat{\theta}_j$ and H bootstrapped future innovations $u^*_{t,s}$ and then compute the forecasting error $\xi^*_{n+\tau,s} - \hat{\xi}^*_{n+\tau,s}$ for each τ. Increase s by one.
5. Execute steps 2 to 4 a total of M times and obtain the empirical distributions of $\xi^*_{n+\tau} - \hat{\xi}^*_{n+\tau}$ for each τ with their respective $(\alpha^*/2)100\%$-quantiles being denoted by $q_\tau(\alpha^*/2)$.

The forecasting intervals are now given by $[\hat{\xi}_{n+\tau} + q_\tau(\alpha^*/2), \hat{\xi}_{n+\tau} + q_\tau(1-\alpha^*/2)]$. Since the error in $\hat{m}(x_{n+\tau})$ is omitted for simplicity, the bootstrapped forecasting intervals under consideration of a Semi-ARMA model are given by

$$[\hat{y}_{n+\tau} + q_\tau(\alpha^*/2), \hat{y}_{n+\tau} + q_\tau(1-\alpha^*/2)]. \tag{17}$$

4.3 Backtesting the results

Given a time series y_t with $t = 1, 2, \dots, n$, a common backtesting approach is to treat its first n_{tr} observations as the actual training data $y_{t,\mathrm{tr}}$ and the final $n_{\mathrm{te}} = n - n_{\mathrm{tr}}$ realizations as the test set. Fitting models to the training data and obtaining one-step-ahead rolling point forecasts $\hat{y}_{n_{\mathrm{tr}}+1,\mathrm{tr}}, \hat{y}_{n_{\mathrm{tr}}+2,\mathrm{tr}}, \dots, \hat{y}_{n_{\mathrm{tr}}+n_{\mathrm{te}},\mathrm{tr}}$ with corresponding forecasting intervals that can be compared to the actual test data is a viable method to assess the quality of a model. For a given confidence level $0 < \alpha < 1$, we expect approximately $n_{\mathrm{te}}(1-\alpha)$ of the test observations to lie outside of the forecasting intervals. Commonly, $n_{\mathrm{te}} = 5$ and $\alpha = 0.95$ (or $\alpha = 0.99$) are used and consequently, 0.25 observations are expected not to be covered by the intervals. We therefore recommend that a model should be clearly rejected at confidence levels $\alpha = 0.95$ and $\alpha = 0.99$, if one or more of the five test observations are not between the forecasting bounds.

Hyndman and Koehler (2006) introduced the mean absolute scaled error (MASE) as a scale-independent measure to quantify the goodness of fit of a model, which is defined by

$$\mathrm{MASE} = \frac{1}{n_{\mathrm{te}}} \sum_{\tau=1}^{n_{\mathrm{te}}} \left| \frac{y_{n_{\mathrm{tr}}+\tau} - \hat{y}_{n_{\mathrm{tr}}+\tau,\mathrm{tr}}}{\frac{1}{n_{\mathrm{tr}}-1} \sum_{t=2}^{n_{\mathrm{tr}}} |y_t - y_{t-1}|} \right|, \tag{18}$$

and also proposed the root mean squared scaled error (RMSSE), which was formally denoted by Fritz et al. (2018) as

$$\mathrm{RMSSE} = \sqrt{\frac{1}{n_{\mathrm{te}}} \sum_{\tau=1}^{n_{\mathrm{te}}} \frac{(y_{n_{\mathrm{tr}}+\tau} - \hat{y}_{n_{\mathrm{tr}}+\tau,\mathrm{tr}})^2}{\frac{1}{n_{\mathrm{tr}}-1} \sum_{t=2}^{n_{\mathrm{tr}}} (y_t - y_{t-1})^2}}. \tag{19}$$

The MASE corresponds to the mean absolute error (MAE), however, each absolute error is scaled by the MAE of a naive forecasting approach applied to the training data. Thus, a MASE that is less than one indicates a better forecasting performance in comparison to the naive approach; more generally, a MASE as close as possible to zero is preferable. Since the criterion is scale-independent, the MASE between different forecasting techniques and also between different datasets is comparable. Further explanations with respect to the RMSSE are omitted, because it can be interpreted similarly to the MASE.

To obtain suitable rolling point forecasts for Semi-ARMA models, we propose to assume that the error in the trend forecasts is approximately negligible. Therefore, $\tilde{u}_t = y_t - \hat{y}_{t,\mathrm{tr}}$ are treated as the true innovations u_t and $\tilde{\xi}_t = y_t - \hat{m}_{\mathrm{tr}}(x_t)$ as the true detrended series ξ_t for $t = n_{\mathrm{tr}}+1, n_{\mathrm{tr}}+2, \ldots, n_{\mathrm{tr}}+n_{\mathrm{te}}$. Let $\hat{\xi}_{n_{\mathrm{tr}}+\tau,\mathrm{tr}}$ be the rolling point forecasts obtained stepwisely following (15) under consideration of the detrended training observations $\hat{\xi}_t$ for $\tau - i \leq 0$ and the fitted training innovations $\hat{u}_t$ for $\tau - j \leq 0$, while $\tilde{\xi}_t$ and $\tilde{u}_t$ are implemented for $\tau - i \geq 1$ and $\tau - j \geq 1$, respectively. Consequently, $\hat{y}_{n_{\mathrm{tr}}+\tau,\mathrm{tr}} = \hat{m}_{\mathrm{tr}}(x_{n_{\mathrm{tr}}+\tau}) + \hat{\xi}_{n_{\mathrm{tr}}+\tau,\mathrm{tr}}$ are the total rolling point forecasts. Furthermore, note that a forecasting interval for one-step ahead rolling forecasts needs to be computed only once for $\tau = 1$ in accordance with the approaches discussed in either Section 4.1 or Section 4.2; for $\tau = 2, 3, \ldots, n_{\text{te}}$, the forecasting interval obtained for $\hat{y}_{n_{\text{tr}}+1,\text{tr}}$ can be recentered around $\hat{y}_{n_{\text{tr}}+\tau,\text{tr}}$.

5 Implementation in the `smoots` package

The `smoots` package (smoothing time series, version 1.1.4, Feng et al., 2021) in R (see R Core Team, 2021) consists of fourteen directly applicable functions, four S3 methods and four datasets. The package allows for the data-driven estimation of nonparametric trend functions or their derivatives in trend-stationary time series with short-memory errors by means of local polynomial regression. With package version 1.1.4, further functions were implemented for testing the linearity or constancy of trend functions, for forecasting and for backtesting. Moreover, a selection of the built-in functions now uses compiled C++ code that is invoked by the `Rcpp` see Eddelbuettel and François, 2011 and `RcppArmadillo` see Eddelbuettel and Sanderson, 2014 packages for less computation time of the algorithms. Since the data-driven estimation functions of the `smoots` package are discussed in detail in Feng et al., 2020b and in the package's manual, only a brief overview of the newly introduced main functions `confBounds()`, `modelCast()` and `rollCast()` will be given in the following.

The linearity or constancy of the (nonparametric) trend function in an additive component model can now be tested graphically by applying the newly added function `confBounds()`. By passing an object of class `smoots` that was generated by either `msmooth()`, `tsmooth()` or `dsmooth()` to the argument `obj`, an (approximately) unbiased estimate of the trend or its derivatives alongside the corresponding confidence bounds is obtained. For `plot = TRUE`, a graphic of these series is created. For the trend, different parametric regression lines can be selected to be shown alongside the trend and its confidence bounds by the argument `p`, where 0, 1, 2, 3 are valid options that define the order of polynomial of the parametric model. This behavior can be suppressed by setting `showPar = FALSE`. For the first derivative, a horizontal line representing the slope of a linear representation of the trend can be displayed or omitted, whereas a line at value zero can be shown in the plot of the second derivative. Moreover, the argument `alpha` represents the confidence level. A 95% confidence level is the default setting with `alpha`

= 0.95. The last argument is rescale, which is only meaningful, if a plot of the first or second derivative is created. For rescale = TRUE, all values in the plot are rescaled from the interval [0,1] according to the scale of the x-axis.

modelCast() is the main forecasting function for Semi-ARMA models within smoots. It requires the output of either msmooth() or tsmooth() as an input and returns point forecasts as well as forecasting bounds. The arguments p and q of the function define the orders of the ARMA model that will be fitted to the stationary part of the series. If both arguments are set to NULL, the best ARMA(P, Q) model for $P, Q = 0, 1, \ldots, 5$ following the Bayesian information criterion will be selected automatically. The forecasting horizon can be set by the argument h. Furthermore, a linear (np.fcast = "lin") and a constant (np.fcast = "const") extrapolation of the estimated trend are available options. To adjust the confidence level of the forecasting intervals, the argument alpha, which specifies α, can be used, so that $[(1-\alpha)100]\%$ forecasting intervals will be determined. Moreover, the bounds can be either bootstrapped (method = "boot") or computed under the normality assumption (method = "norm"). Some additional arguments can be adjusted, if in fact a bootstrap is applied to the data. Users can control the total number of bootstrap iterations via the argument it, which is set to 10,000 iterations by default. By means of the argument n.start, the number of 'burn-in' observations for the simulated ARMA processes within the bootstrap is determined. With pb set to TRUE, a progress bar will be shown in the R console. Moreover, the bootstrap allows for parallel computation by setting the argument cores to a number of processor cores to use. modelCast() usually returns a matrix with the forecasting results. If a bootstrap is applied, setting export.error = TRUE will return a list with the forecasting matrix and another matrix with the bootstrapped predictions errors. The function rollCast() for backtesting Semi-ARMA models shares most of its arguments with modelCast() and thus, most explanations can be skipped to avoid repetition. Nevertheless, the unknown argument K describes the test data size at the end of a given time series and is usually set to K = 5.

6 Application of linearity and stationarity tests

In the following, the three linearity tests proposed in Section 3 are applied to two different time series: the mean monthly temperature changes of the Northern Hemisphere from 1880 to 2018 and the logarithm of the quarterly US-GDP from the first quarter of 1947 to the second quarter of 2019. Both datasets are included within the `smoots` package since version 1.0.1; the former was originally provided by the Goddard Institute for Space Studies of the National Aeronautics and Space Administration, while the latter was obtained from the Federal Reserve Bank of St. Louis. The fitting of Semi-ARMA(1,1) models to these series is demonstrated in Feng et al. (2020b) under usage of the `smoots` package to identify the nonparametric trend function and its derivatives.

In Figures 1(a), (b) and (c), the results of the three graphical tests for linearity for the temperature changes data are illustrated. The red dashed line in each of the figures corresponds to the estimated trend or its first or second derivative, respectively, according to the asymptotically unbiased and normal nonparametric estimator, while the grey and darker grey areas describe the 95%- and the extension to the 99%-confidence intervals. With the solid black lines, the estimated trend and its derivatives are described under consideration of the linearity assumption. To obtain the estimates, the function `confBounds()` of the `smoots` package was called with the options `p = 1` and with both `alpha = 0.95` and `alpha = 0.99`. As input objects `obj`, the output of the `smoots` functions `msmooth()` with its default settings and `dsmooth()` with `d = 1` and `d = 2` and default settings otherwise were generated beforehand.

Since clearly more than 5% of the parametric estimates (blue) lie outside of the 99% confidence intervals in Figures 1(a), (b) and (c), the null hypothesis of a linear trend function can be rejected at the significance level of 5% and 1% according to these three tests. However, according to Figure 1(c), the unbiased estimates of the second derivatives do not deviate too strongly from zero as the expected value of the second derivate in case of linearity for time intervals from 1880 to 1907 and from 1982 to 2018. Nonetheless, all three tests indicate, that the overall trend function in this particular time series is non-linear.

Using identical settings of the function `confBounds()`, the same graphical linearity tests can be conducted for the logarithms of the US-GDP. The results are depicted in

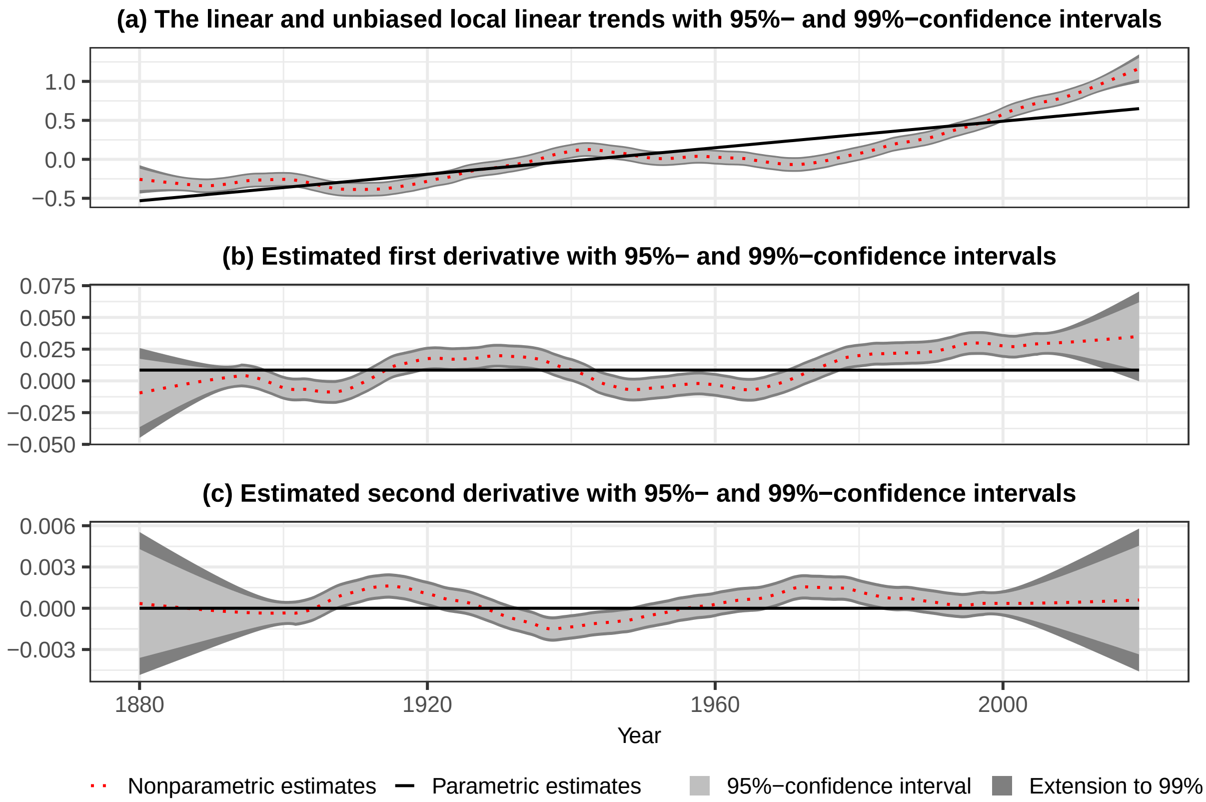

Figure 1: Results of different linearity tests for the temperature changes data.

Figures 2(a), (b) and (c). Note that the nonparametric estimates in Figures 2(b) and (c) differ notably at the right boundary, as the first derivative has a negative slope, whereas it should be positive according to the estimated second derivative. Since for the estimation of the different derivatives distinct bandwidths and therefore diverse numbers of elements for smoothing are used, conflicting results cannot be avoided in some cases. Nonetheless, the uncertainty in the local polynomial estimates increases both with the order of the derivative and at boundary points. We can observe that the confidence intervals for the estimates of the second derivative at the right boundary points also include negative values and therefore, there is not enough evidence to clearly disprove the negativity of the second derivative at those points. Furthermore, regarding the proposed linearity tests, we come to the conclusion once more that a linear trend is not suitable for the data. That is, a linear trend for output growth is a suitable benchmark on average (Jones, 2002). However, there are periods that exhibit a significant difference between the estimated parametric (solid black line) and nonparametric growth rates (dashed red curve). Especially between 1949 and 1965 (the golden age) the growth rates for the US economy were extraordinarily high. As could be also seen from Figure 2(b) with the bursting of the dot-com bubble in 2000, a period of significantly weak growth was introduced (Fritz et al., 2019b).

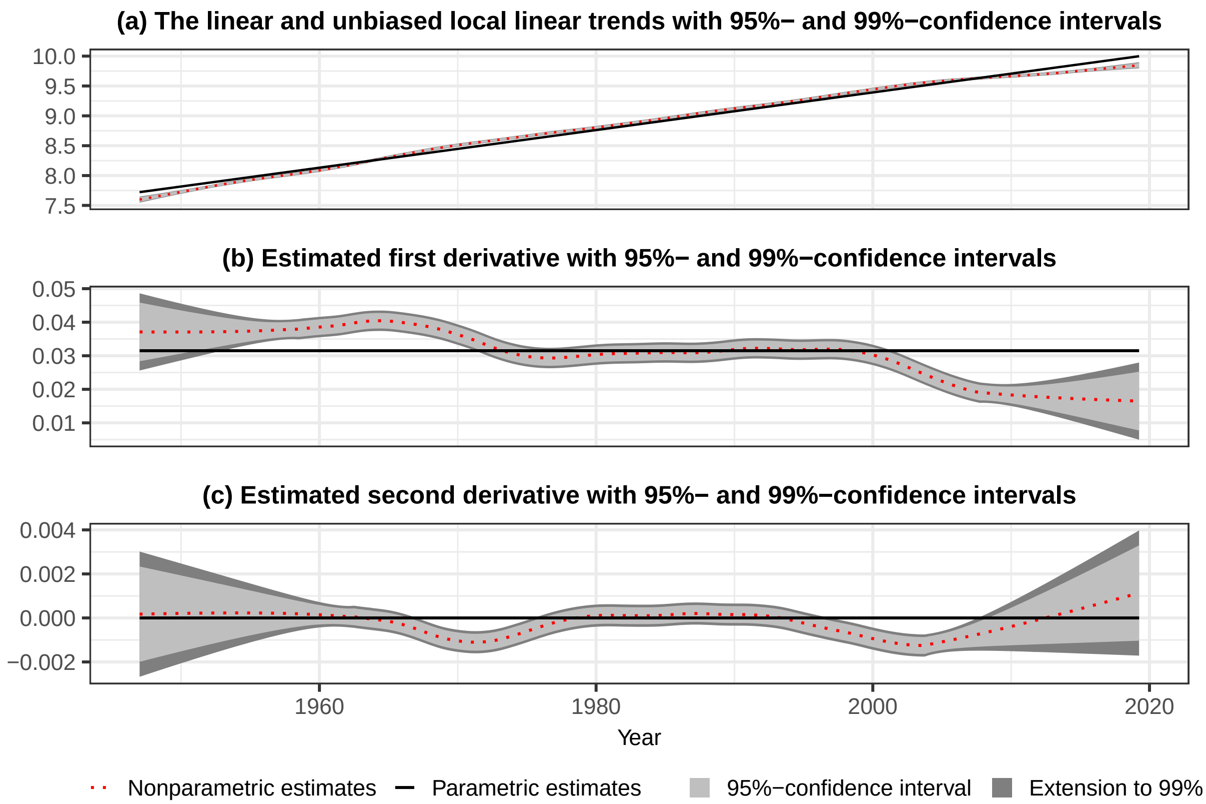

Figure 2: Results of different linearity tests for the US-data.

Generally, notwithstanding that local polynomial regression has automatic boundary correction, the estimation quality is poorer at a boundary point in comparison to the interior. Therefore, the difference between the upper and the lower confidence bounds increases toward the boundaries in Figures 1 and 2. In particular, this effect becomes more apparent for higher orders of derivative of the trend m so that by applying all three methods different test outcomes with regard to the linearity of the trend might be possible. In such a case, a final decision should be made while taking into account both the time series and research question at hand. However, we also propose that with an increasing order of derivative less importance should be attached to the respective test. Aside from linearity tests, the function `confBounds()` can also be applied to conduct a simple stationarity test. Exemplarily, the returns of the German Stock Index (DAX) and to the CBOE Volatility Index (VIX) are considered. Both series are also implemented in the `smoots` package and contain daily observations from January 1990 to July 2019.

Originally, the data was obtained from Yahoo Finance.

To test both series for stationarity, the logarithm of the squared centralized DAX returns and the logarithm of the VIX closing price series need to be calculated, as they follow equation (1). The trend in the transformed DAX returns was estimated using `msmooth()` with the settings `alg = "A"` and `p = 3`, whereas the same function with `alg = "A"` and `p = 1` was called for the VIX data. The corresponding output was then inserted into the function `confBounds()` with further adjustments to `p = 0` and both `alpha = 0.95` and `alpha = 0.99`. Due to `p = 0` a constant mean is estimated from the data instead of a linear trend.

In Figures 3(a) and (b), the graphical stationarity tests for the DAX and the VIX data are depicted, respectively. Since in each case clearly more than 5% of the solid black line is outside of the 99%-confidence bounds, the null hypothesis of the transformed data fluctuating around a constant mean and therefore being stationary can be rejected. Consequently, the DAX returns and the VIX prices admit a nonconstant, nonparametric scale function and are themselves not stationary.

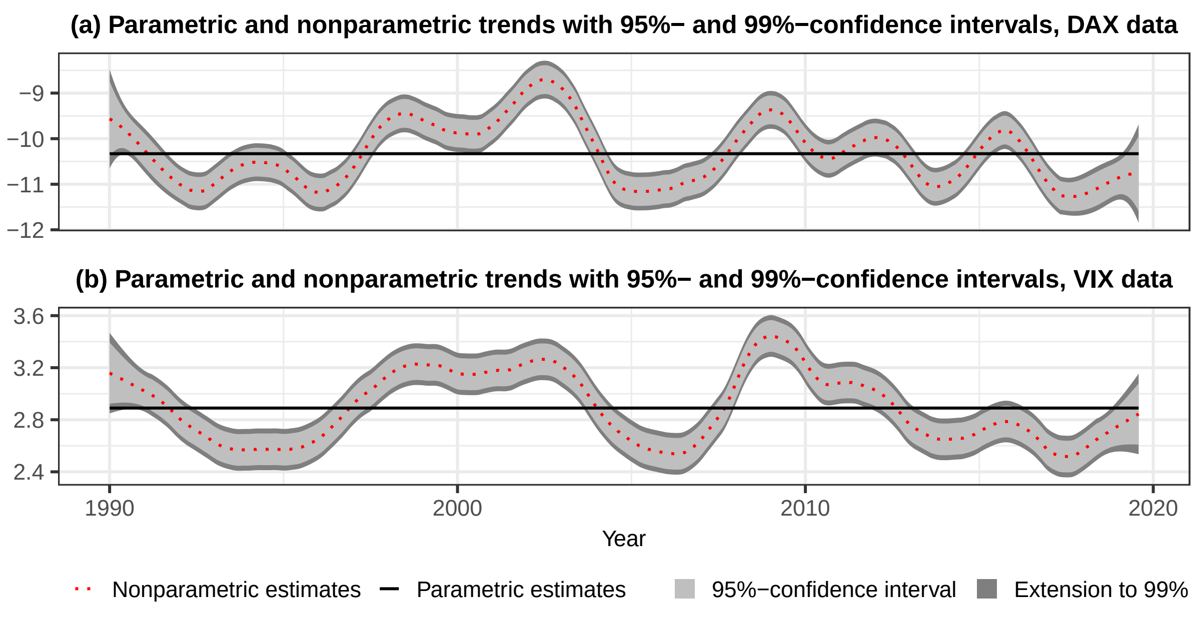

Figure 3: Result of a stationarity test for the DAX and VIX data.

7 Application of the forecasting techniques

7.1 Point forecasts and forecasting intervals

Here, the semiparametric forecasting approaches as well as the backtesting methods described in Section 4 will be employed by means of the functions `modelCast()` and `rollCast()` of the `smoots` package. In Figure 4, predictions of the temperature changes series, the US Log-GDP, the logarithm of the squared DAX returns and the logarithm of the VIX prices are displayed. For simplicity, a Semi-ARMA(1,1) model is assumed for each time series, while the underlying trend functions are estimated by calling the function `msmooth()` with its standard settings. The only exception is the DAX case, for which a local cubic trend (`p = 3`) is fitted.

In Figure 4(a), point predictions as well as 95%- and 99%-forecasting intervals, which were obtained under the normality assumption, are illustrated for the temperature changes of the Northern Hemisphere for the consecutive five future months. The trend was predicted by linear extrapolation as explained in Section 4.1. Consequently, we predict a temperature change of 1.1041°C for January 2019. In the following months, the point predictions slowly increase in value until May 2019 with a change of 1.1375°C. Moreover, the forecasting intervals are symmetric about the predicted values and increase in length for future time points, as the uncertainty with respect to the predicted points increases.

The point forecasts and the 95%-forecasting intervals obtained both under the normality assumption and with the bootstrapping method presented in Section 4.2 are depicted in Figure 4(b) for the US Log-GDP. Once more, the trend is extrapolated linearly and an increase in the point predictions is observable for the five future time points. Furthermore, the obtained 95%-forecasting intervals are almost identical between both methods for this specific time series. For the third quarter of 2019, $[9.8396, 9.8730]$ and $[9.8351, 9.8713]$ are the calculated intervals under normality and with the bootrapping approach, respectively, while their length steadily increases over time until they are computed as $[9.8296, 9.9065]$ and $[9.8269, 9.9060]$ for the third quarter of 2020. Thus, the normality assumption appears to hold approximately in this case.

In Figures 4(c) and (d), different point forecasts for the transformed DAX and VIX series are illustrated, respectively. More specifically, the dashed red lines correspond to the total forecasts, if the trend is extrapolated linearly, while a constant extrapolation of the trend is considered for the predictions represented by the dotted purple lines. Obviously, both approaches give similar results for time points close to the last observation time point. Nevertheless, with a long term perspective, the dotted purple lines will eventually converge towards a fixed value, whereas the dashed red lines keep increasing. Under consideration that the original return series, to which the transformed series displayed in Figure 4(c) belongs, is assumed to follow a Semi-Log-GARCH(1,1) model, a constant extrapolation of the trend is usually more suitable, as the unconditional variance in return series is usually roughly constant for short periods of time. In contrast, under a Semi-Log-ACD model, usually a linear trend extrapolation is preferred on the transformed data as in Figure 4(d), because non-negative financial data often show clearer changes in the overall level and a constant extrapolation would over- or underestimate the future development of the true underlying trend.

7.2 Backtesting of Semi-ARMA models

The function `rollCast()` of the `smoots` package is used to apply the backtesting methods of Section 4.3 to test a Semi-ARMA(1,1) model with regard to the temperature changes as well as the US Log-GDP series. For each series, the function is called twice to obtain 95%- (`alpha = 0.95`) and 99%-forecasting intervals (`alpha = 0.99`). Normality of the point predictions is assumed for the first dataset, whereas bootstrapped forecasting intervals are computed for the Log-GDP. Moreover, the estimated trend functions are extrapolated linearly in both cases and the function `msmooth()` with its standard settings is automatically called within `rollCast()`. The results are observable in Figures 5(a) and (b), respectively. As illustrated, none of the true observations lies outside of the forecasting intervals for neither the temperature changes nor the US Log-GDP series, which is a first indicator that the quality of the forecasting intervals is suitable. Note that the bootstrapped forecasting intervals are not symmetric with respect to the Log-GDP point predictions. Furthermore, in both cases, the rolling point forecasts closely follow the true observations.

To compare the performance of the Semi-ARMA model (SEARMA) not only to the

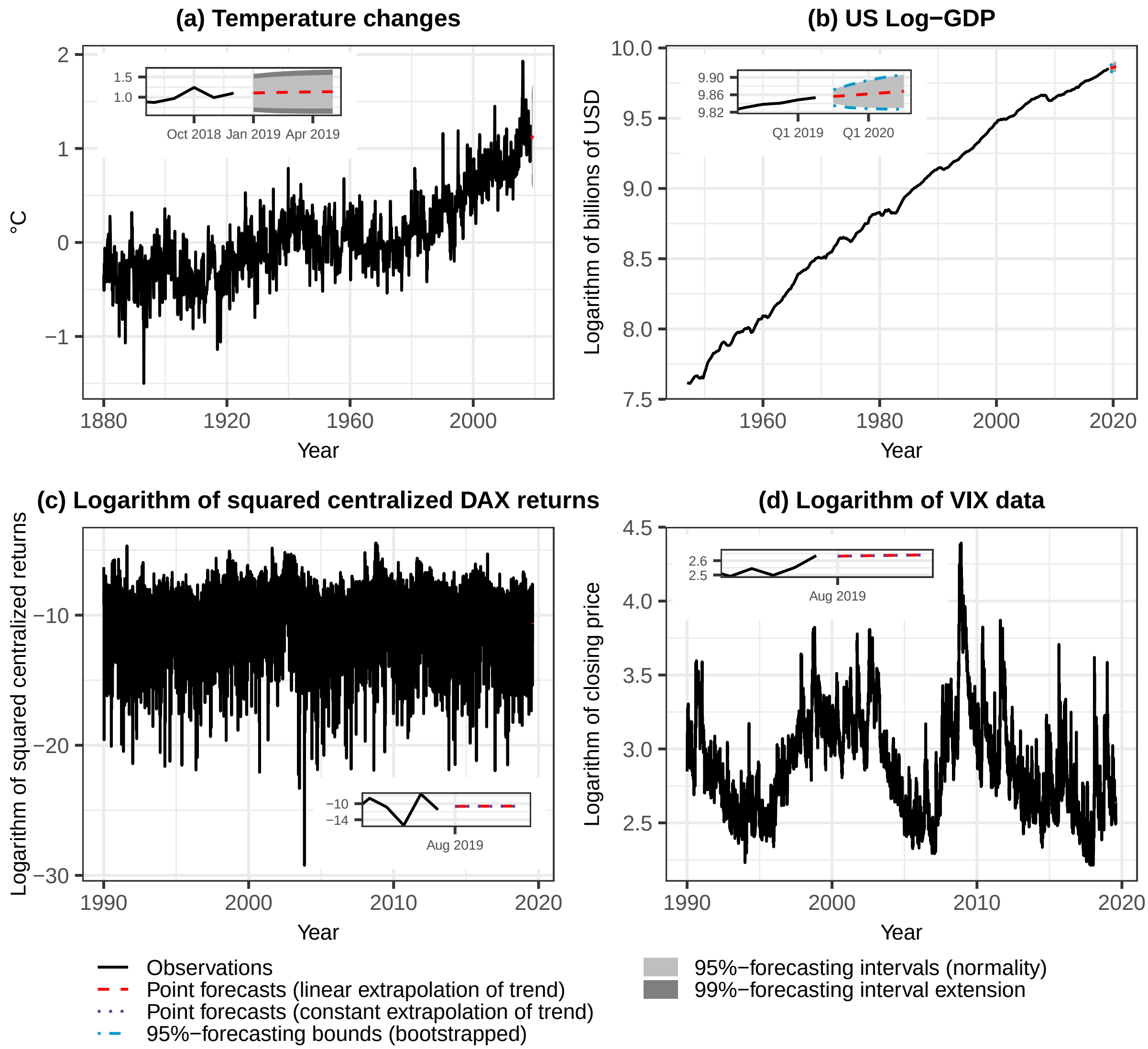

Figure 4: Forecasting of different time series data with Semi-ARMA(1,1) models.

naive approach, we also compute the MASE as well as the RMSSE for a variety of established models, which are not included in the `smoots` package. A combination of either the filter by Hamilton (2018) (HFARMA) or a cubic trend together with an ARMA(1,1) model (CTARMA) for the residuals and a simple ARMA(1,1) model (SIARMA) are considered. In addition to the temperature changes (TD) and the Log-GDP (LGDP) data, the observed monthly employment-population-ratio of the USA from January 1948 to December 2019 (EPR) and the total public debt of the USA in percent of their GDP from the first quarter of 1966 to the fourth quarter of 2019 (LTPD) are inspected. The EPR and LTPD data were collected from the website of the Federal Reserve Bank of St.

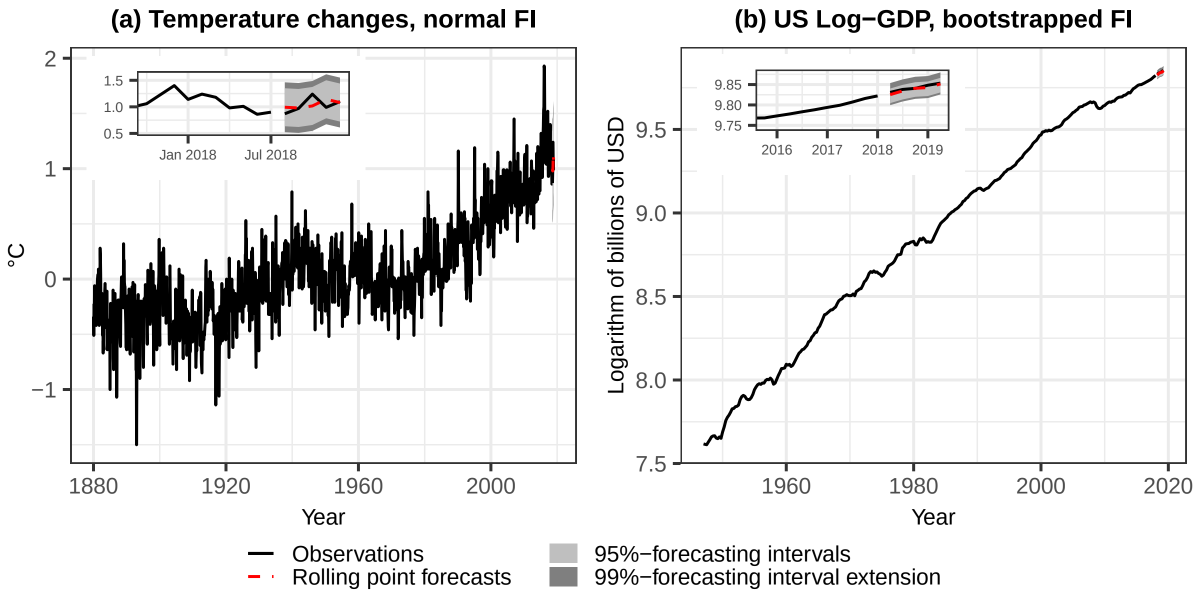

Figure 5: Backtesting Semi-ARMA(1,1) models for different time series data.

Louis. Note that for the filter by Hamilton (2018), we consider the horizon parameter $h_h = 8$ with four regressors for quarterly data and $h_h = 24$ with 12 regressors for monthly observations as proposed by the author.

In Figure 6, the different test series are displayed alongside the various rolling point forecasts. The corresponding MASE and RMSSE values can be found in Table 1. For the datasets TD and EPR, the SEARMA model results in the smallest MASE and RMSSE among all considered models, while for the LTPD data it also exhibits the smallest RMSSE. Furthermore, in all remaining cases, the forecasts according to the SEARMA model are nearly as accurate as those of the other models, e.g. the MASE and RMSSE following the SEARMA model are calculated as 0.3400 and 0.3293 for the LGDP data, respectively, whereas these criteria are derived as 0.2408 and 0.2300 with respect to the most accurate model, i.e. CTARMA. Moreover, we find that for the CTARMA model, the assumption of a cubic trend clearly does not hold except for the LGDP data, while for the SIARMA model, the stationarity is violated in all four cases. Since, moreover, the model HFARMA provides notably worse forecasts in some cases as can be seen in Figures 6(c) and (d), we recommend the usage of the Semi-ARMA model for trend-stationary time series, especially if the trend's structure is complex, as it is the only one of the four inspected models that provides both suitable and stable forecasts for each of the studied examples.

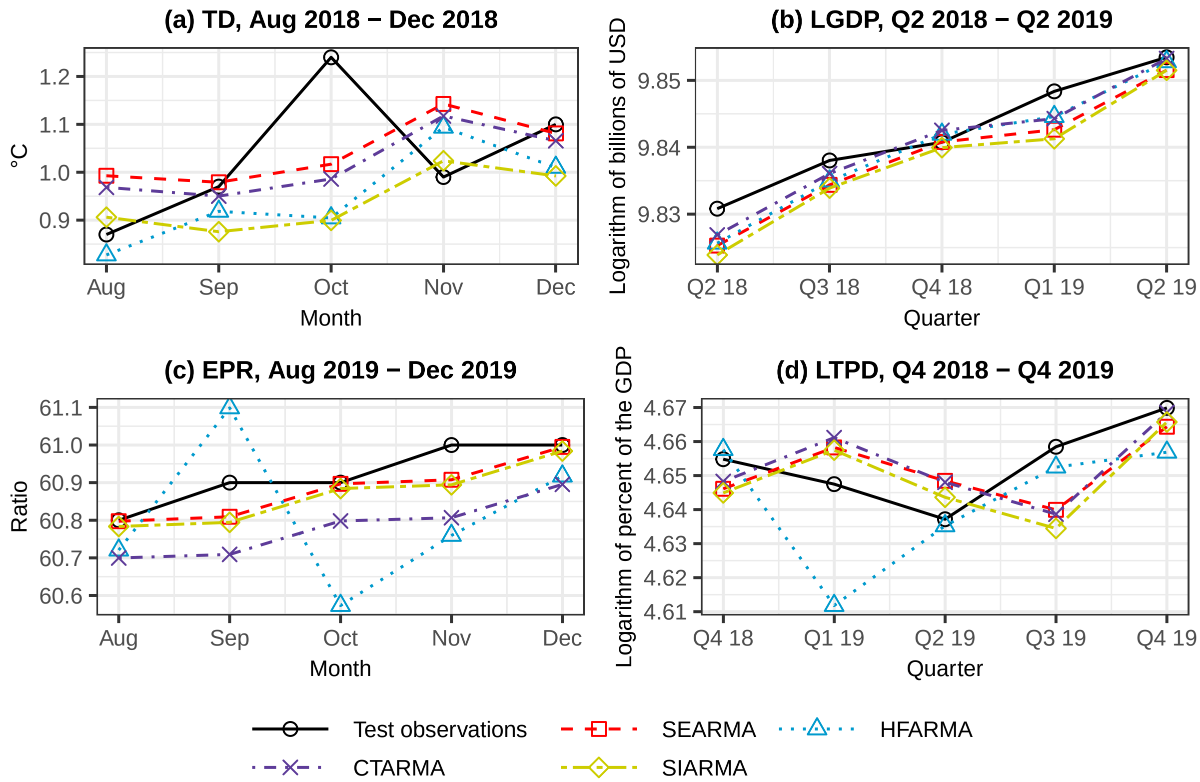

Figure 6: Rolling one-step point forecasts for different test data using assorted models.

Table 1: Accuracy measures for rolling one-step forecasts according to selected models.

Fitted model	MASE				RMSSE			
	TD	LGDP	EPR	LTPD	TD	LGDP	EPR	LTPD
SEARMA	0.6777	0.3400	0.2668	0.8089	0.6252	0.3293	0.2813	0.6500
HFARMA	0.7792	0.2800	1.4442	0.8765	0.6933	0.2658	1.1854	0.9527
CTARMA	0.6874	0.2408	0.9499	0.7704	0.6373	0.2300	0.7033	0.6722
SIARMA	0.7883	0.4207	0.3570	0.8037	0.7828	0.4018	0.3298	0.7122

8 Concluding remarks

In this paper, the basic concepts of the algorithms built into the `smoots` package (version 1.1.4) are discussed. Moreover, new methods for testing the stationarity or linearity of time series that follow an additive component model are presented and the underlying theory is developed. Point forecasts for Semi-ARMA models as well as respective forecasting intervals in case of Gaussian errors and a bootstrapped alternative are examined and simple backtesting approaches for these models are introduced. Furthermore, Semi-Log-GARCH and Semi-Log-ACD models are summarized and the restraint on the log-volatility intercept in order to satisfy unit variance in the parametric part of a Semi-Log-GARCH model is derived. The application of the newly built-in functions within the `smoots` package is elaborated in detail. Actual data examples illustrate the usage of the testing and forecasting functions of the `smoots` package for different types of non-stationary time series and prove that the built package returns correct results with regard to the methods in this paper. Moreover, the conducted comparison of the forecasting accuracy of assorted models indicates that the Semi-ARMA model performs approximately as good as other established models in cases with a simple trend function and better in the presence of a complex trend. Throughout, however, the Semi-ARMA predictions are the most stable in quality. Since the current version of the package is only applicable in cases with short-memory errors, further extensions of the existing algorithms and functions for long-memory need to be developed. In addition, further extensions of the methods and the package with regard to seasonal time series should be studied, as many time series exhibit seasonality. Aside from this, the conversion of more of the existing R code within the package to equivalent C++ code is advisable for future package versions, as a decrease in computation time is expected.

Computational details

The numerical results in this paper were obtained using R 4.3.1 with the `smoots` 1.1.4 package. The illustrations were created using the `ggplot2` 3.4.3 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

The data used were downloaded from different public sources as indicated in the contexts. We are grateful to the CRAN-network for great help during the publication of the R package `smoots`.

Funding

This work was supported by the German DFG project GZ-FE-1500-2-1.

Disclosure statement

The authors report there are no competing interests to declare.

References

Alonso, A. M., Peña, D. and Romo, J. (2002). Forecasting Time Series with Sieve Bootstrap. In: *Journal of Statistical Planning and Inference* 100.1, pp. 1–11. DOI: `10.1016/S0378-3758(01)00092-1`.

Bauwens, L. and Giot, P. (2000). The Logarithmic ACD model: An Application to the Bid-Ask Quote Process of Three NYSE Stocks. In: *Annales d'Economie et de Statistique*, pp. 117–149. DOI: `10.2307/20076257`.

Beran, J. and Feng, Y. (2002). Local Polynomial Fitting with Long-Memory, Short-Memory and Antipersistent Errors. In: *Annals of the Institute of Statistical Mathematics* 54.2, pp. 291–311. DOI: `10.1023/A:1022469818068`.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. In: *Journal of Econometrics* 31.3, pp. 307–327. DOI: `10.1016/0304-4076(86)90063-1`.

Brockwell, P. J. and Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. 3rd ed. Cham, Switzerland: Springer.

Bühlmann, P. (1996). Locally adaptive lag-window spectral estimation. In: *Journal of Time Series Analysis* 17.3, pp. 247–270. DOI: `10.1111/j.1467-9892.1996.tb00275.x`.

– (1997). Sieve Bootstrap for Time Series. In: *Bernoulli* 3.2, pp. 123–148. DOI: `10.2307/3318584`.

Cleveland, W. S. and Loader, C. (1996). Smoothing by Local Regression: Principles and Methods. In: *Statistical Theory and Computational Aspects of Smoothing*. Ed. by W. Härdle and M. G. Schimek. 1st ed. Heidelberg, Germany: Physica-Verlag, pp. 10–49.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ Integration. In: *Journal of Statistical Software* 40.8, pp. 1–18. DOI: `10.18637/jss.v040.i08`.

Eddelbuettel, D. and Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. In: *Computational Statistics and Data Analysis* 71, pp. 1054–1063. DOI: `10.1016/j.csda.2013.02.005`.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. In: *Econometrica: Journal of the Econometric Society*, pp. 987–1007. DOI: `10.2307/1912773`.

Engle, R. F. and Russell, J. R. (1998). Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data. In: *Econometrica*, pp. 1127–1162. DOI: `10.2307/2999632`.

Fan, J. and Gijbels, I. (1992). Variable Bandwidth and Local Linear Regression Smoothers. In: *The Annals of Statistics* 20.4, pp. 2008–2036. DOI: 10.1214/aos/1176348900.

– (1996). *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. 1st ed. New York, NY: Chapman and Hall / CRC.

Feng, Y., Gries, T. and Fritz, M. (2019). Data-Driven Local Polynomial for the Trend and Its Derivatives in Economic Time Series. Paderborn University. Working Paper.

– (2020a). Data-Driven Local Polynomial for the Trend and Its Derivatives in Economic Time Series. In: *Journal of Nonparametric Statistics* 32.2, pp. 510–533. DOI: 10.1080/10485252.2020.1759598.

Feng, Y., Gries, T., Letmathe, S. and Schulz, D. (2020b). The smoots Package in R for Semiparametric Modeling of Trend Stationary Time Series. Paderborn University. Preprint.

Feng, Y. and Heiler, S. (2009). A simple bootstrap bandwidth selector for local polynomial fitting. In: *Journal of Statistical Computation and Simulation* 79.12, pp. 1425–1439. DOI: 10.1080/00949650802352019.

Feng, Y., Letmathe, S. and Schulz, D. (2021). *smoots: Nonparametric Estimation of the Trend and Its Derivatives in TS*. R package version 1.1.1. URL: https://CRAN.R-project.org/package=smoots.

Feng, Y. and Zhou, C. (2015). Forecasting financial market activity using a semiparametric fractionally integrated Log-ACD. In: *International Journal of Forecasting* 31.2, pp. 349–363. DOI: 10.1016/j.ijforecast.2014.09.001.

Forstinger, S. (2018). Modelling and Forecasting Financial and Economic Time Series Using Different Semiparametric ACD Models. PhD thesis. Paderborn, Universität Paderborn.

Francq, C. and Zakoïan, J.-M. (2006). Linear-Representation Based Estimation of Stochastic Volatility Models. In: *Scandinavian Journal of Statistics* 33.4, pp. 785–806. DOI: 10.1111/j.1467-9469.2006.00495.x.

Francq, C. and Zakoïan, J.-M. (2019). *GARCH Models: Structure, Statistical Inference and Financial Applications*. Hoboken, New Jersey, USA: John Wiley & Sons.

Fritz, M., Forstinger, S., Feng, Y. and Gries, T. (2018). Forecasting Economic Growth Processes for Developing Economies. Paderborn University. Preprint.

Fritz, M., Gries, T. and Feng, Y. (2019a). Growth Trends and Systematic Patterns of Booms and Busts-Testing 200 Years of Business Cycle Dynamics. In: *Oxford Bulletin of Economics and Statistics* 81.1, pp. 62–78. DOI: 10.1111/obes.12267.

– (2019b). Secular stagnation? Is there statistical evidence of an unprecedented, systematic decline in growth? In: *Economics Letters* 181, pp. 47–50. DOI: 10.1016/j.econlet.2019.04.021.

Gasser, T., Kneip, A. and Köhler, W. (1991). A Flexible and Fast Method for Automatic Smoothing. In: *Journal of the American Statistical Association* 86.415, pp. 643–652. DOI: 10.2307/2290393.

Geweke, J. (1986). Comment On: Modelling the Persistence of Conditional Variances. In: *Econometric Reviews* 5, pp. 57–61. DOI: 10.1080/07474938608800097.

Hamilton, J. D. (2018). Why You Should Never Use the Hodrick-Prescott Filter. In: *Review of Economics and Statistics* 100.5, pp. 831–843.

Hastie, T. and Loader, C. (1993). Local Regression: Automatic Kernel Carpentry. In: *Statistical Science* 8.2, pp. 120–129. DOI: 10.1214/ss/1177011002.

Hjellvik, V., Yao, Q. and Tjøstheim, D. (1998). Linearity testing using local polynomial approximation. In: *Journal of Statistical Planning and Inference* 68.2, pp. 295–321. DOI: 10.1016/S0378-3758(97)00146-8.

Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice.* 3rd ed. Melbourne, Australia: OTexts. URL: https://otexts.com/fpp3/.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. In: *International Journal of Forecasting* 22.4, pp. 679–688. DOI: 10.1016/j.ijforecast.2006.03.001.

Jones, C. I. (2002). Sources of U.S. Economic Growth in a World of Ideas. In: *American Economic Review* 92.1, pp. 220–239. DOI: 10.1257/000282802760015685.

Lu, X. and Wang, L. (2020). Bootstrap prediction interval for ARMA models with unknown orders. In: *REVSTAT–Statistical Journal* 18.3, pp. 375–396.

Milhøj, A. (1987). *A Multiplicative Parametrization of ARCH Models.* Copenhagen, Denmark: University of Copenhagen, Department of Statistics.

Nelson Jr., H. L. and Granger, C. (1979). Experience with using the Box-Cox transformation when forecasting economic time series. In: *Journal of Econometrics* 10.1, pp. 57–69. DOI: 10.1016/0304-4076(79)90064-2.

Pan, L. and Politis, D. N. (2016). Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions. In: *Journal of Statistical Planning and Inference* 177, pp. 1–27. DOI: 10.1016/j.jspi.2014.10.003.

Pantula, S. G. (1986). Modeling the Persistence of Conditional Variances: A Comment. In: *Econometric Reviews* 5, pp. 79–97.

Pascual, L., Romo, J. and Ruiz, E. (2004). Bootstrap predictive inference for ARIMA processes. In: *Journal of Time Series Analysis* 25.4, pp. 449–465. DOI: 10.1111/j.1467-9892.2004.01713.x.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Ruppert, D. and Wand, M. P. (1994). Multivariate Locally Weighted Least Squares Regression. In: *The Annals of Statistics* 22.3, pp. 1346–1370. DOI: 10.1214/aos/1176325632.

Sucarrat, G., Grønneberg, S. and Escribano, A. (2016). Estimation and Inference in Univariate and Multivariate Log-GARCH-X Models When the Conditional Density is Unknown. In: *Computational Statistics & Data Analysis* 100, pp. 582–594. DOI: 10.1016/j.csda.2015.12.005.

Appendix

A Proofs

Proof of Proposition 1:

Proof. Given the formula of a Semi-Log-GARCH model (9) and the definition of the moments $\mathrm{E}(\epsilon_t^{2m})$ given in (10) for $m = 1, 2, \ldots,$

$$\mathrm{E}(\epsilon_t^2) = \mathrm{Var}(\epsilon_t) = \exp(c) \prod_{i=1}^{\infty} \mu_\eta(\delta_i)$$

can be derived. Since $\mathrm{Var}(\epsilon_t) = 1$ is assumed,

$$\begin{aligned}
&\mathrm{Var}(\epsilon_t) \stackrel{!}{=} 1 \\
\Leftrightarrow\ &\exp(c) \prod_{i=1}^{\infty} \mu_\eta(\delta_i) = 1 \\
\Leftrightarrow\ &c + \ln\left[\prod_{i=1}^{\infty} \mu_\eta(\delta_i)\right] = 0 \\
\Leftrightarrow\ &c = -\ln\left[\prod_{i=1}^{\infty} \mu_\eta(\delta_i)\right] \\
\Leftrightarrow\ &\omega = -\ln\left[\prod_{i=1}^{\infty} \mu_\eta(\delta_i)\right]\left[1 - \sum_{j=1}^{P^*}(\alpha_j + \beta_j)\right]
\end{aligned}$$

is a straightforward calculation. □

B R codes

For simplicity, only the code implemented for selected examples and confidence levels will be shown. Moreover, rescaling derivative estimates via the `smoots` function `rescale()` is omitted.

Installation of the R package `smoots`:

```
R> install.packages(smoots); library(smoots)
```

B.1 Codes for linearity and stationarity tests

Code for example 2:

```
R> Xt2 <- log(gdpUS$GDP); est2.trend <- msmooth(Xt2)
R> est2.d1 <- dsmooth(Xt2); est2.d2 <- dsmooth(Xt2, d = 2)
R> cb2.trend <- confBounds(est2.trend, alpha = 0.95, plot = FALSE)
R> cb2.d1 <- confBounds(est2.d1, alpha = 0.95, plot = FALSE)
R> cb2.d2 <- confBounds(est2.d2, alpha = 0.95, plot = FALSE)
```

Code for example 3:

```
R> close <- dax$Close; return <- diff(log(close))
R> mean.rt <- mean(return); rt <- return - mean.rt; Xt3 <- log(rt^2)
R> est3 <- msmooth(Xt3, alg = "A", p = 3)
R> cb3 <- confBounds(est3, p = 0, alpha = 0.95, plot = FALSE)
```

B.2 Codes for point predictions with prediction intervals

Code for example 2:

```
R> fc2.norm <- modelCast(est2.trend, method = "norm", p = 1, q = 1, h = 5)
R> set.seed(1)
R> fc2.boot <- modelCast(est2.trend, method = "boot", p = 1, q = 1, h = 5,
 +   cores = NULL)
```

Code for example 3:

```
R> fc3.lin <- modelCast(est3, np.fcast = "lin", p = 1, q = 1, h = 5)
R> fc3.const <- modelCast(est3, np.fcast = "const", p = 1, q = 1, h = 5)
```

B.3 Codes for backtesting

Code for example 2:

```
R> set.seed(1)
R> backtest2 <- rollCast(Xt2, alpha = 0.95, method = "boot", p = 1, q = 1,
 +   cores = NULL, plot = FALSE)
```

Recent discussion papers

2026-01	Dominik Schulz Yuanhua Feng Thomas Gries Marlon Fritz Sebastian Letmathe	Diagnosing the trend and bootstrapping the forecasting intervals using a semiparametric ARMA
2025-06	Sarah Kühn Nadja Stroh-Maraun	Community Costs in Neighborhood Help Problems
2025-05	Sarah Kühn	Child Care Allocation Mechanisms: Navigating Incomplete Preference Elicitation
2025-04	Sarah Kühn Papatya Duman Britta Hoyer Thomas Streck Nadja Stroh-Maraun	Non-induced Preferences in Matching Experiments
2025-03	Marlon Fritz Thomas Gries Lukas Wiechers	Growth with Mismatch: Theory and Evidence from TFP Estimates
2025-02	Lukas Wiechers	A Real-Time Analysis of Fundamentals and Bubbles in the S&P 500
2025-01	Claus-Jochen Haake Martin R. Schneider	An Axiomatization of the Banzhaf Index to Measure Influence in Qualitative Comparative Analysis
2024-05	Iuliia Myroshnychenko Thomas Gries	Application of a dynamic welfare-centered energy transition model
2024-04	Claus-Jochen Haake Thomas Streck	A Measure for Contestedness of a Two-Person Bargaining Problem
2024-03	Youssouf Sangare Edem Adotey	Institutions and challenges to development: An Analysis of Economic Growth, Governance, and Development in Sub-Saharan Africa
2024-02	Papatya Duman Claus-Jochen Haake	Size Reduction Reform in German Parliament: a game theoretic analysis of power indices in the Bundestag
2024-01	Iuliia Myroshnychenko Thomas Gries	Developing a dynamic welfare-centered energy transition model: methodological challenges and perspectives
2023-03	Yuanhua Feng Thomas Gries Sebastian Letmathe	FIEGARCH, modulus asymmetric FILog-GARCH and trend-stationary dual long memory time series
2023-02	Claus-Jochen Haake Thomas Streck	Distortion through modeling asymmetric bargaining power
2023-01	Carina Burs	A Model of Cycles and Bubbles under Heterogeneous Beliefs in Financial Markets
2022-06	Marlon Fritz Thomas Gries Lukas Wiechers	An Early Indicator for Anomalous Stock Market Performance
2022-05	Sebastian Letmathe	Data-driven P-Splines under short-range dependence
2022-04	Sebastian Letmathe Yuanhua Feng	An iterative plug-in algorithm for P-Spline regression
2022-03	Claus-Jochen Haake Martin Schneider	Playing games with QCA: Measuring the explanatory power of single conditions with the Banzhaf index