The *smoots* package in R for semiparametric modeling of trend stationary time series

Yuanhua Feng, Thomas Gries, Sebastian Letmathe and Dominik Schulz
Faculty of Business Administration and Economics, Paderborn University

April 28, 2020

Abstract

This paper is an introduction to the new package in R called *smoots* (smoothing time series), developed for data-driven local polynomial smoothing of trend-stationary time series. Functions for data-driven estimation of the first and second derivatives of the trend are also built in. It is first applied to monthly changes of the global temperature. The quarterly US-GDP series shows that this package can also be well applied to semiparametric multiplicative model for non-negative time series via the log-data. Furthermore, we introduced two semiparametric Log-GARCH and Log-ACD models, which can be easily estimated by the *smoots* package. Of course, this package applies to suitable time series from any research area. The *smoots* package also provides a useful tool for teaching time series analysis, because many practical time series follow an additive or a multiplicative component model.

Keywords: Component models, data-driven smoothing of time series, estimation of derivatives, *smoots* package, Semi-Log-GARCH, Semi-Log-ACD

JEL Codes: C14, C51

1 Introduction

This paper provides an introduction to a new package in R called *smoots* (smoothing time series, version 1.0.1, Feng and Schulz, 2019) for data-driven local polynomial smoothing of trend-stationary time series. This package is developed based on the R codes for the practical implementation of the proposals in Feng et al. (2019). Detailed discussion on the methodology and the development of the algorithms may be found in that work. Here, the main algorithm is a fully data-driven IPI (iterative plug-in, Gasser et al., 1991) approach for estimating the trend under stationary time series errors, where the variance factor in the asymptotically optimal bandwidth is estimated by another IPI-approach for a lag-window estimator of the spectral density following Bühlmann (1996). Numerous sub-algorithms determined by different options, such as the choice of the order of local polynomial, the choice of the kernel weighting functions and the choice of the so-called inflation factors for estimating the bias in the asymptotically optimal bandwidth, are included. Further functions for data-driven estimation of the first and second derivatives of the trend are also developed by adapting the idea of Feng (2007). A DPI (direct plug-in) algorithm for smoothing time series was proposed by Francisco-Fernández et al. (2004) following the proposal of Ruppert et al. (1995). The algorithms included in the *smoots* package differ to their proposal in several ways. 1. The IPI-algorithm is usually superior to the DPI (see Beran et al., 2009 for detailed discussion in models with i.i.d. errors). 2. In Francisco-Fernández et al. (2004) the variance factor is estimated under an AR(1) model, this is usually a misspecification. And 3. Data-driven estimation of the derivatives are also included in the current package. Moreover, the user can obtain any estimate with fixed bandwidths chosen beforehand. A function for kernel smoothing is also built in for comparison.

The methodological and computational background for this package is summarized briefly. For further details we refer the reader to Feng et al. (2019) and references therein. Our focus is to illustrate the wide application spectrum of the *smoots* package for semi-parametric modeling of time series. We propose to estimate the nonparametric trend using the current package in the first stage and to fit e.g. an ARMA model to the residuals with the standard *arima* function in R. This idea is first shown with monthly Northern Hemisphere temperature changes. Then the proposal is applied to the log-transformation of the

quarterly US-GDP data using a semiparametric log-local-linear growth model. Moreover, two new semiparametric models for financial time series are proposed to show further application of this package. Firstly, a Semi-Log-GARCH model is defined by introducing a scale function into the Log-GARCH (logarithmic GARCH, Pantula, 1986, Geweke, 1986 and Milhøj, 1988). The latter is a special extension of the well-known GARCH (generalized autoregressive conditional heteroskedasticity) model of Engle (1982) and Bollerslev (1986) and is just a slightly restricted ARMA model for the log-transformation of the squared returns. The usefulness of the Log-GARCH is recently rediscovered by Francq et al. (2013). The application of this proposal is illustrated by the DAX returns. Secondly, a Semi-Log-ACD model is proposed as an extension of the Type I Log-ACD (Bauwens and Giot, 2000), which is closely related to the Semi-Log-GARCH. Like the ACD (autoregressive conditional duration, Engle and Russell, 1998) model, the Semi-Log-ACD can be applied to different non-negative financial data, such as realized volatilities and trading volumes. In this paper, this new model is applied to the CBOE Volatility Index (VIX). Datasets for all of those examples are built in the proposed package. Of course, the above mentioned proposals can be applied to any suitable time series from other research areas. In addition, the *smoots* package provides a useful tool for teaching time series analysis, because it helps a lecturer to obtain automatically detrended real data examples to show the application of parametric time series models.

The methods and IPI-algorithms are summarized in Sections 2 and 3. The usage of the R functions is described in Section 4. Section 5 illustrates the simple application of those functions. The Semi-Log-GARCH and Semi-Log-ACD are investigated in Sections 6 and 7 with examples. Section 8 concludes. The used R codes are put in the appendix.

2 Local polynomial regression for time series

The main algorithm of the *smoots* package is a fully automatic non-parametric procedure for estimating a deterministic trend in an additive time series model with observations y_t, $t = 1, ..., n$. The data under consideration can e.g. be from environmental statistics, economics or finance. The basic nonparametric time series model is defined by

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

where $x_t = t/n$ denotes the rescaled time, m is a smooth function and ξ_t is a zero mean stationary process. This model defines a nonparametric approach for trend-stationary time series. Let $\gamma_\xi(\tau)$ denote the acf (autocovariances) of ξ_t, it is assumed that $\sum_{k=0}^{\infty}(\tau+1)^4|\gamma_\xi(\tau)| < \infty$, i.e. $\gamma_\xi(\tau)$ converge to zero quickly enough. Under this regularity condition a data-driven IPI-algorithm for estimating the variance factor in the asymptotically optimal bandwidth can be developed following Bühlmann (1996). If ξ_t follows a stationary ARMA model, this condition is automatically fulfilled. However, models with long-memory errors are excluded by this assumption.

In the following necessary properties of local polynomial estimation of $m^{(\nu)}$, the ν-th derivative of m, will be summarized briefly. The local polynomial estimate of $m^{(\nu)}(x)$ is obtained by minimizing

$$Q = \sum_{t=1}^{n}\left\{y_t - \sum_{j=0}^{p} b_j(x)(x_t - x)^j\right\}^2 W\left(\frac{x_t - x}{h}\right), \tag{2}$$

where h is the bandwidth, W is a second order kernel function with compact support [-1, 1] and is used here as the weighting function, and p is the order of polynomial. It is assumed that $p-\nu$ is odd. We have $\hat{m}^{(\nu)}(x) = \nu!\hat{b}_\nu(x)$, which is asymptotically equivalent to some kernel regression with a k-th order kernel $K(u)$ and automatic boundary correction, where $k = p+1$. In this paper, the following weighting functions are considered:

$$W(u) = C_\mu(1-u^2)^\mu \mathbb{I}_{[-1,1]}(u), \tag{3}$$

where C_μ is a standardization constant that is indeed irrelevant for calculating the weights in (3) and $\mu = 0$, 1, ..., is a smoothness parameter. For automatic bandwidth selection using the *smoots* package, only $\mu = 0, 1, 2$ and 3 are allowed, corresponding to the use of the uniform, Epanechnikov, bisquare and triweight kernels, respectively.

An IPI-algorithm is developed based on the asymptotically optimal bandwidth h_A, which is the minimizer of the AMISE (asymptotic integrated squared error):

$$\text{AMISE}(h) = h^{k-\nu}\frac{I[m^{(k)}]\beta^2}{(k!)^2} + \frac{2\pi c_f(d_b - c_b)R(K)}{nh^{2\nu+1}}, \tag{4}$$

where $I[m^{(k)}] = \int_{c_b}^{d_b}[m^{(k)}(x)]^2dx$, $\beta = \int_{-1}^{1}u^kK(u)du$, $R(K) = \int_{-1}^{1}K^2(u)du$, $c_f = f(0)$ is the value of the spectral density at the frequency zero and $0 \le c_b < d_b \le 1$ are introduced to reduce the effect of the estimates at the boundary points. Then we have

$$h_A = n^{-\frac{1}{2k+1}} \left(\frac{2\nu+1}{2(k-\nu)} \frac{2\pi c_f (k!)^2 (d_b - cb) R(K)}{I[m^{(k)}]\beta^2} \right)^{\frac{1}{2k+1}}. \tag{5}$$

After estimating and removing the nonparametric trend, any suitable parametric model can be fitted to the residuals for further econometric analysis. For instance, we can assume that ξ_t follows an ARMA model:

$$\xi_t = \varphi_1 \xi_{t-1} + ... + \varphi_r \xi_{t-l} + \psi_1 \varepsilon_{t-1} + ... + \psi_s \varepsilon_{t-s} + \varepsilon_t, \tag{6}$$

where ε_t are i.i.d. (independent identically distributed) innovations. A semiparametric ARMA (Semi-ARMA) model is then defined by (1) and (6).

3 The proposed IPI-algorithms

Two unknown quantities in h_A are c_f and $I[m^{(k)}]$. If they can be estimated suitably, we can insert their estimates into h_A to obtain a selected bandwidth following the plug-in rule. For a given bandwidth, $\hat{I}[m^{(k)}]$ is not affected by the error correlation, which can be estimated using known approaches in the literature proposed for models with independent errors. Therefore, only the estimation of c_f will be discussed in detail.

3.1 Data-driven estimation of c_f

Let $r_{t,\mathrm{V}} = y_t - \hat{m}_{\mathrm{V}}$ denote the residuals obtained from a pilot estimate using a bandwidth h_{V} and denote the sample acf calculated from $r_{t,\mathrm{V}}$ by $\hat{\gamma}(l)$. In this paper we propose to use the following lag-window estimator of c_f:

$$\hat{c}_{f,\mathrm{M}} = \frac{1}{2\pi} \sum_{l=-M}^{M} w_l \hat{\gamma}(l), \tag{7}$$

where $w_l = l/(M + 0.5)$ are weights calculated according to some lag-window with the window-width M. And, M will be selected by the following IPI-algorithm proposed by Feng et al. (2019), which is adjusted from that of Bühlmann (1996).

i) Let $M_0 = [n/2]$ be the initial window-width, where $[\cdot]$ denotes the integer part.

ii) Global steps: Estimate $\int(f(\lambda)^2)d\lambda$ in the j-th iteration following Bühlmann (1996). Denote by $\int f^{(1)}(\lambda)d\lambda$ the integral of the first generalized derivative of $f(\lambda)$. Estimate it using the window-width $M_j' = [M_{j-1}/n^{2/21}]$, the proposal in Bühlmann (1996) and the Bartlett-window. Obtain M_j by inserting this quantity into Eq. (5) in Bühlmann (1996). Carry out this procedure iteratively until convergence is reached or until a maximum of 20 iterations. Denote the selected M by $\hat{M}_G$.

iii) Local adaptation at $\lambda = 0$: Calculate $\int f^{(1)}(\lambda)d\lambda$ again using $M' = [\hat{M}_G/n^{2/21}]$. Obtain the finally selected window-width $\hat{M}$ by inserting the estimates into the formula of the local optimal bandwidth at $\lambda = 0$ in (5) of Bühlmann (1996).

It is proposed to use the Bartlett-window throughout the whole algorithm for simplicity. If $\hat{M}$ converges, it is usually not affected by the starting value M_0. Hence, any $1 \leq M_0 \leq [n/2]$ can be used in the proposed algorithm.

3.2 The IPI-algorithm for estimating m

In this subsection the data-driven IPI-algorithm for selecting the bandwidth for local linear and local cubic estimators $\hat{m}$, with $k = 2$ and 4, respectively, under correlated errors will be described. Note that, although the variance factor c_f can be estimated well from residuals of $\hat{m}$, Feng et al. (2019) showed that the asymptotically optimal bandwidth for estimating c_f should be $\mathrm{CF} \cdot h_A$, not h_A itself, where

$$\mathrm{CF} = \left\{ 2k \left[\frac{2K(0)}{R(K)} - 1 \right] \right\}^{1/(2k+1)} \tag{8}$$

is a correction factor to obtain the asymptotically optimal bandwidth for estimating $\hat{c}_f$ from $\hat{h}$, which is the same as given in Feng and Heiler (2009) and is always bigger than 1. Theoretically, c_f should be estimated from $r_{t,\mathrm{V}}$ obtained with the bandwidth $\mathrm{CF} \cdot \hat{h}$. The proposed main IPI-algorithm for selecting the bandwidth proceeds as follows.

i) Start with an initial bandwidth h_0 given beforehand.

ii) Obtain $r_{t,\mathrm{V}}$ using h_{j-1} or $\mathrm{CF} \cdot h_{j-1}$ and estimate c_f from $r_{t,\mathrm{V}}$ as proposed above.

iii) Let $\alpha = 5/7$ or $5/9$ for $p = 1$, and $\alpha = 9/11$ or $9/13$ for $p = 3$, respectively. Estimate $I[m^{(k)}]$ with $h_{d,j} = h_{j-1}^{\alpha}$ and a local polynomial of order $p_d = p + 2$. We obtain

$$h_j = \left(\frac{[k!]^2}{2k\beta^2} \frac{2\pi \hat{c}_f (d_b - c_b) R(K)}{I[\hat{m}^{(k)}]} \right)^{1/(2k+1)} \cdot n^{-1/(2k+1)}. \tag{9}$$

vi) Increase j by 1. Repetitively carry out Steps *ii*) and *iii*) until convergence is reached or until e.g. $J = 20$ iterations are achieved. Let $\hat{h} = h_j$ be the selected bandwidth.

In the developed package the initial bandwidths $h_0 = 0.1$ for $p = 1$ and $h_0 = 0.2$ for $p = 3$ are used. For $p = 3$, we propose to use $c_b = 1 - d_b = 0.05$ to reduce the effect of the estimates at the boundary points. That is, the bandwidth is selected only using 90% of the observations in the middle part. For $p = 1$, $c_b = 1 - d_b = 0$ are simply used. The bandwidth $h_{d,j} = h_{j-1}^{\alpha}$ for estimating the k-th derivative is roughly fixed using a so-called EIM (exponential inflation method). The first α value is chosen so that $I[\hat{m}^{(k)}]$ and hence $\hat{h}$ will achieve their optimal rates of convergence. Now, the algorithm will be denoted by **AlgA**. If the second α value is used, $\hat{m}^{(k)}$, but not $\hat{h}$, will achieve its optimal rate of convergence. This algorithm is called **AlgB**. And $\hat{h}$ selected by **AlgB** is larger than that by **AlgA**. For further details on those topics we refer the reader to Feng et al. (2019).

3.3 Data-driven estimation of m' and m''

The proposed IPI-algorithm can be easily adapted to select bandwidths for estimating $\hat{m}^{(\nu)}$. In the following, only the cases with $\nu = 1$ or 2 are considered, where c_f is estimated by means of a data-driven pilot estimate $\hat{m}$ of the order p_p, say. Then $m^{(\nu)}$ will be estimated with $p = \nu + 1$ and $k = \nu + 2$. As before, $m^{(k)}$ for calculating the bias factor will be estimated with $p_d = p + 2$ and a correspondingly inflated bandwidth. This leads to the following two-stage procedure.

i) In the first stage $\hat{c}_f$ is obtained by the main IPI-algorithm with $p_p = 1$ or $p_p = 3$.
ii) Then an IPI-procedure as proposed above to select the bandwidth for estimating $m^{(\nu)}$ according to (5) is carried out with fixed $\hat{c}_f$ obtained in i).

Note that $\hat{m}^{(\nu)}$ is asymptotically equivalent to some kernel estimator with boundary correction. Explicit forms of the equivalent kernels for estimating $m^{(\nu)}$ in the middle part may be found in Müller (1988). The corresponding inflation factors (i.e. the α values) are determined by p or k.

4 Practical Implementation in R

The package *smoots* developed based on the algorithms described in the last section consists of five directly usable functions, two S3 methods and four datasets as examples. The first four functions are called *msmooth*, *tsmooth*, *gsmooth* and *knsmooth*, designed for estimating the trend in different ways. Data-driven estimation can be carried out by each of the first two of functions, where *msmooth* is a user-friendlier simplified version of *tsmooth*. Local polynomial estimation of $m^{(\nu)}$ and kernel smoothing of m with an arbitrary bandwidth fixed beforehand can be carried out by *gsmooth* and *knsmooth*, respectively. In the first case one should choose p such that $p - \nu$ is odd to avoid the boundary problem. Those functions allow a flexible application of this package by an experienced user. Data-driven estimation of the first or second derivative can be obtained by *dsmooth*.

The functions for selecting the bandwidth will be described in more detail. In the function *tsmooth* and in any other function, the first argument y stands for the input time series. For the second argument p, the order of polynomial, the user has the choice between *1* and *3*, whereas local linear regression with $p = 1$ is the default setting. Moreover, the argument *mu* stands for the smoothness of the weighting function, which can take the value 0, 1, 2 or 3, with $\mu = 1$ for the Epanechnikov kernel or the optimal weighting function being the preset option. Via the argument *Mcf*, the estimation method of the variance factor c_f can be chosen. Here, "*NP*" is the default approach corresponding to the data-driven estimation of c_f described in Section 3.1. In addition, three parametric options with the choices "*ARMA*", "*AR*" and "*MA*" are provided, if the error term is assumed to follow an ARMA, AR or MA model, respectively. Now, the maximal MA or AR order is set to be 5 and the best model will be selected by the BIC. However, we do not recommend those choices, because it is better to estimate the trend fully non-parametrically. Furthermore, the inflation factor α to calculate the bandwidth for estimating the k-th derivative involved in h_A is controlled by the argument *InfR*. The option "*Opt*" sets $\alpha_{1,O} = 5/7$ and $\alpha_{3,O} = 9/11$ for $p = 1$ and $p = 3$ respectively, whereas "*Nai*" incorporates the naive inflation rates $\alpha_{1,N} = 5/9$ and $\alpha_{3,N} = 9/13$. For this argument a third option "*Var*"is also included, which sets $\alpha_{1,S} = \alpha_{3,S} = 1/2$ to achieve a most stable selected bandwidth. In addition to the previously described arguments, also the starting bandwidth is adjustable via *bStart*. By default, it is set to *0.15*, which

is a compromise between the proposed starting values 0.1 and 0.2 for $p = 1$ and $p = 3$ respectively. Note that if $\hat{h}$ converges, it is usually not affected by *bStart*. Whether the bandwidth for estimating c_f should be enlarged or not is controlled by the choice *bvc* = "*Y*" or *bvc* = "*N*" with the former being the default setting. Furthermore, with the argument *bb* the boundary bandwidth can be specified. For the default *bb = 1* the k-nearest neighbor method is applied, which fixes the total bandwidth at $2\hat{h}$ at each observation point, whereas for *bb = 0* the total bandwidth is shortened at boundary points. Via the argument *cb* with its default *cb = 0.05* the percentage of observations omitted at each boundary c_b (see also Section 2) within the IPI-algorithm can be adjusted. Finally, the smoothing approach can be set by the last argument *method*. The two options are for local polynomial regression ("*lpr*") and kernel regression ("*kr*"). Local polynomial regression is the standard approach in the *smoots* package, whereas kernel regression is included as a comparison.

For the function *msmooth*, the arguments *Mcf*, *InfR*, *bvc*, *bb* and *cb* are replaced by the argument *alg*. The purpose of this argument is to implement algorithms with predefined settings of the omitted arguments. Here, the two algorithms **AlgA** and **AlgB** described in section 3.2 with *InfR*="*Opt*" and *InfR*="*Nai*" can be directly chosen by *alg* = "*A*" and *alg* = "*B*" respectively. For local linear regression we propose the use of **AlgA** with the optimal inflation factor. For local cubic regression with a moderate n, **AlgB** with the stronger inflation factor should be used. If n is big enough, e.g. bigger than 400, the combination of $p = 3$ and **AlgA** will also lead to suitable results. In case when slight over smoothing is wished/required, the inflation factor "*Nai*" or even "*Var*" should be employed. When applying the derivative estimation function *dsmooth*, the argument *d* with the default value *1* represents the order of derivative that will be estimated. In addition to the first derivative, also the second derivative (*d = 2*) can be estimated data-drivenly. Here, c_f is estimated via *tsmooth* with a pilot local polynomial regression of the order *pp* and a starting bandwidth *bStart.p*. For further details on the functions we refer the reader to the user's guideline for this package. Beside the above functions, two S3 methods are also built in the package: a *print* and a *plot* method for objects of class *smoots*, a newly introduced class of objects created by the *smoots* package. They allow for a quick and detailed overview of the estimation results. The *smoots* objects themselves are generally lists consisting of different components such as input data and estimation results.

5 Simple application of *smoots*

In this and the next two sections the wide applicability of the above mentioned functions will be illustrated by four real data examples. Those datasets are built in the package so that the reader can also use them. They are: *tempNH*, *gdpUS*, *dax* and *vix*, which contain observations of the mean monthly temperature changes of the Northern Hemisphere, the US GDP and daily financial data of the German stock index (DAX) and the CBOE Volatility Index (VIX), respectively. For further information see also the documentation on the data within the *smoots* package.

5.1 Direct application of the Semi-ARMA

To show the application of the additive Semi-ARMA model defined by (1) and (6), the time series of the mean monthly Northern Hemisphere temperature changes from 1880 to 2018 (NHTM) is chosen. The data are downloaded from the website of the National Aeronautics and Space Administration (NASA). For this purpose, the function *tsmooth* is applied. The used settings of the arguments for this function are equivalent to those in algorithm A with $p = 1$. Figure 1(a) shows the observations together with the estimated trend. Here, the selected bandwidth is 0.1221. We see, the estimated trend fits the data very well. In particular, the trend increases steadily after 1970 that might be a signal for possible global warming in the last decades. The residuals are displayed in Figure 1(b), which look quite stationary. This indicates that Model (1) is a suitable approach for this time series. Moreover, Figures 1(c) and 1(d) illustrate the data-driven estimates of the first and second derivatives of the trend respectively, which fit the features of the trend function very well and provide us further details about the global temperature changes.

Consequently, an ARMA(1, 1) model is fitted to the residual series using the *arima* function in R, which results in

$$\xi_t = 0.7489\xi_{t-1} - 0.3332\varepsilon_{t-1} + \varepsilon_t. \tag{10}$$

We see, the dependence of errors is dominated by a strong positive AR parameter with a moderate negative MA coefficient.

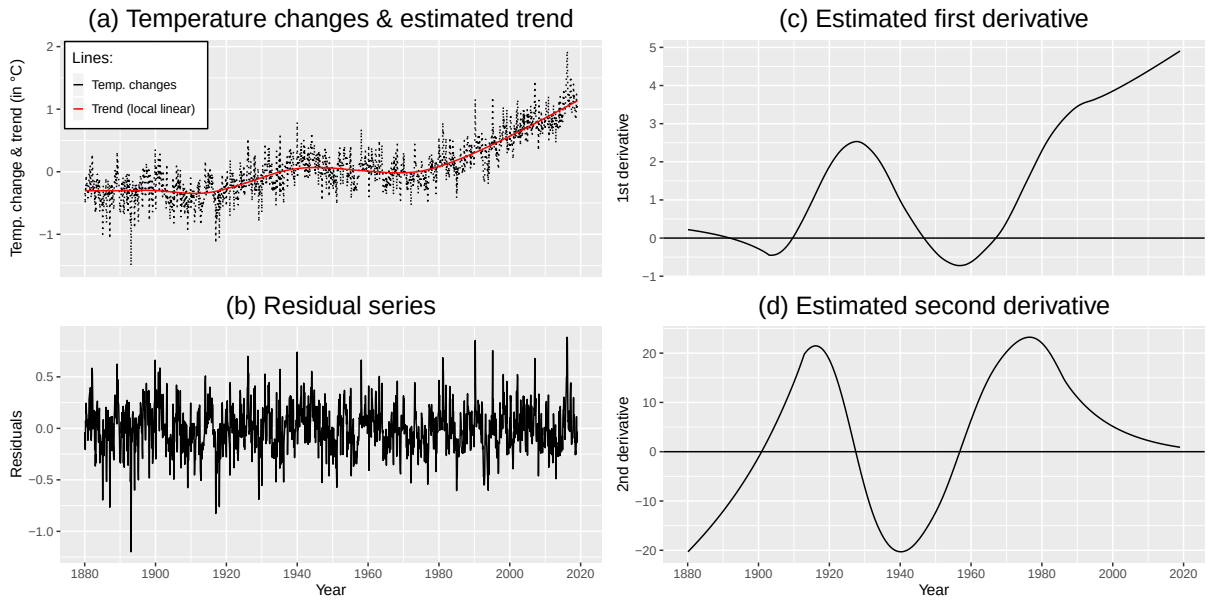

Figure 1: Estimated trend, residuals and the trend's derivatives for the NHTM series

5.2 A semiparametric log-local-linear growth model

A well-known approach in developing economics is the log-linear growth model. Assume that the log-transformation of a macroeconomic time series follows Models (1) and (6), we achieve a semiparametric local polynomial, in particular a local-linear extension of this theory. To show this application, the series of the quarterly US-GDP from the first quarter of 1947 to the second quarter of 2019, downloaded from the Federal Reserve Bank of St. Louis, is chosen. Data-driven local linear regression is applied to estimate the trend from the log-data using $AlgA$ with the selected bandwidth 0.1325. A kernel regression estimate using the same bandwidth is also carried out for comparison. The results together with the log-data are displayed in Figure 2(a). We see that the two trend estimates in the middle part coincide with each other. They differ from each other only at the boundary points and the kernel estimate is affected by a clear boundary problem. Thus, the local linear method should be used. Residuals of this estimate are shown in Figure 2(b). Again, the estimated first and second derivatives are given in Figures 2(c) and 2(d), respectively, which help us to discover more detailed features of the economic development in the US. Furthermore, the following ARMA(1, 1) model is obtained from the residuals:

$$\xi_t = 0.9079\xi_{t-1} + 0.2771\varepsilon_{t-1} + \varepsilon_t. \tag{11}$$

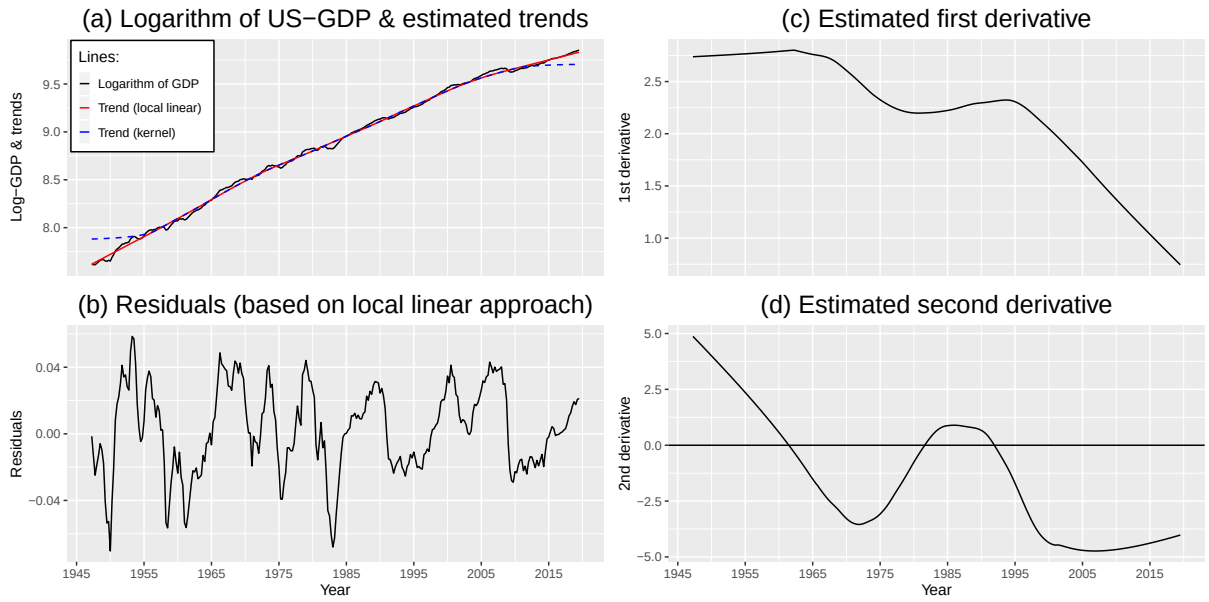

Figure 2: Estimation results for the log-quarterly US-GDP series

6 The Semi-Log-GARCH model

The most well-known volatility models are the ARCH (autoregressive conditional heteroskedasticity) and the GARCH (generalized ARCH) models introduced by Engle (1982) and Bollerslev (1986). A special extension of the GARCH model is the Log-GARCH (logarithmic GARCH) introduced by Pantula (1986), Geweke (1986) and Milhøj (1988). This model received less attention until several years ago, when Francq et al. (2013) rediscovered its usefulness. Note that most of the GARCH extensions are defined for stationary return series. In practice it is however found that the unconditional variance of financial returns may change slowly over time and are hence non-stationary. To overcome this problem a semiparametric GARCH (Semi-GARCH) approach is proposed by Feng (2004) by defining the return series as a product of a (deterministic) smooth scale function and a GARCH model. Another well-known closely related approach is the Spline-GARCH introduced by Engle and Rangel (2008). In this paper we will introduce a Semi-Log-GARCH (semiparametric Log-GARCH) by assuming that the stationary part follows a Log-GARCH model. Consequently, the scale function in the Semi-Log-GARCH model

will be estimated by means of the log-transformation of the squared returns as proposed by Engle and Rangel (2008).

Denote the returns by r_t, $t = 1, ..., n$. The Semi-Log-GARCH model is defined by

$$r_t = \sqrt{v(x_t)}\zeta_t \text{ with } \zeta_t = \sqrt{h_t}\eta_t \text{ and} \tag{12}$$

$$\ln(h_t) = \omega + \sum_{i=1}^{l} \alpha_i \ln(\zeta_{t-i}^2) + \sum_{j=1}^{s} \beta_j \ln(h_{t-j}), \tag{13}$$

where $v(x_t) > 0$ is a smooth local variance component, $s(x_t) = \sqrt{v(x_t)}$ is called the scale function and η_t are i.i.d. random variables with zero mean and unit variance. It is assumed that ζ_t also has unit variance and $\zeta_t \neq 0$ almost surely, so that the model is well-defined. Let $y_t = \ln[(r_t)^2]$, $\xi_t = \ln(\zeta_t^2) - \mu_{lz}$ and $m(x_t) = \ln[v(x_t)] + \mu_{lz}$, where $\mu_{lz} = E[\ln(\zeta_t^2)]$. We see that the log-transformation of r_t^2 of the Semi-Log-GARCH model has the form $y_t = m(x_t) + \xi_t$, which is a special case of Model (1). Furthermore, define $\varepsilon_t = \ln(\eta_t^2) - \mu_{le}$ with $\mu_{le} = E[\ln(\eta_t^2)]$, which are i.i.d. zero mean innovations in the stationary process ξ_t and we have $\xi_t = \ln(h_t) + \varepsilon_t - (\mu_{lz} - \mu_{le})$. According to Francq and Sucarrat (2013), ξ_t has the following ARMA representation:

$$\xi_t = \sum_{i=1}^{l^*} \varphi_i \xi_{t-i} + \sum_{j=1}^{s} \psi_j \varepsilon_{t-j} + \varepsilon_t \tag{14}$$

with $l^* = \max(l, s)$. Thus, the Semi-Log-GARCH model is equivalent to a Semi-ARMA of the log-transformation of r_t^2 with the restriction that the AR order should not be less than the MA order. Hence, this model can be simply estimated using the *smoots* package and the *arima* function. The relationship between the coefficients in (13) und (14) are $\alpha_i = \varphi_i + \psi_i$, $\beta_j = -\psi_j$, where the non-existing coefficients are assumed to be zero, and $\omega = \big(1 - \sum_{i=1}^{l^*} \varphi_i\big)\mu_{lz} - \big(1 + \sum_{j=1}^{s} \psi_j\big)\mu_{le}$. After obtaining $\hat{m}(x_t)$ and $\hat{\xi}_t$, $\hat{v}(x_t)$ and $\hat{h}_t$ can be obtained by $\hat{v}(x_t) = \hat{C}_v \exp[\hat{m}(x_t)]$ and $\hat{h}_t = \hat{C}_h \exp(\hat{\xi}_t)$, where C_v and C_h are determined by the facts that both of $\text{var}\,(\zeta_t)$ and $\text{var}\,(\eta_t)$ are 1 and can be estimated from the rescaled returns and through standardizing the innovations. Although $\hat{\omega}$ can be obtained similarly, it is however unnecessary. We see, $\hat{h}_t$ can be calculated without using its explicit form in the definition. The parametric part of the Semi-Log-GARCH can also be estimated directly using the R package *lgarch*. This will however not be considered in the current paper.

The DAX series from 1990 to July 2019 downloaded from Yahoo Finance is chosen to show the application of the Semi-Log-GARCH model. Note that an observed return can be sometimes exactly zero. To overcome this problem, the log-transformation is calculated for the squared centralized returns, which are a.s. non-zero. This would even be a necessary treatment, if the returns had a very small, but non-zero mean. The centralized log-returns and the log-transformed data with two estimated trends are displayed in Figures 3(a) and 3(b). Those trends are obtained by the local linear regression (blue) and the local cubic regression (red). In both cases *AlgA* was employed. The selected bandwidths are 0.0869 and 0.1013, respectively. Results in Figure 3(b) indicate that the unconditional variance of the DAX-returns changes slowly over time. The local cubic trend is chosen for further analysis, because here the results of the local linear approach are over-smoothed. The following ARMA(1, 1) model is obtained from the residuals of the log-data

$$\xi_t = 0.9692\xi_{t-1} - 0.9221\varepsilon_{t-1} + \varepsilon_t, \tag{15}$$

whereas the re-transformed Log-GARCH(1, 1) formula is given by

$$\ln(h_t) = 0.0685 + 0.0471\ \ln(\zeta_{t-1}^2) + 0.9221\ \ln(h_{t-1}) \tag{16}$$

following the idea of Sucarrat et al. (2016). The estimated volatility series ($\sqrt{\hat{h}_t}$) and total volatility ($\hat{s}(x_t)\sqrt{\hat{h}_t}$) are displayed in Figures 3(c) and 3(d).

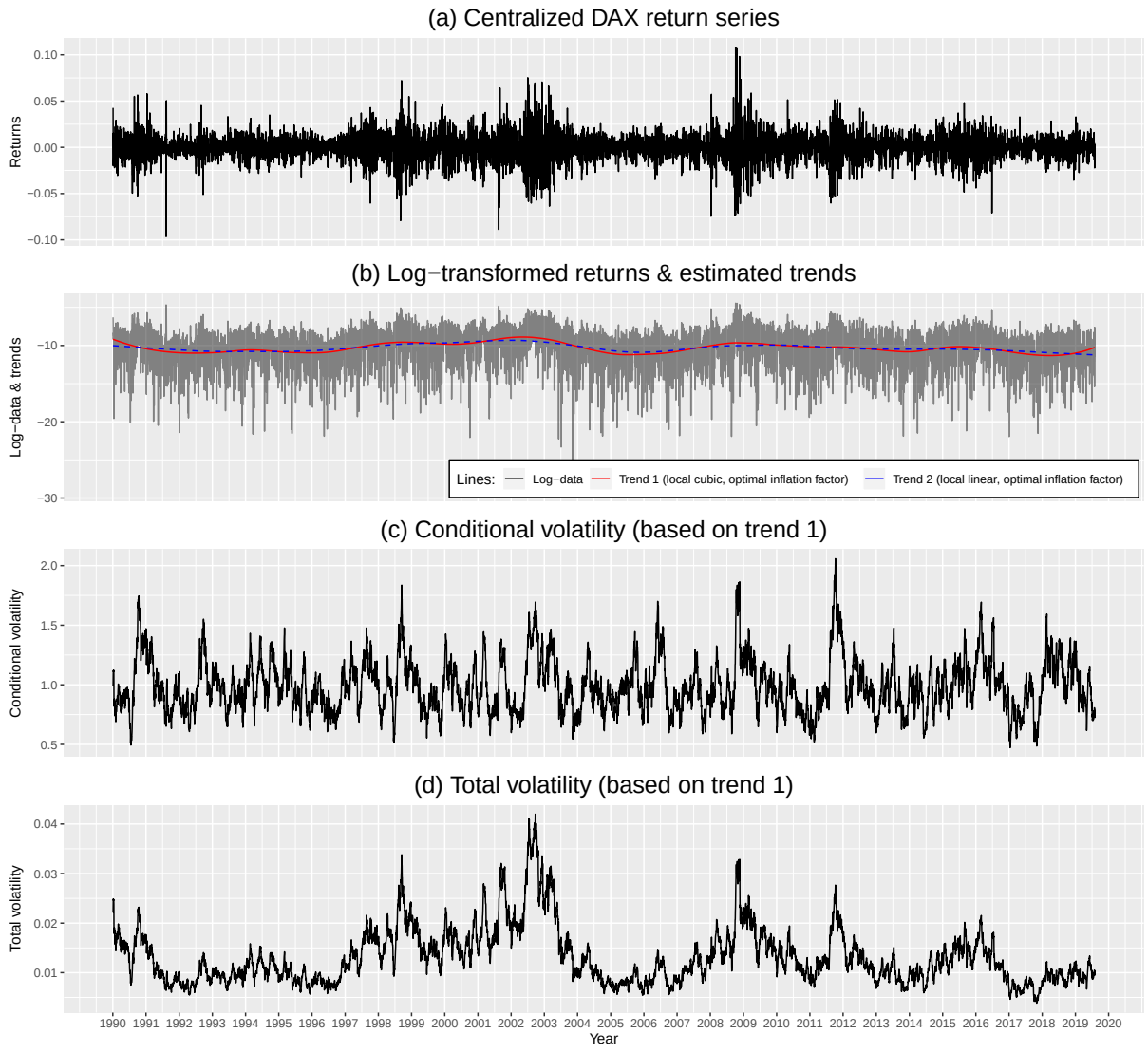

Figure 3: Results of the Semi-Log-GARCH model for DAX returns.

7 The Semi-Log-ACD model

A more general framework for modeling non-negative financial time series is the ACD (autoregressive conditional duration, Engle and Russell, 1998) model, which corresponds to a squared GARCH model and can be applied to both of high-frequency or daily financial data. Logarithmic extensions of this approach were introduced by Bauwens and Giot (2000), where the Type I definition (called a Log-ACD) corresponds to a squared form of the Log-GARCH. Semiparametric generation of the Log-ACD (Semi-Log-ACD) was

defined and applied to different kinds of non-negative financial data by Forstinger (2018). In this paper, the application of the Semi-Log-ACD model will be illustrated by the CBOE Volatility Index (VIX) from 1990 to July 2019, denoted by V_t, $t = 1, ..., n$. The data was again downloaded from Yahoo Finance. The Semi-Log-ACD model for V_t is defined by

$$V_t = g(x_t)\lambda_t e_t \tag{17}$$

and that $u_t = \lambda_t e_t$ follows a Log-ACD, where $g \geq 0$ is a smooth mean function in V_t, $\lambda_t \geq 0$ is the conditional mean and e_t is an i.i.id. series of non-negative random variables. It is assumed that $E(\lambda_t) = E(e_t) = 1$. Further investigation on this model can be carried out similarly to that on the Semi-Log-GARCH model by replacing r_t^2, h_t and η_t^2 there with V_t, λ_t and e_t, respectively. And the Semi-Log-ACD model can be similarly estimated. Discussion on those details is omitted. For further information we refer the reader to Forstinger (2018) and references therein.

The original series of V_t is shown in Figure 4(a). The trend was estimated from the log-transformation of V_t using both *AlgA* and *AlgB* with the selected bandwidths 0.0771 and 0.1598, respectively. The data (black), the local linear trend (red) and the local cubic trend (blue) are displayed in Figure 4(b). The results using both algorithms are quite similar, except for that the local cubic estimates are smoother. From the residuals of the local linear approach the following ARMA(1, 1) model is obtained:

$$\xi_t = 0.9626\xi_{t-1} - 0.0707\varepsilon_{t-1} + \varepsilon_t. \tag{18}$$

Subsequently, the estimated log-form of the conditional mean function is given by

$$\ln(\lambda_t) = 0.0010 + 0.8919\ \ln(u_{t-1}) + 0.0707\ \ln(\lambda_{t-1}). \tag{19}$$

The estimated conditional means in the log-data and the total means in the original data by the local linear approach are shown in Figures 4(c) and 4(d). We see the results fit the data very well. This model can be applied for forecasting the VIX in the near future.

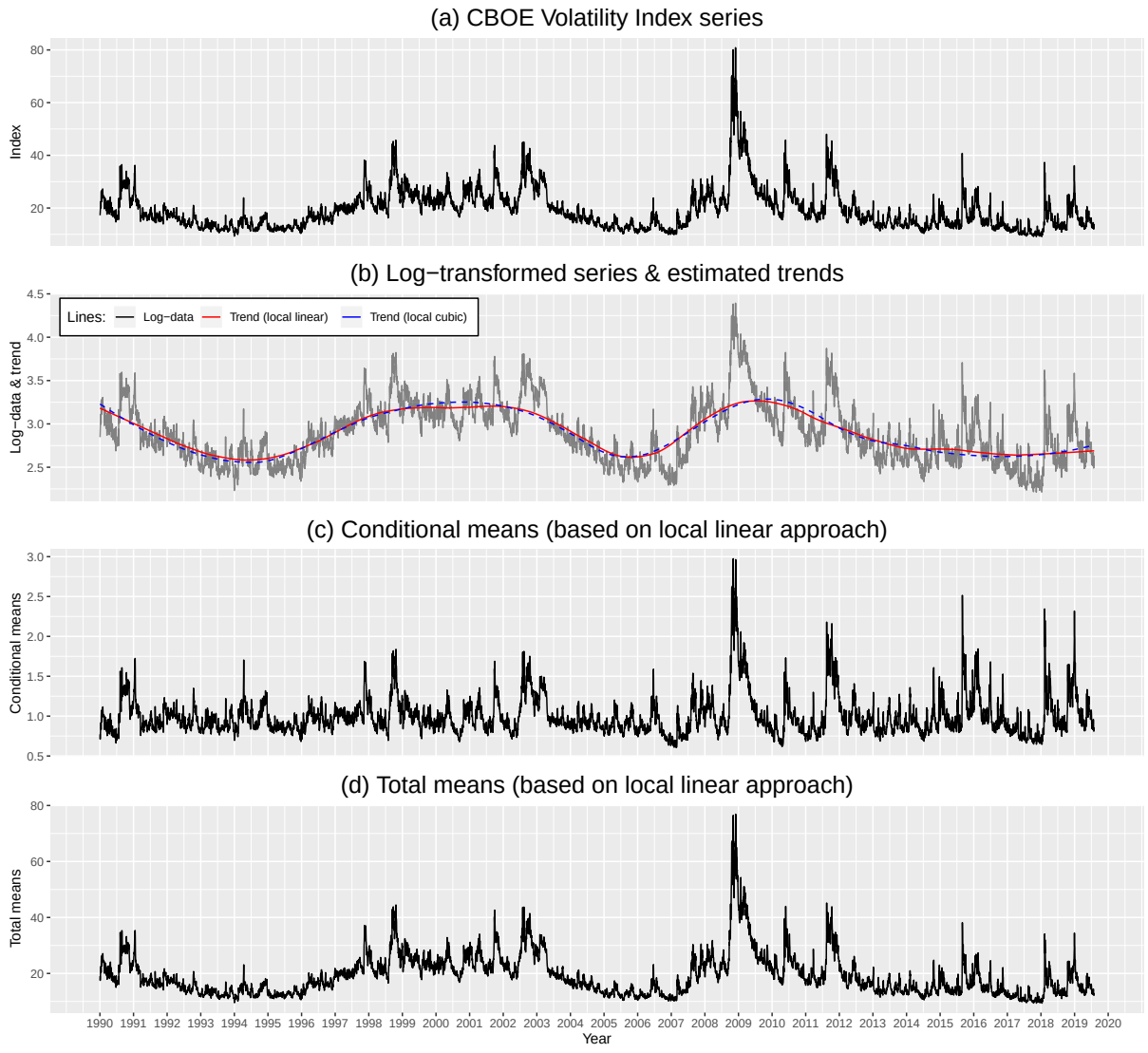

Figure 4: smoots applied to VIX, 1990 - July 2019

8 Concluding remarks

In this paper the methodological background for developing the R package *smoots* (version 1.0.1) is summarized. The usage of the main functions in this package is explained in detail. To show the wide applicability of this approach two new semiparametric models for analyzing financial time series are also introduced. The four data examples showed that the proposed approach can be applied to different kinds non-stationary time series and the developed R package works very well for data-driven implementation of those

semiparametric time series models. In particular, non-negative time series following a semiparametric multiplicative model can be easily estimated via the log-transformation. But, it is found that the errors in some examples could exhibit clear long memory. However, the current package version only works under short memory assumption. It is hence worthy to study the possible extension of the current approach to semiparametric time series models with long memory errors. Further extensions of the proposals in this paper, the development of suitable forecasting procedures and tools for testing stationarity and linearity of time series should also be studied in the future and considered in future versions of the *smoots* package or in supplementing R packages.

References

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.

Beran, J., Feng, Y. and Heiler, S. (2009). Modifying the double smoothing bandwidth selector in nonparametric regression. In: *Statistical Methodology* 6.5, pp. 447–465.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. In: *Journal of econometrics* 31.3, pp. 307–327.

Bühlmann, P. (1996). Locally adaptive lag-window spectral estimation. In: *Journal of Time Series Analysis* 17.3, pp. 247–270.

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.

Engle, R. F. and Rangel, J. G. (2008). The spline-GARCH model for low-frequency volatility and its global macroeconomic causes. In: *The Review of Financial Studies* 21.3, pp. 1187–1222.

Engle, R. F. and Russell, J. R. (1998). Autoregressive conditional duration: a new model for irregularly spaced transaction data. In: *Econometrica*, pp. 1127–1162.

Feng, Y. (2004). Simultaneously modeling conditional heteroskedasticity and scale change. In: *Econometric Theory* 20.3, pp. 563–596.

– (2007). On the asymptotic variance in nonparametric regression with fractional time-series errors. In: *Nonparametric Statistics* 19.2, pp. 63–76.

Feng, Y., Gries, T. and Fritz, M. (2019). Data-driven local polynomial for the trend and its derivatives in time series. Discussion paper. Paderborn University.

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.

Feng, Y. and Schulz, D. (2019). *smoots: Nonparametric Estimation of the Trend and Its Derivatives in TS*. R package version 1.0.1. URL: `https://CRAN.R-project.org/package=smoots`.

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

Francisco-Fernández, M., Opsomer, J. and Vilar-Fernández, J. M. (2004). Plug-in bandwidth selector for local polynomial regression estimator with correlated errors. In: *Nonparametric Statistics* 16.1-2, pp. 127–151.

Francq, C. and Sucarrat, G. (2013). An exponential chi-squared QMLE for Log-GARCH models via the ARMA representation. In:

Francq, C., Wintenberger, O. and Zakoïan, J.-M. (2013). GARCH models without positivity constraints: Exponential or Log GARCH? In: *Journal of Econometrics* 177.1, pp. 34–46.

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.

Geweke, J. (1986). Modeling the persistence of conditional variances: a comment. In: *Econometric Reviews* 5, pp. 57–61.

Milhøj, A. (1988). *A Multiplicative Parametrization of ARCH Models*. Universitetets Statistiske Institut.

Müller, H.-G. (1988). *Nonparametric Regression Analysis of Longitudinal Data*. Berlin: Springer.

Pantula, S. G. (1986). Modeling the persistence of conditional variances: a comment. In: *Econometric Reviews* 5, pp. 79–97.

Ruppert, D., Sheather, S. J. and Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. In: *Journal of the American Statistical Association* 90.432, pp. 1257–1270.

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.

Appendix

Installation of the *smoots* package from CRAN:

```
install.packages("smoots")
library(smoots)
```

Code for Example 1:

```
tempChange <- smoots::tempNH$Change
est_temp <- smoots::tsmooth(tempChange, p = 1, mu = 2, Mcf = "NP",
  InfR = "Opt", bStart = 0.1, bvc = "Y", method = "lpr")
d1_temp <- smoots::dsmooth(tempChange, d = 1, mu = 2, pp = 3, bStart.p = 0.2,
  bStart = 0.15)
d2_temp <- smoots::dsmooth(tempChange, d = 2, mu = 3, pp = 1, bStart.p = 0.1,
  bStart = 0.2)
arma1 <- stats::arima(est_temp$res, order = c(1, 0, 1), include.mean = FALSE)
```

Code for Example 2:

```
l_gdp <- log(smoots::gdpUS$GDP)
gdp_t1 <- smoots::msmooth(l_gdp, p = 1, mu = 1, bStart = 0.1, alg = "A",
  method = "lpr")
gdp_t2 <- smoots::msmooth(l_gdp, p = 1, mu = 1, bStart = 0.1, alg = "A",
  method = "kr")
gdp_d1 <- smoots::dsmooth(l_gdp, d = 1, mu = 1, pp = 1, bStart.p = 0.1,
  bStart = 0.15)
gdp_d2 <- smoots::dsmooth(l_gdp, d = 2, mu = 1, pp = 1, bStart.p = 0.1,
  bStart = 0.2)
arma2 <- stats::arima(gdp_t1$res, order = c(1, 0, 1), include.mean = FALSE)
```

Code for Example 3:

```
dax_close <- smoots::dax$Close; dax <- diff(log(dax_close))
rt <- dax - mean(dax); yt <- log(rt ^ 2)
estim3 <- smoots::msmooth(yt, p = 3, alg = "A")
estim3.2 <- smoots::msmooth(yt)
```

```
m_xt <- estim3$ye
xi <- estim3$res
s_star <- exp(m_xt / 2)
zeta_star <- rt / s_star
mulz <- log(1 / var(zeta_star))
s <- s_star / exp(0.5 * mulz)
arma3 <- arima(xi, order = c(1, 0, 1), include.mean = FALSE)
beta <- -arma3$coef[[2]]; alpha <- arma3$coef[[1]] - beta
mule <- -log(mean(exp(arma3$residuals)))
omega <- mulz * (1 - arma3$coef[[1]]) - (1 - beta) * mule
ht <- exp(xi - arma3$residuals + (mulz - mule))
vol <- s * sqrt(ht)
```

Code for Example 4:

```
vix <- smoots::vix$Close; lnV <- log(vix)
estim4.1 <- smoots::msmooth(lnV)
estim4.2 <- smoots::msmooth(lnV, p = 3, alg = "B")
m_xt <- estim4.1$ye
xi <- estim4.1$res
g_star <- exp(m_xt)
zeta2_star <- vix / g_star
mulz <- log(1 / mean(zeta2_star))
g <- g_star / exp(mulz)
arma4 <- arima(xi, order = c(1, 0, 1), include.mean = FALSE)
beta <- -arma4$coef[[2]]; alpha <- arma4$coef[[1]] - beta
mule <- -log(mean(exp(arma4$residuals)))
omega <- mulz * (1 - arma4$coef[[1]]) - (1 - beta) * mule
lambda <- exp(xi - arma4$residuals + mulz - mule)
means <- g * lambda
```