The smoots Package in R for Semiparametric Modeling of Trend Stationary Time Series

by Yuanhua Feng, Thomas Gries, Sebastian Letmathe and Dominik Schulz

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 a semiparametric multiplicative component model for non-negative time series via the log-transformation. Furthermore, we introduced a semiparametric Log-GARCH and a semiparametric Log-ACD model, which can be easily estimated by the **smoots** package. Of course, this package applies to suitable time series from any other 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.

1 Introduction

This paper provides an introduction to a new package in R called **smoots** (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. (2020). 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 related proposal may be found in Francisco-Fernández et al. (2004). 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, which is usually a misspecification. (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 **smoots** package complements the Comprehensive R Archive Network (CRAN), as already existing base R functions or CRAN packages for local polynomial smoothing either do not implement an automated bandwidth selection or, if such bandwidth algorithms do exist, they are only suitable for data with i.i.d. (independently and identically distributed) errors, which is an assumption that is often violated for time series. In more detail, notwithstanding that the **stats** package offers the functions `lowess()` and `loess()` for computing the local polynomial estimates of the regression function given one or multiple predictor variables, a smoothing bandwidth has to be selected arbitrarily in both cases. Similar functionality is provided by the **locpol** (Ojeda Cabrera, 2018), **KernSmooth** (Wand, 2021) and **lokern** (Herrmann and Maechler, 2021) packages, however, derivative estimation approaches and various bandwidth selection algorithms for the regression function, such as the leave-one-out estimator (Allen, 1974) and the plug-in algorithm by Ruppert et al. (1995), are built into them. Moreover, within **lokern**, the bandwidth for estimating the first or second derivative of the regression function can be selected automatically. Nevertheless, these traditional data-driven bandwidth selection approaches are only suitable in case of data with i.i.d. errors (Hart, 1991; Altman, 1990; Opsomer, 1997) and can therefore often not be considered for time series. Explicit CRAN packages for detrending a time series nonparametrically are **rmaf** (Qiu, 2015) and **mFilter** (Balcilar, 2019). While with **rmaf**, a fully data-driven refined moving average and a cubic smoothing spline, which requires a manually selected smoothness parameter, can be applied, the trend as well as the cyclical component of a time series can be extracted with the filters implemented in the **mFilter** package like the Baxter-King (Baxter and King, 1999) or Hodrick-Prescott (Hodrick and Prescott, 1997) filters among others. However, all filtering functions in the **mFilter** package are not fully data-driven. Instead, they make use of default values

recommended within scientific literature. Thus, to our knowledge, there are not any packages on CRAN that implement a data-driven local polynomial regression for the trend of a time series or its derivatives.

The methodological and computational background for the **smoots** package is summarized briefly hereinafter. For further details we refer the reader to Feng et al. (2020) and references therein. Our focus is to illustrate the wide application spectrum of the **smoots** package for semiparametric 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 of the **stats** package in R. This idea is first shown with monthly Northern Hemisphere temperature changes data obtained from the website of the National Aeronautics and Space Administration (NASA). Then the proposal is applied to the log-transformation of the quarterly US-GDP data, retrieved from the website of the Federal Reserve Bank of St. Louis, 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; Milhøj, 1987). The latter is a special extension of the seminal ARCH (autoregressive conditional heteroskedasticity, Engle, 1982) and GARCH (generalized ARCH, Bollerslev, 1986) models and is just a slightly restricted ARMA model for the log-transformation of the squared returns (see e.g. Sucarrat, 2019). The usefulness of the Log-GARCH has recently been rediscovered by Francq et al. (2013). The application of this proposal is illustrated by the DAX returns collected from Yahoo Finance. 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) obtained from Yahoo Finance. Datasets for all of those examples are built in the proposed package. The **smoots** package can be applied to suitable time series from other research areas. In addition, the **smoots** package provides a useful tool for teaching time series analysis, which 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 the following two sections. The general usage of the R functions is then described in another section. Three further sections illustrate the applications of those functions in simple cases and with respect to the Semi-Log-GARCH and the Semi-Log-ACD.

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 for example 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. Under the regularity conditions given in Bühlmann (1996), a data-driven IPI-algorithm for estimating the variance factor in the asymptotically optimal bandwidth can be developed. As indicated by Feng et al. (2020), the required conditions are fulfilled by an ARMA process with finite eighth moment. However, models with long-memory errors are excluded by those assumptions.

The local polynomial estimator of $m^{(\nu)}(x)$, the ν-th derivative of m, 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:

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

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 mean integrated squared error):

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

where $I\left[m^{(k)}\right] = \int_{c_b}^{d_b} \left[m^{(k)}(x)\right]^2 dx$, $\beta = \int_{-1}^{1} u^k K(u)\, du$, $R(K) = \int_{-1}^{1} K^2(u)\, du$, $c_f = f(0)$ 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

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

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:

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

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

3 The proposed IPI-algorithms

Since both c_f and $I\left[m^{(k)}\right]$ in (5) are unknown, we need to obtain and insert appropriate estimates of these two quantities into this formula to achieve a selected bandwidth $\hat{h}_A$. However, the estimation of c_f and $I\left[m^{(k)}\right]$ requires the use of two additional bandwidths. Hence, iterative bandwidth selection procedures should be employed. The estimation of $I\left[m^{(k)}\right]$ is not influenced by the correlation structure, which can be simply obtained by means of established IPI-algorithms for models with independent errors. Consequently, solely a comprehensive review of the estimation approach for c_f will be given.

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. (2020), 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 $\int \left(f(\lambda)^2\right) d\lambda$ 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 $M_j' = \left[M_{j-1}/n^{2/21}\right]$, 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 $M' = \left[\hat{M}_G/n^{2/21}\right]$. 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.

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. (2020) showed that the asymptotically optimal bandwidth for estimating c_f should be $\text{CF} \cdot h_A$, not h_A itself, where

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

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 $r_{t,\text{V}}$ obtained with the bandwidth $\text{CF} \cdot \hat{h}$. The proposed main IPI-algorithm for selecting the bandwidth proceeds as follows.

i) Start with an initial bandwidth h_0 given beforehand.

ii) Obtain $r_{t,\text{V}}$ using h_{j-1} or $\text{CF} \cdot h_{j-1}$ and estimate c_f from $r_{t,\text{V}}$ 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 $I\left[m^{(k)}\right]$ with $h_{d,j} = h_{j-1}^{\alpha}$ and a local polynomial of order $p_d = p + 2$. We obtain

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

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

In the developed package the initial bandwidth $h_0 = 0.15$ for both $p = 1$ and $p = 3$ is used. Also, for both $p = 1$ and $p = 3$, the default value $c_b = 1 - d_b = 0.05$ is proposed 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. 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 $I\left[\hat{m}^{(k)}\right]$ 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. (2020).

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 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. Moreover, for simplicity, shared arguments between functions will only be discussed once. With

```
tsmooth(y, p, mu, Mcf, InfR, bStart, bvc, bb, cb, method),
```

the trend in equidistant and trend-stationary time series with short-memory can be estimated via an automated local polynomial. Its arguments are as follows.

- `y` is the input time series.
- `p` reflects the order of polynomial and currently only `p = 1` (local linear) and `p = 3` (local cubic) are selectable.
- `mu` corresponds to the smoothness parameter μ of the second order kernel function (3) used for weighting. Only `0`, `1`, `2` and `3` are valid options.
- `Mcf` defines the method for estimating c_f. For `Mcf = "NP"`, the default, c_f is estimated nonparametrically as described in the section *Data-driven estimation of c_f*. If `"AR"`, `"MA"` or `"ARMA"` are selected, ξ_t in model (1) is assumed to follow an AR, MA or ARMA process, respectively, during the bandwidth selection and thus, c_f is estimated parametrically.
- `InfR` sets the inflation rate α considered for the bandwidth (see also Step iii) in the section *The IPI-algorithm for estimating m*). The options are `InfR = "Opt"`, which corresponds to $\alpha = 5/7$ for a local linear and $\alpha = 9/11$ for a local cubic regression, `InfR = "Nai"` with $\alpha = 5/9$ and $\alpha = 9/13$ for local linear and local cubic regressions, respectively, and `InfR = "Var"`, which always sets $\alpha = 1/2$.
- `bStart` is the (relative) starting bandwidth for the bandwidth selection algorithm. The default is `bStart = 0.15`. Note that $\hat{h}$ is usually not affected by the initial value.
- With the argument `bvc` the estimation of c_f can be adjusted even further. By setting `bvc = "Y"`, the bandwidth for estimating c_f will be enlarged by the factor CF as in (8). CF is omitted for `bvc = "N"`.
- `bb` describes the boundary method. If `bb = 0` is selected, the number of observations considered for smoothing is decreasing towards the boundaries, while it is constant throughout for `bb = 1`.
- `cb` is the proportion of observations at each boundary that is omitted in the bandwidth selection process.
- Via `method` the final smoothing method, after the bandwidth has been selected, can be defined. The default `method = "lpr"` corresponds to local polynomial estimates, whereas with `method = "kr"` a kernel regression is conducted.

A simplified version of `tsmooth()` for estimating a time series trend with a data-driven local polynomial is

```
msmooth(y, p, mu, bStart, alg, method),
```

which shares most of its arguments with `tsmooth()`. The only unknown argument is `alg`.

- `alg` defines specific argument settings of `tsmooth()` as named subalgorithms. The two algorithms **AlgA** and **AlgB** described in the section *The IPI-algorithm for estimating m* can be directly chosen by `alg = "A"` and `alg = "B"`, respectively.

In accordance with Feng et al. (2020), we propose the use of **AlgA** with the optimal inflation factor for local linear regression. 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.

To estimate the first or second derivative of a time series trend with a data-driven local polynomial, the function

```
dsmooth(y, d, mu, pp, bStart.p, bStart)
```

should be employed. With this function, the first and second derivatives are always estimated using a local quadratic and local cubic regression, respectively. To obtain $\hat{c}_f$, a data-driven pilot estimate of the trend function via `tsmooth()` is computed.

- `d` specifies the order of derivative of the trend that will be estimated. Currently, `d = 1` and `d = 2` are valid options.
- `pp` corresponds to the order of polynomial considered for the pilot estimate of the trend function. `pp = 1` and `pp = 3` are available.
- `bStart.p` is the starting bandwidth for the data-driven pilot estimate of the trend.

Note that here, `bStart` is the initial bandwidth considered in the bandwidth selection for the derivative series. 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. Since the package is available on CRAN, the commands `install.packages("smoots")` and `library(smoots)` can be used to install it and attach it within the current R environment.

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 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 $p = 1$.

```
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)
```

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.

```
arma1 <- stats::arima(est_temp$res, order = c(1, 0, 1), include.mean = FALSE)
```

Consequently, an ARMA(1, 1) model is fitted to the residual series $\tilde{\xi}_t = y_t - \hat{m}(x_t)$ using the `arima()` function in R, which results in

$$\tilde{\xi}_t = 0.7489\tilde{\xi}_{t-1} - 0.3332\varepsilon_{t-1} + \varepsilon_t. \tag{10}$$

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

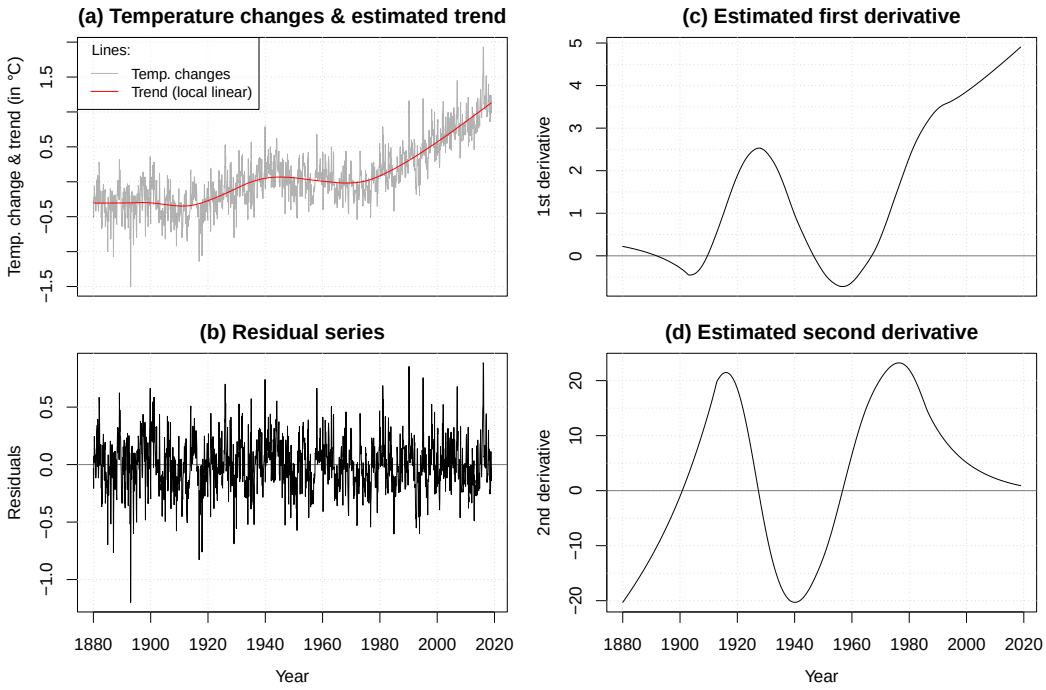

Figure 1: The NHTM series and the estimated trend are displayed in (a). The residuals, as well as the estimated first and second derivatives are shown in (b), (c) and (d), respectively.

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.

```
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)
```

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.

```
arma2 <- stats::arima(gdp_t1$res, order = c(1, 0, 1), include.mean = FALSE)
```

Furthermore, the following ARMA(1, 1) model is obtained from the residuals:

$$\tilde{\xi}_t = 0.9079\tilde{\xi}_{t-1} + 0.2771\varepsilon_{t-1} + \varepsilon_t. \tag{11}$$

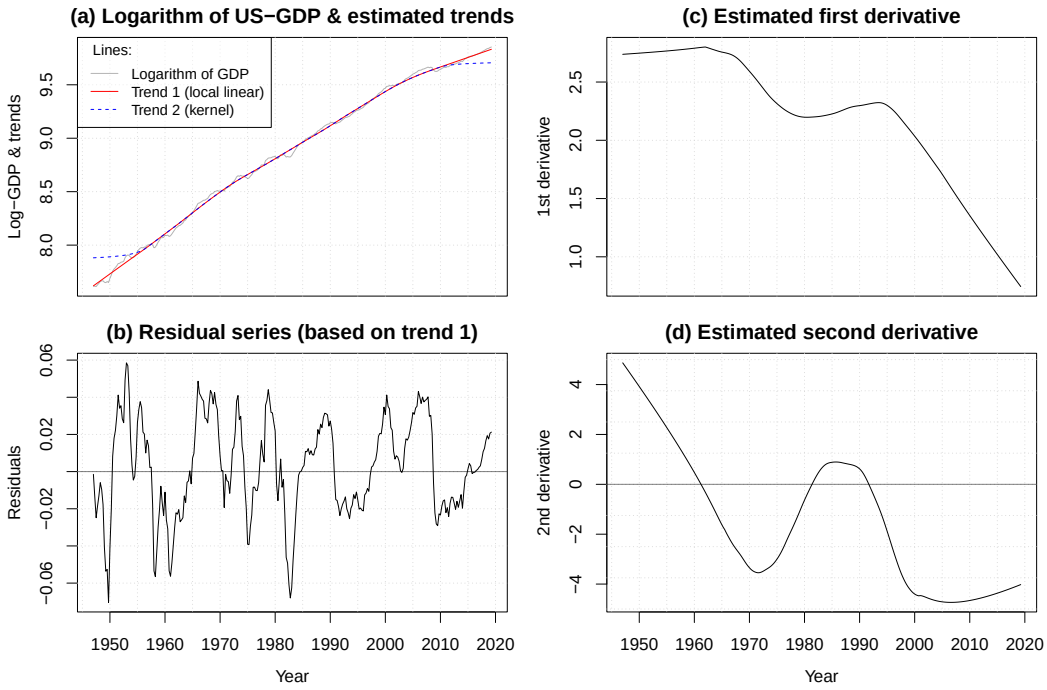

Figure 2: The log-transformed GDP series and the estimated trends are displayed in (a). The residuals based on the estimated local linear trend, as well as the estimated first and second derivatives are shown in (b), (c) and (d), respectively.

6 The Semi-Log-GARCH model

Most of the GARCH extensions, including the Log-GARCH (Pantula, 1986; Geweke, 1986; Milhøj, 1987), are defined for stationary return series. Recently, Francq et al. (2013) rediscovered the usefulness of the Log-GARCH. 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), defined as a Log-GARCH with a smooth scale function. We propose to estimate the scale function in the Semi-Log-GARCH model based on the log-transformation of the squared returns as proposed by Engle and Rangel (2008).

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

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

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

where $v(x_t) > 0$ is a smooth local variance component, $h_t > 0$ are conditional variances 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 $y_t = \ln\left(r_t^2\right)$, $\xi_t = \ln\left(\zeta_t^2\right) - \mu_{lz}$ and $m(x_t) = \ln[v(x_t)] + \mu_{lz}$, where $\mu_{lz} = E\left[\ln\left(\zeta_t^2\right)\right]$. 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 $\varepsilon_t = \ln\left(\eta_t^2\right) - \mu_{le}$ with $\mu_{le} = E\left[\ln\left(\eta_t^2\right)\right]$, which are i.i.d. zero mean innovations in the stationary process ξ_t. According to Francq and Sucarrat (2018), 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 of the **stats**

package. To obtain the total volatilities $\sigma_t = \sqrt{v(x_t)\, h_t}$, we suggest to utilize the three-step estimation procedure as proposed in Section 3.3 of Sucarrat (2019), however, we recommend to explicitly conduct the auxiliary regression in Step 1 in Sucarrat (2019) by using a data-driven local polynomial.

i) Obtain estimates of the nonparametric trend function $\hat{m}(x_t)$ in y_t via the **smoots** package.

ii) Fit an ARMA(l^*, s) model (14) to the residuals $\tilde{\xi}_t = y_t - \hat{m}(x_t)$, for example with the `arima()` function of the **stats** package, and compute the ARMA residuals $\hat{\varepsilon}_t$.

iii) Consider $\hat{\mu}_{le} = -\ln\left[\frac{1}{n}\sum_{i=1}^{n}\exp(\hat{\varepsilon}_i)\right]$ as an estimator of the log-moment $E\left[\ln\left(\eta_t^2\right)\right]$. Then the total volatilities are estimated as

$$\hat{\sigma}_t = \exp\left\{\left[\tilde{\xi}_t - \hat{\varepsilon}_t + \hat{m}(x_t) - \hat{\mu}_{le}\right]/2\right\}. \tag{15}$$

Estimates of the conditional volatilities are also calculable similarly by following the same three-step procedure while replacing (15) with $\sqrt{\hat{h}_t} = \exp\left[\left(\tilde{\xi}_t - \hat{\varepsilon}_t + \hat{\mu}_{lz} - \hat{\mu}_{le}\right)/2\right]$, where we suggest $\hat{\mu}_{lz} = -\ln\left[\frac{1}{n}\sum_{i=1}^{n}\exp\left(\tilde{\xi}_i\right)\right]$. An important characteristic of the illustrated estimation procedure is that explicit estimates of the Log-GARCH parameters ω, α_i, for $1 \leq i \leq p$, and β_j, for $1 \leq j \leq q$, are not required for computing $\hat{\sigma}_t$ or $\sqrt{\hat{h}_t}$ (Sucarrat, 2019). If, however, necessary, estimates could be derived by considering the relationships between the coefficients in (13) and (14), which are $\alpha_i = \varphi_i + \psi_i$, $\beta_j = -\psi_j$, where the non-existing coefficients are assumed to be zero, and $\omega = \left(1 - \sum_{i=1}^{l^*}\varphi_i\right)\mu_{lz} - \left(1 + \sum_{j=1}^{s}\psi_j\right)\mu_{le}$. Note that the parametric part of the Semi-Log-GARCH can also be estimated directly using the R package **lgarch** (Sucarrat, 2015). Nonetheless, this approach will not be considered in the current paper.

In the following, 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.

```
# Calculate the centralized log-returns
dax_close <- smoots::dax$Close; dax <- diff(log(dax_close))
rt <- dax - mean(dax); yt <- log(rt ^ 2)
```

Subsequently, the previously described estimation procedure according to Sucarrat (2019) is implemented by estimating the trend function in the logarithm of the squared returns `yt` with the function `msmooth()` of the **smoots** package. More specifically, a local cubic trend is fitted, while employing *AlgA*.

```
# Step 1: Estimate the trend in the log-transformed data using 'smoots'
estim3 <- smoots::msmooth(yt, p = 3, alg = "A")
m_xt <- estim3$ye

# Step 2: Fit an ARMA model to the residuals
xi <- estim3$res
arma3 <- arima(xi, order = c(1, 0, 1), include.mean = FALSE)

# Step 3: Estimate further quantities and the total volatilities
mu_le <- -log(mean(exp(arma3$residuals)))
vol <- exp((xi - arma3$residuals + m_xt - mu_le) / 2)
```

For reference, `estim3.2 <- smoots::msmooth(yt, p = 1, alg = "A")` is called, i.e. a local linear trend under consideration of *AlgA* is estimated as well. The centralized log-returns and the log-transformed data with the two estimated trends (local linear: blue; local cubic: red) are displayed in Figures 3(a) and 3(b). Moreover, 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. Ultimately, 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 (see Step 2 in the code) is obtained from the residuals of the log-data

$$\tilde{\xi}_t = 0.9692\tilde{\xi}_{t-1} - 0.9221\varepsilon_{t-1} + \varepsilon_t, \tag{16}$$

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

$$\ln(h_t) = 0.0685 + 0.0471\ln\left(\zeta_{t-1}^2\right) + 0.9221\ln(h_{t-1}) \tag{17}$$

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

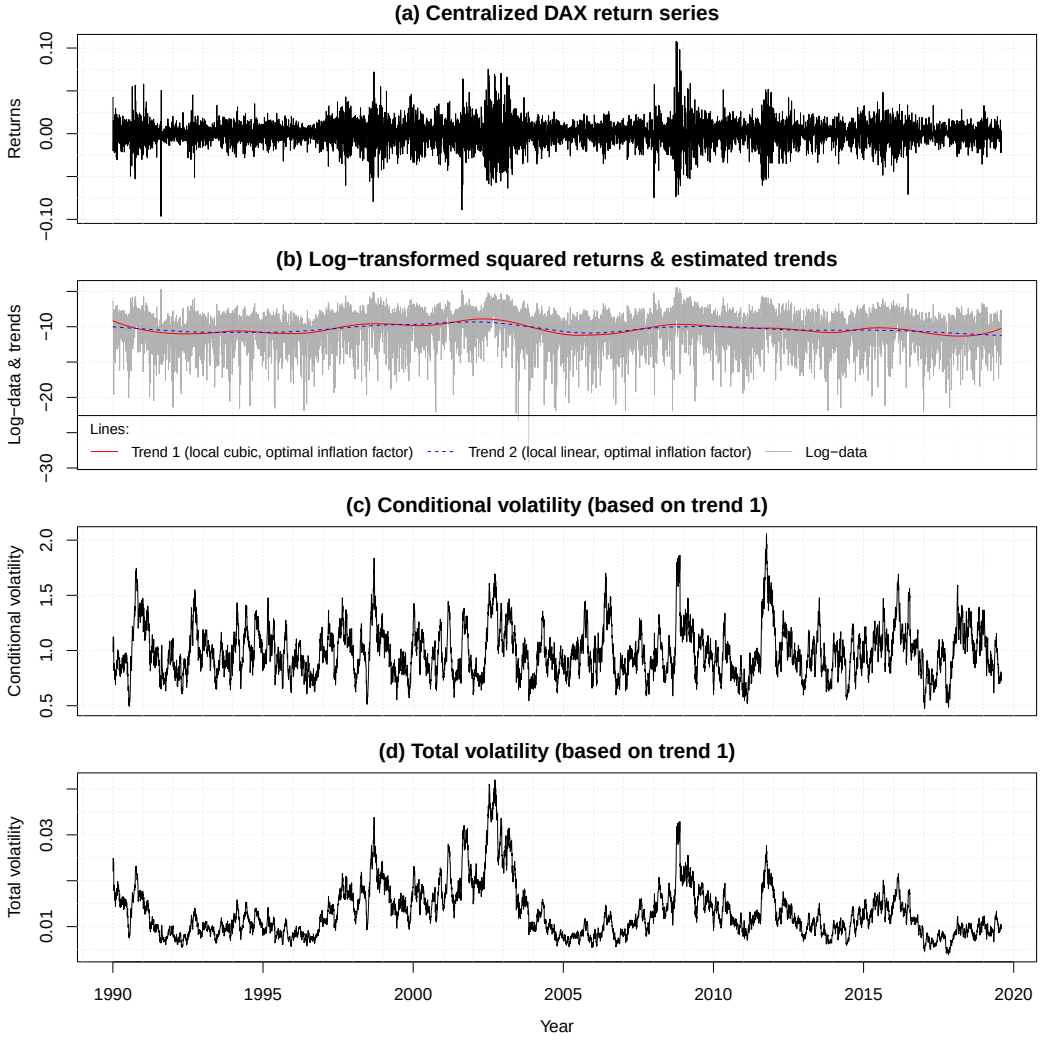

Figure 3: The demeaned DAX returns are displayed in (a). The log-transformed squared returns and the estimated trends are shown in (b). Based on the estimated local cubic trend, the corresponding estimated conditional and total volatilities are illustrated in (c) and (d), respectively.

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 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 generalization 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

$$V_t = g(x_t)\,\lambda_t e_t, \tag{18}$$

where $u_t = \lambda_t e_t$ follows a Log-ACD and $g \geq 0$ is a smooth mean function in V_t, $\lambda_t \geq 0$ is the conditional mean and e_t is an i.i.d. 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. The Semi-Log-ACD model can be similarly estimated. Discussion of those details is omitted. For further information we refer the reader to Forstinger (2018) and references therein.

```
# Calculate the logarithm of the index
V <- smoots::vix$Close; lnV <- log(V)
```

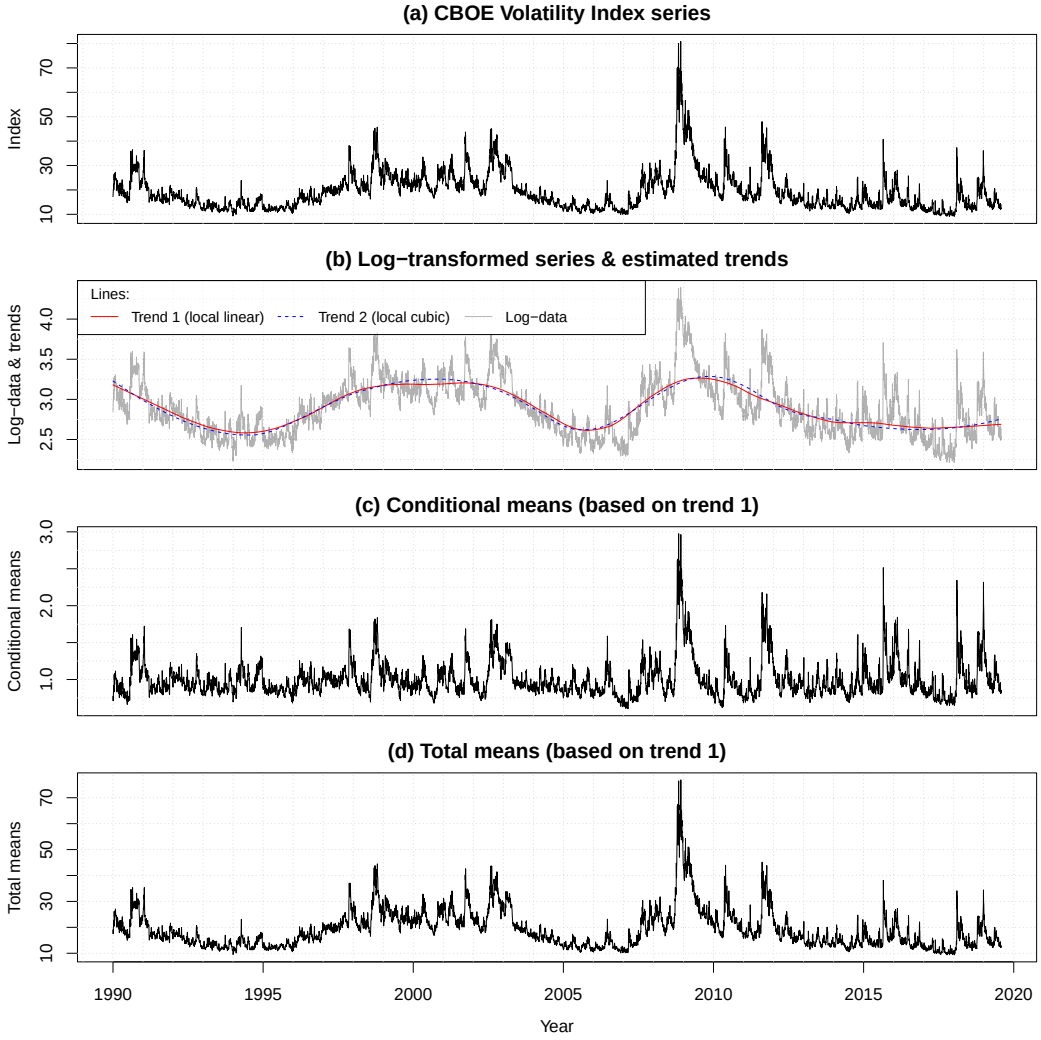

Figure 4: The VIX series is displayed in (a). The log-transformed index and the estimated trends are shown in (b). Based on the estimated local linear trend, the corresponding estimated conditional and total means are illustrated in (c) and (d), respectively.

```
# Step 1: Estimate the trend in the log-transformed data using 'smoots'
estim4 <- smoots::msmooth(lnV)
estim4.2 <- smoots::msmooth(lnV, p = 3, alg = "B")
m_xt <- estim4$ye

# Step 2: Fit an ARMA model to the residuals
xi <- estim4$res
arma4 <- arima(xi, order = c(1, 0, 1), include.mean = FALSE)

# Step 3: Estimate further quantities and the total means
mu_le <- -log(mean(exp(arma4$residuals)))
means <- exp(xi - arma4$residuals + m_xt - mu_le)
```

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 the local cubic estimates are smoother. From the residuals of the local linear approach the following ARMA(1, 1) model (see Step 2 in the code) is obtained:

$$\tilde{\xi}_t = 0.9626\tilde{\xi}_{t-1} - 0.0707\varepsilon_{t-1} + \varepsilon_t. \tag{19}$$

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

$$\ln(\lambda_t) = 0.0010 + 0.8919\ln(u_{t-1}) + 0.0707\ln(\lambda_{t-1}). \tag{20}$$

The estimated conditional means 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 future.

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. Data examples show 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. It is found that the errors in some examples could exhibit clear long memory. However, the current package is developed 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, such as the development of suitable forecasting procedures and tools for testing stationarity of the errors or linearity of the deterministic trend, should also be studied in the future and included in future versions of the **smoots** package.

9 Computational details

The numerical results in this paper were obtained using R 4.1.1 with the **smoots** 1.0.1 package and the **stats** 4.1.1 package. R itself and all packages used are available from CRAN at https://CRAN.R-project.org/.

Acknowledgments: This work was supported by the German DFG project GZ-FE-1500-2-1. 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**. We would also like to thank Dr. Marlon Fritz for helpful discussions.

Bibliography

D. M. Allen. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974. URL https://doi.org/10.1080/00401706.1974.10489157. [p182]

N. S. Altman. Kernel smoothing of data with correlated errors. *Journal of the American Statistical Association*, 85(411):749–759, 1990. URL https://doi.org/10.1080/01621459.1990.10474936. [p182]

M. Balcilar. *mFilter: Miscellaneous Time Series Filters*, 2019. URL https://CRAN.R-project.org/package=mFilter. R package version 0.1-5. [p182]

L. Bauwens and P. Giot. The logarithmic ACD model: An application to the bid-ask quote process of three NYSE stocks. *Annales d'Economie et de Statistique*, (60):117–149, 2000. URL https://doi.org/10.2307/20076257. [p183, 191]

M. Baxter and R. G. King. Measuring business cycles: Approximate band-pass filters for economic time series. *Review of Economics and Statistics*, 81(4):575–593, 1999. URL https://doi.org/10.1162/003465399558454. [p182]

J. Beran, Y. Feng, and S. Heiler. Modifying the double smoothing bandwidth selector in nonparametric regression. *Statistical Methodology*, 6(5):447–465, 2009. URL https://doi.org/10.1016/j.stamet.2009.04.001. [p182]

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986. URL https://doi.org/10.1016/0304-4076(86)90063-1. [p183]

P. Bühlmann. Locally adaptive lag-window spectral estimation. *Journal of Time Series Analysis*, 17(3):247–270, 1996. URL https://doi.org/10.1111/j.1467-9892.1996.tb00275.x. [p182, 183, 184]

R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, 50(4):987–1007, 1982. URL https://doi.org/10.2307/1912773. [p183]

R. F. Engle and J. G. Rangel. The Spline-GARCH model for low-frequency volatility and its global macroeconomic causes. *The Review of Financial Studies*, 21(3):1187–1222, 2008. URL https://doi.org/10.2139/ssrn.939447. [p189]

R. F. Engle and J. R. Russell. Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66(5):1127–1162, 1998. URL https://doi.org/10.2307/2999632. [p183, 191]

Y. Feng. Simultaneously modeling conditional heteroskedasticity and scale change. *Econometric Theory*, 20(3):563–596, 2004. URL https://doi.org/10.1017/S0266466604203061. [p189]

Y. Feng. On the asymptotic variance in nonparametric regression with fractional time-series errors. *Nonparametric Statistics*, 19(2):63–76, 2007. URL https://doi.org/10.1080/10485250701381737. [p182]

Y. Feng and S. Heiler. A simple bootstrap bandwidth selector for local polynomial fitting. *Journal of Statistical Computation and Simulation*, 79(12):1425–1439, 2009. URL https://doi.org/10.1080/00949650802352019. [p185]

Y. Feng and D. Schulz. *smoots: Nonparametric Estimation of the Trend and Its Derivatives in TS*, 2019. URL https://CRAN.R-project.org/package=smoots. R package version 1.0.1. [p182]

Y. Feng, T. Gries, and M. Fritz. Data-driven local polynomial for the trend and its derivatives in economic time series. *Journal of Nonparametric Statistics*, 32(2):510–533, 2020. URL https://doi.org/10.1080/10485252.2020.1759598. [p182, 183, 184, 185, 186]

S. Forstinger. *Modelling and Forecasting Financial and Economic Time Series Using Different Semiparametric ACD Models*. PhD thesis, Paderborn, Universität Paderborn, 2018. [p191]

M. Francisco-Fernández, J. Opsomer, and J. M. Vilar-Fernández. Plug-in bandwidth selector for local polynomial regression estimator with correlated errors. *Nonparametric Statistics*, 16(1-2):127–151, 2004. URL https://doi.org/10.1080/10485250310001622848. [p182]

C. Francq and G. Sucarrat. An exponential chi-squared QMLE for Log-GARCH models via the ARMA representation. *Journal of Financial Econometrics*, 16(1):129–154, 2018. URL https://doi.org/10.1093/jjfinec/nbx032. [p189]

C. Francq, O. Wintenberger, and J.-M. Zakoïan. GARCH models without positivity constraints: Exponential or Log GARCH? *Journal of Econometrics*, 177(1):34–46, 2013. URL https://doi.org/10.1016/j.jeconom.2013.05.004. [p183, 189]

T. Gasser, A. Kneip, and W. Köhler. A flexible and fast method for automatic smoothing. *Journal of the American Statistical Association*, 86(415):643–652, 1991. URL https://doi.org/10.1080/01621459.1991.10475090. [p182]

J. Geweke. Comment on: Modelling the persistence of conditional variances. *Econometric Reviews*, 5(1): 57–61, 1986. URL https://doi.org/10.1080/07474938608800097. [p183, 189]

J. D. Hart. Kernel regression estimation with time series errors. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(1):173–187, 1991. URL https://doi.org/10.1111/j.2517-6161.1991.tb01816.x. [p182]

E. Herrmann and M. Maechler. *lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth*, 2021. URL https://CRAN.R-project.org/package=lokern. R package version 1.1-9. [p182]

R. J. Hodrick and E. C. Prescott. Postwar US business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1):1–16, 1997. URL https://doi.org/10.2307/2953682. [p182]

A. Milhøj. *A Multiplicative Parameterization of ARCH Models*. University of Copenhagen, Department of Statistics, 1987. [p183, 189]

H.-G. Müller. *Nonparametric Regression Analysis of Longitudinal Data*. Springer, New York, NY, 1988. URL https://doi.org/10.1007/978-1-4612-3926-0. [p185]

J. L. Ojeda Cabrera. *locpol: Kernel Local Polynomial Regression*, 2018. URL https://CRAN.R-project.org/package=locpol. R package version 0.7-0. [p182]

J. Opsomer. Nonparametric regression in the presence of correlated errors. In *Modelling Longitudinal and Spatially Correlated Data*, pages 339–348. Springer, New York, NY, 1997. URL https://doi.org/10.1007/978-1-4612-0699-6_30. [p182]

S. G. Pantula. Modeling the persistence of conditional variances: A comment. *Econometric Reviews*, 5 (1):71–74, 1986. URL https://doi.org/10.1080/07474938608800099. [p183, 189]

D. Qiu. *rmaf: Refined Moving Average Filter*, 2015. URL https://CRAN.R-project.org/package=rmaf. R package version 3.0.1. [p182]

D. Ruppert, S. J. Sheather, and M. P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995. URL https://doi.org/10.1080/01621459.1995.10476630. [p182]

G. Sucarrat. *lgarch: Simulation and Estimation of Log-GARCH Models*, 2015. URL https://CRAN.R-project.org/package=lgarch. R package version 0.6-2. [p190]

G. Sucarrat. The log-GARCH model via ARMA representations. In *Financial Mathematics, Volatility and Covariance Modelling*, volume 2, pages 336–359. Routledge, London, 1 edition, 2019. URL https://doi.org/10.4324/9781315162737. [p183, 190]

G. Sucarrat, S. Grønneberg, and A. Escribano. Estimation and inference in univariate and multivariate log-GARCH-X models when the conditional density is unknown. *Computational Statistics & Data Analysis*, 100:582–594, 2016. URL https://doi.org/10.1016/j.csda.2015.12.005. [p190]

M. Wand. *KernSmooth: Functions for Kernel Smoothing Supporting Wand & Jones (1995)*, 2021. URL https://CRAN.R-project.org/package=KernSmooth. R package version 2.23-20. [p182]

Yuanhua Feng
Department of Economics
Faculty of Business Administration and Economics
Paderborn University
Warburger Straße 100, 33098 Paderborn
Germany
yuanhua.feng@uni-paderborn.de

Thomas Gries
Department of Economics
Faculty of Business Administration and Economics
Paderborn University
Warburger Straße 100, 33098 Paderborn
Germany
thomas.gries@uni-paderborn.de

Sebastian Letmathe
Department of Economics
Faculty of Business Administration and Economics
Paderborn University
Warburger Straße 100, 33098 Paderborn
Germany
sebastian.letmathe@uni-paderborn.de

Dominik Schulz
Department of Economics
Faculty of Business Administration and Economics
Paderborn University
Warburger Straße 100, 33098 Paderborn
Germany
dominik.schulz@uni-paderborn.de