tvReg: Time-varying Coefficient Linear Regression for Single and Multi-Equations in R

Isabel Casas
The University of Southern Denmark
Basque Center for Applied Mathematics

Rubén Fernández-Casal
University of A Coruña

Abstract

The source code of the package **tvReg** is publicly available for download from the Comprehensive R Archive Network. The six basic functions in this package are the `tvLM`, `tvAR`, `tvSURE`, `tvPLM`, `tvVAR` and `tvIRF`, which cover a large range of semiparametric models with time-varying coefficients. Moreover, this package provides methods for the graphical display of results, extraction of the residuals and fitted values, bandwidth selection, nonparametric estimation of the time-varying variance-covariance matrix of the error term, and four estimation procedures: the time-varying ordinary least squares implemented in the `tvOLS`, the time-varying generalised least squares in the `tvGLS`, the time-varying random effects in the `tvRE` and the time-varying random fixed effects in the `tvFE` methods. Applications to risk management, portfolio management, asset management and monetary policy are used as examples of these functions usage.

Keywords: Time-varying coefficients, nonparametric, SURE, panel, VAR, IRF, autoregressive, R.

1. Introduction

A very popular research area has been brewing in the field of kernel smoothing statistics applied to linear models with time-varying coefficients. In the field of econometrics, Robinson (1989) was the first to analyse these models for linear regressions with time-varying coefficients and stationary variables. Since then, this literature has extended to models with fewer restrictions in the dependence of the variables, to models with time dependence in the error term and to multi-equation models. More details can be found in the following sections. Although these models are potentially applicable to a large number of areas, no comprehensive computational implementation is, to our knowledge, formally available in any of the commercial programming languages. Especially, in the field of multi-equation models with varying coefficients.

Parametric linear models (LM) are probably the most popular and broadly used of all statistical models. Their application extends to, perhaps, every field of quantitative research. Just to mention some, they are found in biostatistics, finance, economics, business, climate, linguistics, psychology, engineering and oceanography. Linear regression is the first model that students learn to explain the relationship between variables in introductory Statistics courses. The autoregressive (AR) model is its extension to fit time series data that show short dependency with the past. Panel linear models (PLM) are also widely used to account for the heterogeneity in the cross-section and time dimensions. Seemingly unrelated equations (SURE) and vector autoregressive models (VAR) are the extensions of linear regressions and autoregressive models to the multi-equation framework. Programs with these algorithms are found in all major programming languages. Particularly in R, the function and methods `lm`, `ar` and `arima` are part of any basic R installation and fit linear regression and autoregressive models. The package **plm** (Croissant and Millo 2018, 2008) contain methods for panel data models. The package **systemfit** (Henningsen and Hamann 2007) allows the estimation of coefficients in systems of linear regressions, both with equation error terms correlated among equations (SURE) or uncorrelated. Finally, the package **vars** (Pfaff 2008) provides the tools to fit VAR models and impulse response functions (IRF). All these functions assume that the coefficients are constant. This assumption might not be true when a time series runs for a long period, and the relationships among variables do change.

In comparison to parametric models, the appeal of nonparametric models is their flexibility and robustness to functional form misspecification with spline-based and kernel-based regression methods being the two main nonparametric estimation techniques, citep[e.g.][]Eubank1999. However, fully nonparametric models are not appropriate when many regressors are in play, as their rate of convergence decreases with the number of regressors, the infamous "curse of dimensionality". In the case of cross-section data, one popular alternative to avoid this problem are the generalised additive models (GAM), introduced by Hastie and Tibshirani (1993). The GAM is a family of semiparametric models that extends parametric linear models by allowing for non-linear relationships of the explanatory variables and still retaining the additive structure of the model. In the case of time-series data, the most suitable alternative to nonparametric models is the linear models whose coefficients change over time or follow the dynamics of another random variable. In R, single-equation GAM models can be estimated with packages **gam** and **mgcv** whose methods are mainly designed to fit to cross-sectional data. Whereas functions in the **tvReg** are mainly concerned with time-series models both for a single-equation and multiple-equation models. An extension of **mgcv** to include time-varying coefficient VAR (TVVAR) models is implemented in the package **mgm** by (Haslbeck and Waldorp 2016). Thus, there exists some functionality overlapping between the **mgm** and the **tvReg**. However, the **mgm** lacks the time-varying impulse response function (TVIRF) which is essential to interpret the of a TVVAR model. On top of all this, the **tvReg** contains tools to estimate time-varying coefficients seemingly unrelated equation (TVSURE), panel linear models (TVPLM) and autoregressive (TVAR) models. Time-varying coefficient linear models can be expressed in state space form, which assumes that the coefficients change over time in a determined way for example, as a Brownian motion. These models can be estimated using the Kalman filter or Bayesian techniques, for instance. Packages **MARSS** (Holmes, Ward, and Wills 2012) and **bvarsv**(Krueger 2015) implement this approach for VAR models with time-varying parameters (TVP-VAR) which implement the Carter and Kohn (1994) algorithm.

Summing up, this paper presents a review of the most common time-varying coefficient linear models, their estimation using kernel smoothing techniques, and the usage of functions and methods in the package **tvReg**. Along these lines, Table 1 offers a glimpse at the **tvReg** full functionality, displaying a summary of its methods, classes and functions. The release of the **tvReg** provides computational tools to aid researchers of empirical disciplines that benefit from dynamic linear models of this kind.

The remainder of this paper is organised as follows. The theoretical frameworks for single

Function	Class	Methods for class	Functions for class
`tvLM`	`tvlm`	`bw`, `coef`, `confint`, `fitted`, `forecast`, `plot`, `predict`, `print`, `resid`, `summary`, `tvOLS`	
`tvAR`	`tvar`	`bw`, `coef`, `confint`, `fitted`, `forecast`, `plot`, `predict`, `print`, `resid`, `summary`, `tvOLS`	
`tvPLM`	`tvPLM`	`coef`, `confint`, `fitted`, `forecast`, `plot`, `predict`, `print`, `resid`, `summary`,	`tvRE`, `tvFE`
`tvSURE`	`tvsure`	`bw`, `coef`, `confint`, `fitted`, `forecast`, `plot`, `predict`, `print`, `resid`, `summary`, `tvGLS`	
`tvVAR`	`tvvar`	`bw`, `coef`, `confint`, `fitted`, `forecast`, `plot`, `predict`, `print`, `resid`, `summary`, `tvOLS`	`tvAcoef`, `tvBcoef`, `tvIRF`, `tvPhi`, `tvPsi`
`tvIRF`	`tvirf`	`coef`, `confint`, `plot`, `print` `summary`	

Table 1: Structure of the package **tvReg**.

and multi-equation models are presented in Section 2 and Section 4, respectively. Details of their implementation and usage into **tvReg** can be found in Section 3 and Section 5 with applications to economic and financial data. For completion, Section 6 discusses the nonparametric estimation of the variance-covariance function because it is part of functions `tvSURE` and `tvIRF`. Prediction and forecast are computed with the `predict` and `forecast` methods, which are addressed in Section 7. Finally, the recapitulation of the paper and plans for further development of the package **tvReg** is assigned to Section 8.

2. Single-equation linear models with time-varying coefficients

A classical linear model (LM) is generally expressed by

$$y_t = x_t^\top \beta + u_t, \quad t = 1, \ldots, T, \tag{1}$$

where y_t is the response or dependent variable, $x_t = (x_{1t}, x_{2t}, \ldots, x_{dt})^\top$ is a vector of regressors at time t, $\beta = (\beta_0, \beta_1, \ldots, \beta_d)^\top$ is the vector of coefficients and u_t is the error term which satisfies $\mathsf{E}(u_t|x_t) = 0$ and $\mathsf{E}(u_t^2|x_t) = \sigma^2$. The autoregressive model with p lags, AR(p), is a special case of LM, where the set of regressors can contain lagged values of y_t. The coefficients of these models can be estimated consistently with the ordinary least squares estimator (OLS),

$$\hat{\beta} = \left(X^\top X\right)^{-1} \left(X^\top Y\right),$$

under certain conditions. If the coefficients change with time, like in the time-varying coefficient linear model (TVLM) expressed by

$$y_t = x_t^\top \beta(z_t) + u_t, \quad t = 1, \ldots, T. \tag{2}$$

Note that there are not enough degrees of freedom in this case for a meaningful OLS estimation. Here, the dependent variable, regressors and error term are defined as in (1); however, the coefficients are not constant, but are functions of the variable z_t, the smoothing variable. Thus, $\beta(z_t) = (\beta_0(z_t), \beta_1(z_t), \ldots, \beta_d(z_t))^\top$ vary over time.

It is important to differentiate between two types of smoothing variables: 1) $z_t = \tau = t/T$ is the rescaled time with $\tau \in [0,1]$, and 2) z_t is the value at time t of the random variable Z. In other words, time-varying coefficients may be defined as unknown functions of time, $\beta(z_t) = f(\tau)$, or as unknown functions of a random variable, $\beta(z_t) = f(z_t)$. The former was firstly studied in Robinson (1989) for stationary processes and generalised to nonstationary processes and correlated errors by Chang and Martinez-Chombo (2003) and Cai (2007) among others. Recently, Chen, Gao, Li, and Silvapulle (2017) apply it to the Heterogeneous Auto-Regressive (HAR) model of Corsi (2009) for the realized volatility of S&P 500 index returns. It is a very flexible approach, but forecasts are not consistent because there is no information from the dependent variable at time $T+1$. On the other hand, the case of TVLM with $\beta(z_t) = f(z_t)$ has been studied when $\{(x_t, z_t, u_t)\}$ are iid or stationary processes by Hastie and Tibshirani (1993) and Cai, Fan, and Yao (2000); and nonstationary regressors or/and nonstationary z_t have been studied by Chang and Martinez-Chombo (2003), Cai, Li, and Park (2009), Sun, Cai, and Li (2013) and Gao and Phillips (2013). Das (2005), Xiao (2009) and Henderson, Kumbhakar, Li, and Parmeter (2015) have used the approach for instrumental variables, cointegration and SUR frameworks, respectively.

The time-varying coefficients are obtained by combining OLS and the local polynomial kernel estimator, which is extensively studied in Fan and Gijbels (1996). This will be denoted by the time-varying OLS (TVOLS) estimator herein. Two versions of this estimator are developed in **tvReg**: i) the TVOLS that uses the local constant (`lc`) kernel method, also known as the Nadaraya-Watson estimator; and ii) the TVOLS which uses the local linear (`ll`) method. Assuming that $\beta(\cdot)$ is twice differentiable, an approximation of $\beta(z_t)$ around z is given by the Taylor rule, $\beta(z_t) \approx \beta(z) + \beta(z)^{(1)}(z_t - z)$, where $\beta^{(1)}(z) = d\beta(z)/dz$ is its first derivative. The estimates resolve the following minimisation:

$$(\hat{\beta}(z_t), \hat{\beta}^{(1)}(z_t)) = \arg\min_{\theta_0, \theta_1} \sum_{t=1}^{T} \left[y_t - x_t^\top \theta_0 - (z_t - z) x_t^\top \theta_1 \right]^2 K_b(z_t - z).$$

Roughly, these methodologies fit a set of weighted local regressions with an optimally chosen window size. The size of these windows is given by the bandwidth b, and the weights are given by $K_b(z_t - z) = b^{-1} K(\frac{z_t - z}{b})$, for a kernel function $K(\cdot)$. The local linear estimator general expression is

$$\begin{pmatrix} \hat{\beta}_t \\ \hat{\beta}_t^{(1)} \end{pmatrix} = \begin{pmatrix} S_{T,0}(z_t) & S_{T,1}^\top(z_t) \\ S_{T,1}(z_t) & S_{T,2}(z_t) \end{pmatrix}^{-1} \begin{pmatrix} T_{T,0}(z_t) \\ T_{T,1}(z_t) \end{pmatrix} \tag{3}$$

with

$$S_{T,s}(z_t) = \frac{1}{T}\sum_{i=1}^{T} x_i^\top X_i (z_i - z_t)^s K\left(\frac{z_i - z_t}{h}\right)$$
$$T_{T,s}(z_t) = \frac{1}{T}\sum_{i=1}^{T} x_i^\top (z_i - z_t)^s K\left(\frac{z_i - z_t}{h}\right) y_i$$

and $s = 0, 1, 2$. The particular case of the local constant estimator is calculated by $\hat{\beta}_t = S_{T,0}^{-1}(z_t)T_{T,0}(z_t)$ and it is only necessary that $\beta(\cdot)$ has one derivative.

These two estimators are consistent and asymptotically normal for several types of dependency of $\{(x_t, z_t, u_t)\}$ (see dependency assumptions in the aforementioned literature). Necessary assumptions on the size of the bandwidth, kernel regularity and error moments are left out of this text as can be easily found in the related literature.

2.1. Time-varying coefficient AR model

Autoregressive models use lagged values of the dependent variable as predictors to explain the dynamics and development of a process over time. This dependency is linear in nature and the coefficients can be estimated by OLS. Variables in the model must be stationary to ensure a meaningful estimation of the coefficients. Mathematically, the AR(p) model is expressed by

$$y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} + \gamma_1 x_{1t} + \ldots + \gamma_d x_{dt} + u_t. \tag{4}$$

Note that the dependent variable, y_t, in (4) depends on its own lags and may also depend on other exogenous variables.

The time-varying coefficient AR (TVAR) model is an AR model whose coefficients are functions of z_t, which as before can be a rescaled time period or a random process. For example, the TVAR(p) has the p-lags of the dependent variables as regressors,

$$y_t = \beta_0(z_t) + \beta_1(z_t)y_{t-1} + \ldots + \beta_p(z_t)y_{t-p} + \gamma_1(z_t)x_{1t} + \ldots + \gamma_d(z_t)x_{dt} + u_t, \quad \text{for } t = 1, \ldots, T. \tag{5}$$

This function is a particular case of (2) and can be estimated using the TVOLS displayed in Equation (3). Again, variable z_t can represent the rescaled time, $\tau = t/T$, or an instance of a random variable. For example, Chen and Tsay (1993) and Chen and Liu (2001) define the functional coefficient AR (FAR) model as a TVAR whose coefficients are functions of the dependent variable lagged values,

$$y_t = \beta_0(y_{t-p}) + \beta_1(y_{t-p})X_{t-1} + \ldots + \beta_d(y_{t-p})X_{t-p} + u_t.$$

3. tvReg for single-equation linear models

The package **tvReg** implements the functions `tvLM` and `tvAR` to fit TVLM and TVAR models, respectively, to time series data. The `tvLM` follows the standards of the function `lm` with main arguments `formula` and `data`. The main arguments of the `tvAR` are `y`, which is a vector of values with the dependent variable, and the model number of lags `p`.

3.1. Standard usage of tvLM

The function `tvLM` fits a TVLM using the `tvOLS` method. The only mandatory argument is `formula`, which should be a single formula for a single-equation model. This arguments follows the standard regression `formula` in R. The function `tvLM` returns an object of the class attribute `tvlm`. For illustration, the following model is generated:

$$y_t = \beta_{1t} x_{1t} + \beta_{2t} x_{2t} + u_t, \quad t = 1, \ldots, T, \tag{6}$$

where $\beta_{1t} = \sin(2\pi\tau)$ and $\beta_{2t} = 2\tau$ with $\tau = t/T$ and $T = 1000$. The regressors, $x_{1t} \sim t_2$ (symmetric) and $x_{2t} \sim \chi^2_4$, are independent of the error term, $u_t \sim \chi^2_2$ which has an exponential dependency in the covariance matrix given by $Cov(u_t, u_{t+h}) = e^{-|h|/10}$. The process generation and the fitting of a classical LM, a TVLM and a GAM is shown in the following chunk:

```
> set.seed (42)
> N <-1000
> tau <- seq(1:N)/N
> d <- data.frame(tau, beta1 = sin(2 * pi * tau), beta2 = 2 * tau,
+                  x1 = rt(N, df = 2), x2 = rchisq(N, df = 4))
> error.cov <- exp(-as.matrix(dist(tau))/10)
> L <- t(chol(error.cov))
> error <-  L %*% rchisq(N, df = 2)
> d <- transform(d, y = x1 * beta1 + x2 * beta2 + error)
> ## LM
> lm1 <- stats::lm(y ~ x1 + x2, data = d)
> ## TVLM
> tvLM1 <- tvLM(y ~ x1 + x2,  data = d, bw = 0.05, est ="ll")
> ## GAM
> library("mgcv")
> gam1 <- mgcv::gam(y ~ s(tau, by = x1) + s(tau, by = x2),
+                              data = d)
```

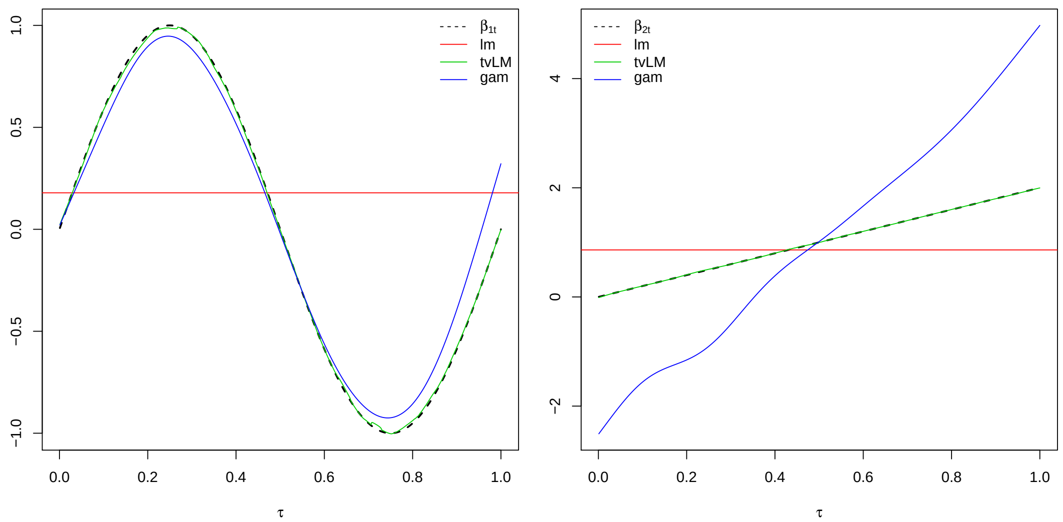
Figure 1 comparing the different estimators is generated with the code below to compare the above estimates with the true β_{1t}, β_{2t}.

```
> par(mar = c(4, 2, 1, 1))
> lm_beta <- coef(lm1)[c("x1", "x2")]
> tvLM_beta <- coef(tvLM1)[, c("x1", "x2")]
> gam_beta <- predict(gam1, type = "terms")/d[, c("x1", "x2")]
> for(i in 1:2)
+ {
+   betai <- d[[paste0("beta", i)]]
+   ylim <- range(betai, tvLM_beta[, i], gam_beta[,i])
+   plot(d$tau, betai, type = "l", ylim = ylim, lwd = 2,
+        xlab = expression(tau), ylab ="", lty = 2)
+   abline(h = lm_beta[i], col = 2)
+   lines(d$tau, tvLM_beta[, i], col =3)
```

```
+   lines(d$tau, gam_beta[, i], col =4)
+   if (i==2)
+     legend("topleft", c(expression(beta[2*t]), "lm",  "tvLM", "gam"),
+          col =1:4, lty = c(2,1, 1, 1), bty = "n")
+   else
+     legend("topright", c(expression(beta[1*t]), "lm",  "tvLM", "gam"),
+          col =1:4, lty = c(2,1, 1, 1), bty = "n")
+ }
```

As expected, the estimates from `lm` are constant and lie around the average of all β_{1t} and β_{2t}, while the estimates of `tvLM` and `gam` follow the dynamics of the varying coefficients. Besides the estimates of `gam` fit β_{1t} well, but not β_{2t} although the latter is a simple linear function. This issue is caused by the autocorrelated error term. On the other hand, the tvLM, although it requires for a longer computation time, it is able to fit both coefficients well.

Figure 1: Comparison of the `lm`, `tvLM` and `gam` estimates of β_{1t} and β_{2t}. The true values are plotted in black, the red line is the `lm` estimate, the green line refers to the `tvLM` estimates and the blue line is the `gam` estimates.



In addition to `formula`, the function `tvLM` has other arguments to control and choose the desired estimation procedure:

Smoothing random variable
: `tvLM` assumes by default that model coefficients are unknown functions of rescaled time, $\tau = t/T$ and therefore argument `z` is set to `NULL` by default. The user can modify this setting by entering a numeric vector in argument `z` with the values of the random smoothing variable over the corresponding time period.

Data
: Argument `data` can be used to specify the data frame or matrix with the model variables. If it is not specified, the function searches variables in the global environment or in the search path.

Bandwidth

When argument `bw` is set to `NULL` in `tvLM`, it is automatically selected by leave-one-out cross-validation. This minimisation can be a bit slow for large datasets, and it should be avoided if the user knows an appropiate value of the bandwidth for the required problem. The user can enter values in `bw` to obtain undersmoothed or oversmoothed estimates as needed.

Kernel type

The two choices for this argument are `tkernel = "Epa"` (default) and `tkernel ="Gaussian"`. The former refers to the Epanechnikov kernel, which is compact in [-1, 1]. The authors do not recommend the use of the Gaussian kernel because in general, it requires more calculations and it is slower.

Degree of local polynomial

The default estimation methodology is the Nadaraya-Watson or local constant, which is set as (`est = "lc"`) and it fits a constant at each interval defined by the bandwidth. The argument `est = "ll"` can be chosen to perform a local linear estimation (i.e., to fit a polynomial of order 1).

Singular fit

The `tvOLS` method used in the estimation wraps the `lm.wfit` method, which at default allows the fitting of a low-rank model, and the estimation coefficients can be `NA`s. The user can change the argument `singular.ok` to `FALSE`, so that the program stops in case of a low-rank model.

The package **tvReg** also includes the functionality to compute confidence intervals for the coefficients of class attributes `tvlm`, `tvar`, `tvplm`, `tvsure` and `tvirf` by extending the `confint` method. The algorithm in Chen *et al.* (2017) to calculate bootstrap confidence intervals has been adapted for all these class attributes. Argument `level` is set to 0.95 (95% confidence interval) by default. Argument `runs` (100 by default) is the number of resamples used in the bootstrapping calculation. Note that the calculation using `runs = 100` can take long, so we suggest to try a small value in `runs` first to get an initial intuition of the results. Because coefficients are time-varying, only wild bootstrap residual resampling is implemented. Two choices of wildbootstrap are allowed in argument `tboot`: the default one proposed in Mammen (1993) (`tboot = "wild"`); and the standard normal (`tboot = "wild2"`).

Coefficient estimates from all replications are stored in the `BOOT` variable. In this way, calculations do not need to be done again if the user chooses a different `level` for the same object. For example, the calculation of the 90% confidence interval of the object `tvLM1` is calculated with the `confint` method. Posteriorly, the 95% interval is calculated quickly because the resample calculations in the first interval are re-used for the second. Details in the chunk below:

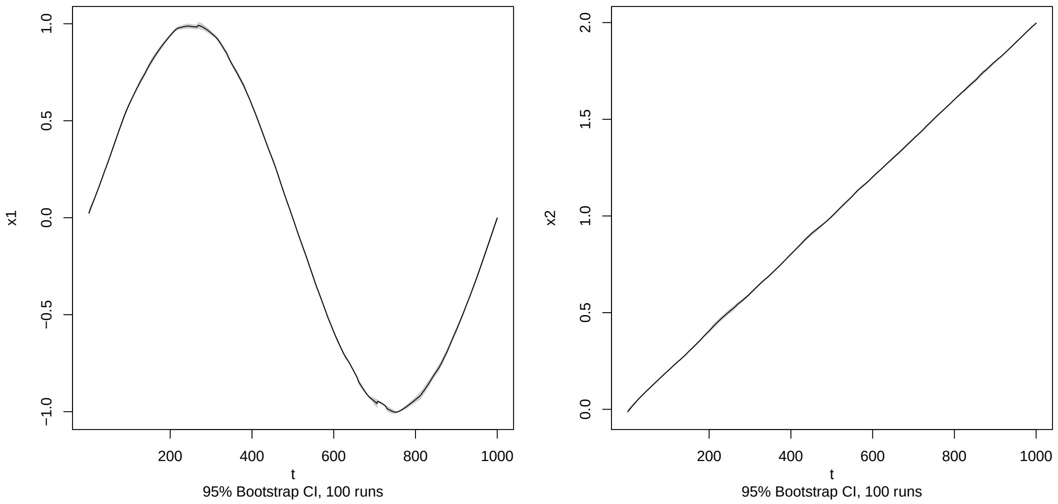
```
> tvLM1.90 <- confint (tvLM1, level = 0.90)
> tvLM1.95 <- confint(tvLM1.90, level = 0.95)
```

The `plot` method is implemented for each of the five class attributes in **tvReg**. The coefficient estimates and their confidence intervals (if calculated) are plotted for class attributes `tvlm`, `tvar`, `tvplm`, `tvsure` and `tvirf`. The fitted values and the residuals are plotted for class attribute `tvvar` because the interpretation of its coefficients is not relevant.

For illustration, the chunk below produces Figure 2 with the time-varying coefficients of `tvLM1.90`:

```
> plot(tvLM1.95, vars = 2:3)
```

Figure 2: Plots of the coefficients estimates of object `tvLM1` with a 95% confidence intervals.



3.2. Standard usage of tvAR

A TVAR model is a particular case of TVLM whose regressors contain lagged values of the dependent variable, `y`. The number of lags is given by the model order set in the argument `p`. Other exogenous variables can be included in the model using the argument `exogen`, which accepts a vector or a matrix with the same number of rows as the argument `y`. An intercept is included by default unless the user enters `type="none"` into the function call. Econometrically, this function also wraps the `tvOLS` estimator, which needs a bandwidth `bw` that is automatically selected when the user does not enter any number. An object of the class attribute `tvar` is returned by the function `tvAR`.

The user can provide additional optional arguments to modify the default estimation of the function `tvAR`. See Section 3.1 to understand the usage of arguments `bw`, `tkernel`, `est` and `singular.ok`. In addition, the function `tvAR` has the following arguments:

Type
The default model contains an intercept (i.e., it has a mean different from zero). The user can set argument `type = "none"`, so the model has mean zero.

Coefficient restrictions
An autoregressive process of order `p` does not necessarily contain all the previous p lags of y_t. Argument `fixed`, with the same format as in the function `arima` from the

package **stats**, permits to impose these restrictions. The order of variables in the model is: intercept (if any), lag 1, lag 2, ..., lag p and exogenous variable (if any). By default, the argument `fixed` is a vector of `NA`s with length the number of coefficients in the model. The user can enter a vector in the argument `fixed` with zeros in the positions corresponding to the restricted coefficients.

The following lines of code show an example of how to use the argument `fixed`. Time series y_t is generated as a constrained AR(6) process with the function `arima.sim`, and a constrained TVAR(6) with fitted intercept. The model only includes the first, third and sixth lags because `fixed = c(NA, NA, 0, NA, 0, NA, 0)`, it has a non-zero mean because `type = "const"` and no other exogenous variables are related to y_t. The bandwidth for the local linear, `est ="ll"`, is selected by leave-one-out cross-validation and it is large because the simulated model is linear. The `print` method is implemented for each of the class attributes in `tvReg`. It displays the mean of the coefficient estimates over time and the mean of their confidence interval, if calculated. It also displays the value of the bandwidth(s). Its usage is also illustrated for the class attribute `tvar` in the lines of code below:

```
> Y <- arima.sim(n = 100, list(ar = c(0.8, 0, -0.4, 0, -0.2))) + 2
> AR6 <- arima(Y, order = c(6, 0, 0), fixed = c(NA, NA, 0, NA, 0, NA, 0))
> tv.AR6 <- tvAR(Y, p = 6, type = "const",
+                  fixed = c(0, NA, NA, 0, NA, 0, NA), est ="ll")

Calculating regression bandwidth...

> coefficients(AR6)

       ar1        ar2        ar3        ar4        ar5        ar6  intercept
 1.3487075 -0.5099395  0.0000000 -0.2020011  0.0000000  0.2028098  0.0000000

> tv.AR6

Class:  tvar

Mean of coefficient estimates:
===============================
   y.l1    y.l2    y.l4    y.l6
 1.3575 -0.5213 -0.2076  0.2061

Bandwidth:  20
```

3.3. Application to risk management

The realized variance (RV) model was popularised in the financial literature by Andersen and Bollerslev (1998), who show that the use of intraday data can offer an accurate forecast of daily variance. It is defined as $RV_t = \sum_{i=1}^{N} r_{it}^2$, where r_{it} is the price return at minute i of day

t. The autocorrelation function of the RV also shows signs of long memory in the process, which can be accounted for by the heterogeneous RV (HAR) model of Corsi (2009),

$$RV_t = \beta_0 + \beta_1 RV_{t-1} + \beta_2 RV_{t-1|t-5} + \beta_3 RV_{t-1|t-22} + u_t. \tag{7}$$

Here, $RV_{t-1|t-k} = \frac{1}{k}\sum_{i=1}^{j} RV_{t-i}$. In this model, the current RV_t depends on its immediately previous value, RV_{t-1}, its medium-term memory factor, $RV_{t-1|t-5}$ and its long-term memory factor, $RV_{t-1|t-22}$. Basically, the HAR model may be seen as an AR(1) model with two exogenous variables.

It is likely that changes in the business cycles affect the coefficients in (7). Chen *et al.* (2017) coined the time-varying coefficient HAR as the TVC-HAR model, whose coefficients are functions of the rescaled time period. In the notation of this paper, the TVC-HAR is a TVAR(1) with two exogenous variables and can be fitted with the function `tvAR` as shown in the `R` chunk below, where the fitted model is named `tvHAR`. The `RV` dataset contains daily variables running from January 3, 1990 until December 19, 2007 that have been computed from 5 minute intraday data from Store (2017). This period coincides with the period in Bollerslev, Tauchen, and Zhou (2009). The variable names in this dataset are `RV`, `RV_lag`, `RV_week`, `RV_month` and `RQ_lag_sqrt` and correspond to the RV_t, RV_{t-1}, $RV_{t-1|t-5}$, $RV_{t-1|t-22}$ and $RQ_{t-1}^{1/2}$ in Model (7).

```
> data("RV")
> RV2 <- head(RV, 2000)
> HAR <- with(RV2, arima(RV, order = c(1, 0, 0),
+                        xreg = cbind(RV_week, RV_month)))
> tvHAR <- with(RV2, tvAR(RV, p = 1, bw = 0.8,
+                         exogen = cbind(RV_week, RV_month)))
```

Bollerslev, Patton, and Quaedvlieg (2016) extended the Model (7) further to control for the effect of the realized quarticity on the relationship between the future RV and its near past values. They present the HARQ model,

$$RV_t = \beta_0 + (\beta_1 + \beta_{1Q} RQ_{t-1}^{1/2})\ RV_{t-1} + \beta_2\ RV_{t-1|t-5} + \beta_3\ RV_{t-1|t-22} + u_t. \tag{8}$$

The HARQ model is a HAR model whose RV_{t-1} term's coefficient is a linear function of the squared root of the realized quarticity (RQ) at time $t-1$. The RQ changes over time and it will be larger during periods of more uncertainty. Casas, Mao, and Veiga (2018) appreciated that the variation of this coefficient may not be linear and proposed the TVHARQ model,

$$RV_t = \beta_0(z_t) + \beta_1(z_t)\ RV_{t-1} + \beta_2(z_t)\ RV_{t-1|t-5} + \beta_3(z_t)\ RV_{t-1|t-22} + u_t, \tag{9}$$

where the smoothing variable, z_t, is the $RQ_{t-1}^{1/2}$. This model is a TVAR(1) process and can be estimated with the function `tvAR` or with the function `tvLM` as it is shown in the chunk below.

```
> HARQ <- with(RV2, lm(RV ~ RV_lag + I(RV_lag*RQ_lag_sqrt) + RV_week +
+                      RV_month))
```

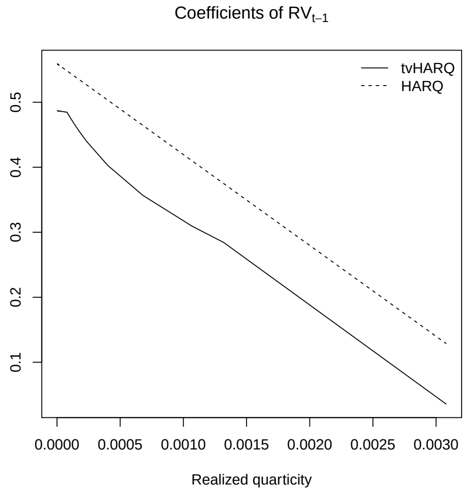
```
> tvHARQ <- with(RV2, tvAR(RV, p = 1, exogen = cbind(RV_week, RV_month),
+                          z = RQ_lag_sqrt, bw = 0.003))
> print(HARQ)

Call:
lm(formula = RV ~ RV_lag + I(RV_lag * RQ_lag_sqrt) + RV_week +
    RV_month)

Coefficients:
            (Intercept)                   RV_lag  I(RV_lag * RQ_lag_sqrt)
              2.168e-06                5.594e-01               -1.399e+02
                RV_week                 RV_month
              1.127e-01                2.014e-01

> print(tvHARQ)

Class:  tvar

Mean of coefficient estimates:
===============================
(Intercept)        y.l1     RV_week    RV_month
  2.497e-06   4.853e-01   1.312e-01   2.261e-01

Bandwidth:  0.003
```

The piece of code below creates Figure 3 with the coefficients of RV_{t-1} for both models. Note that an autoregressive loses the information in the `p` first data points.

```
> RV2 <- RV2[-1,]
> coef1 <- tvHARQ$coef[,2]
> coef2 <- HARQ$coef[2] + HARQ$coef[3] * RV2$RQ_lag_sqrt
> ylim <- range(coef1, coef2)
> RQ.sort <- sort(RV2$RQ_lag_sqrt)
> ind <- sort(RV2$RQ_lag_sqrt, index.return = TRUE)$ix
> plot(RQ.sort, coef1[ind], ylim = ylim, xlab = "Realized quarticity",
+      main = expression("Coefficients of " * RV[t-1]), ylab ="", type ="l")
> lines(RQ.sort, coef2[ind], lty = 2)
> legend("topright", c("tvHARQ", "HARQ"), col = 1, lty = c(1, 2), bty ="n")
```

It is appreciated in Figure 3 that the relationship between the RV and its first lag decreases linearly as the lagged RQ increases in the HARQ model. This effect is also observable in the TVHARQ model in a nonlinear manner: values of the lagged RQ near zero correspond to coefficients around 0.5, and the coefficient decreases down to around 0.3 for larger realized quarticity. An outlier for large RQ is present in both models with similar values of the coefficients.

Figure 3: Coefficients of RV_{t-1} for the HARQ and TVHARQ models.



4. Multi-equation linear models with time-varying coefficients

A multi-equation model formed by a set of linear models is defined when each equation has its own dependent variable and possible different regressors. Seemingly unrelated equations, panel data models and vector autoregressive models are included in this category.

4.1. Time-varying coefficients SURE

The SURE was proposed by Zellner (1962), motivated by the idea that several variables may be related in the way they vary (i.e, the variance-covariance matrix of the system error term is nondiagonal). This model is referred to as the seemingly unrelated equations model (SURE). The SURE model is useful to exploit the correlation structure between the error terms of each equation. Suppose that there are N linear regressions of different dependent variables,

$$y_i = X_i^\top \beta_i + u_i \quad i = 1, \dots, N, \tag{10}$$

where $y_i = (y_{i1}, \dots, y_{iT})^\top$ denotes the values over the recorded time period of the $i-$th dependent variable. Each equation in (10) may have a different number of exogenous variables, p_i. The regressors for equation i are $X_i = (x_{i1}, \dots, x_{ip_i})$, from which each element is a vector of dimension $T \times 1$. The constant coefficients of equation i are $\beta_i = (\beta_{i1}, \dots, \beta_{ip_i})^\top$. The error term $u_i = (u_{i1}, \dots, u_{iT})$ is a random process such that $\mathbb{E}(u_{it}) = \mathbb{E}(u_{it}|x_{it}) = 0$ and $\mathbb{E}(u_{it}u_{i't'}) = \delta_{tt'}\sigma_{ii't}$, where $\delta_{tt'} = 0$ if $t \neq t'$ and 1 if $t = t'$. Stacking the N equations on top of each other, the general system can be written in matrix form:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & X_N \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = X\beta + u. \tag{11}$$

Extending Model (11) to a model with time-varying coefficients, we obtain the TVSURE whose compact formula at each time t is given by

$$Y_t = X_t\beta(z_t) + U_t \quad i = 1, \ldots, N \quad t = 1, \ldots, T, \tag{12}$$

where $Y_t = (y_{1t} \ldots y_{Nt})^\top$, $X_t = diag(x_{1t} \ldots x_{Nt})$ and $\beta_{z_t} = \left(\beta_1(z_t)^\top, ..., \beta_N(z_t)^\top\right)^\top$ is a vector of order $P = p_1 + p_2 + \ldots + p_N$. The error vector, $U_t = (u_{1t} \ldots u_{Nt})^\top$, has zero mean and covariance matrix $\mathbb{E}\left(U_t U_t^\top\right) = \Sigma_t$ with elements $\sigma_{ii't}$. As before, the smoothing variable z_t may be t/T or the value of a random variable at time t.

The estimation may be done separately for each equation as if there is no correlation in the error term across equations; i.e, using the estimator TVOLS in Equation (3). System (11) has a total of N different TVLMs with possibly N different bandwidths, b_i. A second option is to use the correlation matrix of the error term. This is called the time-varying generalised least squares (TVGLS) estimation. Its mathematical expression is the same as (3) with the following matrix components:

$$\begin{aligned} S_{T,s}(z_t) =& \frac{1}{T}\sum_{i=1}^{T} X_i^\top K_{B,it}^{1/2} \Sigma_i^{-1} K_{B,it}^{1/2} X_i (Z_i - z_t)^s \\ T_{T,s}(z_t) =& \frac{1}{T}\sum_{i=1}^{T} X_i^\top K_{B,it}^{1/2} \Sigma_i^{-1} K_{B,it}^{1/2} Y_i (Z_i - z_t)^s, \end{aligned} \tag{13}$$

where $K_{B,it} = diag(K_{b_1,it}, ..., K_{b_N,it})$ and $K_{b_i,it} = (Tb_i)^{-1}K((Z_i - z_t)/(Tb_i))$ is the matrix of weights introducing smoothness according to the vector of bandwidths, $B = (b_1, \ldots, b_N)^\top$. Note that this minimisation problem accounts for the time-varying structure of the variance-covariance matrix of the errors, Σ_t.

The TVGLS assumes that the error variance-covariance matrix is known. In practice, this is unlikely and it must be estimated, resulting in the Feasible TVGLS estimator (TVFGLS). This estimator consists of two steps:

Step 1 Estimate Σ_t based on the residuals of a line by line estimation (i.e, when Σ_t is the identity matrix). If Σ_t is known to be constant, the sample variance-covariance matrix from the residuals is a consistent estimator of it. If Σ_t changes over time, a nonparametric estimator such the one explained in Section 6 is a consistent alternative.

Step 2 Estimate the coefficients of the TVSURE by plugging in $\hat{\Sigma}_t$ from Step 1 into Equation (13).

To ensure a good estimation of Σ_t, the iterative TVFLGS may be used. First, do Steps 1-2 as above to obtain the residuals from Step 2, and repeat Step 2 until the estimates of Σ_t converge or the maximum number of iterations is reached.

The case of TVSURE has been studied by Henderson *et al.* (2015) when the smoothing variable is a random variable changing with time and by Orbe, Ferreira, and Rodriguez-Poo (2005) and Casas, Ferreira, and Orbe (2019a) when the smoothing variable is τ.

4.2. Time-varying coefficients panel data models

Panel data models are a particular case of SURE models with a the same variables for each equation but measured for different cross-section units, such as countries, and for different points in time. All equations have the same coefficients apart from the intercept which can be different for different cross-sections. Therefore, the data from all cross-sections can be pooled together. The individual effects, α_i, account for the heterogeneity imbeded in the cross-section dimension. This package only take into account balanced panel datasets, i.e. with the same number of data points for each cross-section unit. Theoretically, a panel data model looks like

$$y_{it} = \alpha_i + x_{it}^\top \beta + u_{it} \qquad \Sigma = \begin{pmatrix} \sigma_\nu^2 & \sigma_\alpha^2 & \dots & \sigma_\alpha^2 \\ \sigma_\alpha^2 & \sigma_\nu^2 & \dots & \sigma_\alpha^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\alpha^2 & \sigma_\alpha^2 & \dots & \sigma_\nu^2 \end{pmatrix}, \quad i = 1, \dots, N, \tag{14}$$

for $t = 1, \dots, T$. The indiosyncratic error is not serially correlated, $E(u_t, u_s) = 0, s \neq t$ with mean zero and a constant variance σ_u^2. The α_i is a random variable with variance σ_α^2. In addition, $\sigma_\nu^2 = \sigma_u^2 + \sigma_\alpha^2$. Some of the three classical estimators of panel data models are explained below. Details on the necessary conditions for asymptotic results can be found in Croissant and Millo (2008); Wooldridge (2010); Croissant and Millo (2018), among others.

1. The pooled ordinary least square (POLS) estimator is given by

$$\hat\beta_{POLS} = (X^\top X)^{-1} X^\top Y,$$

   where $Y = (y_{11}, \dots, y_{1T}, \dots, y_{N1}, \dots, y_{NT})^\top$, X is defined similar to Y. This estimator ignores the panel structure. It assumes that $E(\alpha_i|X_i) = 0$. It can be proven to be consistent and asymptotically normal under certain conditions. However, it is not efficient and t-, F-, z- and Wald-tests based on its standard errors are not valid.

2. The random effects (RE) estimator corrects for this inefficiency by considering the estimation of Σ from the POLS estimation residuals,

$$\hat\beta_{RE} = (X^\top \hat\Sigma^{-1} X)^{-1} X^\top \hat\Sigma^{-1} Y.$$

3. The fixed effects (FE) or within estimator that considers that $E(\alpha_i|X_i) \neq 0$,

$$\hat\beta_{FE} = (\ddot X^\top \ddot X)^{-1} \ddot X^\top \ddot Y.$$

   The variables elements are demeaned over time and therefore all time-independent variables, including α_i disappear after the transformation: $\ddot y_{it} = y_{it} - \bar y_i$, $\ddot x_{itk} = x_{itk} - \bar x_{ik}$, $\ddot u_{it} = u_{it} - \bar u_i$.

These models are not able to show the coefficient dynamics which can be corrected using a time-varying coefficients panel data model. Recent developments in this kind of models can be found in Sun, Carroll, and Li (2009); Dong, C. Jiti Gao, J. and Peng, B. (2015); Dong and Peng (2018); Casas, Gao, Peng, and Xie (2019b) among others, with general model,

$$y_{it} = \alpha_i + x_{it}^\top \beta(z_t) + u_{it} \quad i = 1, \dots, N, \quad t = 1, \dots, T. \tag{15}$$

Note that the smoothing variable only changes over time, not like in the SURE model where it changed over i and t. The three correspondent estimator are:

1. The time-varying pooled ordinary least squares (TVPOLS) has the same expression than estimator (3) with the following terms:

$$
\begin{aligned}
S_{T,s}(z_t) =& X^\top K^*_{b,t} X (Z-z_t)^s \\
T_{T,s}(z_t) =& X^\top K^*_{b,t} Y (Z-z_t)^s,
\end{aligned}
\tag{16}
$$

   where $K^*_{b,t} = I_N \otimes \text{diag}\{K_b(z_1 - z_t), \dots, K_b(z_T - z_t)\}$. Note that it is not possible to ignore the panel structure in the semiparametric model because the coefficients change over time. The consistency and asymptotic normality of this estimator needs the classical assumptions about the kernel and the regularity of the coefficients, available in the related literature.

2. The time-varying random effects (TVRE) estimator is also given by Equation (16) with a non-identity Σ:

$$
\begin{aligned}
S_{T,s}(z_t) &= X^\top K^{*1/2}_{b,t} \Sigma_t^{-1} K^{*1/2}_{b,t} X (Z-z_t)^s \\
T_{T,s}(z_t) &= K^{*1/2}_{b,t} \Sigma_t^{-1} K^{*1/2}_{b,t} Y (Z-z_t)^s.
\end{aligned}
\tag{17}
$$

   Note that this is a simpler case of (13) with the same bandwidth for all equations. The variance-covariance matrix is estimated in the same way using the residuals from the TVPOLS and it may be an iterative algorithm until convergence of the coefficients.

3. The time-varying fixed effects (TVFE) estimator. Unfortunately, the transformation for the within estimation does not work in the time-varying coefficients model because the coefficients depend on time (Sun *et al.* 2009, explain the issue in detail). Therefore, it is necessary to make the assumption that $\sum_{i=1}^N \alpha_i = 0$ for identification. The terms in the TVFE estimator are:

$$
\begin{aligned}
S_{T,s}(z_t) =& X^\top W_{b,t} X (Z-z_t)^s \\
T_{T,s}(z_t) =& X^\top W_{b,t} Y (Z-z_t)^s,
\end{aligned}
\tag{18}
$$

   where $W_{b,t} = D_t^\top K^*_{b,t} D_t$, $D_t = I_{NT} - D(D^\top K^*_{b,t} D)^{-1} D^\top K^*_{b,t}$, $D = (-1_{N-1}, I_{N-1})^\top \otimes 1_T$, and 1_k is the unity vector of length k. The fixed effects are given by,

$$
\hat{\alpha} = (D^\top K^*_{b,t} D)^{-1} D^\top K^*_{b,t} (Y - X^\top \beta).
$$

   Finally, $\hat{\alpha}_i = \frac{1}{T} \sum_{t=1}^T \alpha_{it}$ for $i = 2, \dots, N$.

4.3. Time-varying coefficient VAR model

Macroeconomic econometrics experienced a revolution when Sims (1980) presented the vector autoregressive (VAR) model: a new way of summarising relationships among several variables while getting around the problem of endogeneity of structural models. The model coefficients and variance-covariance matrix may be estimated by maximum likelihood, OLS or GLS. VAR coefficients and the variance-covariance matrix do not have a direct economic interpretation;

however, it is possible to use them to recover a structural model by imposing a number of restrictions and so analyse the transmission of a shock, for example, a new monetary policy, to the macroeconomy using the impulse response function (IRF). Lütkepohl (2005) dive into the theoretical properties of these models in detail.

The data generation VAR(p) process is an N-equation system of AR(p) processes,

$$y_t = A_1 y_{t-1} + \ldots + A_p y_{t-p} + u_t,$$

where $y = (y_{1t}, \ldots, y_{Nt})$, the A_i are $N \times N$ coefficient matrices and u_t is the error term with mean zero and $E(u_t u_s^\top) = \Sigma_u$ if $t = s$, and zero otherwise. When this process is covariance-stationary, it can be written in its Wold representation,

$$y_t = \sum_{i=0}^{\infty} \Phi_i u_{t-i},$$

where $\Phi_0 = I_N$ and

$$\Phi_i = \sum_{j=1}^{i} \Phi_{i-j} A_j, \quad i = 1, 2, \ldots, \tag{19}$$

with $A_j = 0$ for $j > p$. The elements Φ_i are commonly interpreted as the impulse responses of the system. Once A_j are known, the IRF can be calculated by the recursive algorithm (19). To ease interpretation, many authors prefer to orthogonalise the error term. So they choose a matrix P, such that $\Sigma_u = PP^\top$ and define

$$y_t = \sum_{i=0}^{\infty} \Psi_i w_{t-i},$$

where $\Psi_i = \Phi_i P$ and $w_t = P^{-1} u_t$. The accumulated response of a shock over a period is defined by $\Theta_j = \sum_{i=0}^{j} \Phi_i$ and its orthogonal version by $\Xi_j = \sum_{i=0}^{j} \Psi_i$, which are quantities very commonly used in the interpretation of the effect of macroeconomic policy effects (see, for example, Iwata and Wu 2006; Vargas-Silva 2008; Ellahie and Ricco 2017).

Thus, the VAR model has lagged values of y_t as regressors to which further exogenous variables can be added as regressors. Unless the model is constrained, all variables are the same for every equation, which simplifies the algebra. The time-varying coefficient vector autoregressive model (TVVAR) is assume to be an N-dimensional system of TVAR(p) processes like

$$Y_t = A_{0,t} + A_{1,t} Y_{t-1} + \ldots + A_{p,t} Y_{t-p} + U_t, \quad t = 1, 2, \ldots, T. \tag{20}$$

In Equation (20), $Y_t = (y_{1t}, \ldots, y_{Nt})^\top$ and coefficient matrices at each point in time $A_{j,t} = (a_{1t}^j, \ldots, a_{Nt}^j)$, $j = 1, \ldots, p$ are of dimension $N \times N$. Then, notation $A_{j,t}$ means that the elements of this matrix are unknown functions of either the rescaled time value, τ, or of a random variable at time t. The innovation, $U_t = (u_{1t}, \ldots, u_{Nt})$, is an N-dimensional identically distributed random variable with $E(U_t) = 0$ and possibly a time-varying positive definite variance-covariance matrix,

$$E(U_t U_s^\top) = \begin{cases} \Sigma_t & \text{if t=s} \\ 0 & \text{otherwise.} \end{cases}$$

When matrix $A_{j,t}$ is a function of τ, then process (20) is locally stationary in the sense of (Dahlhaus 1997), which occurs when the functions in matrices $A_{j,t}$ are constant or change smoothly over time. Then, process (20) at time t has a well defined unique solution given by the Wold representation,

$$\bar{y}_t = \sum_{j=0}^{\infty} \Phi_{j,t} U_{t-j}, \tag{21}$$

such that $|Y_t - \bar{y}_t| \rightarrow 0$ almost surely. Matrix $\Phi_{0,t} = I_N$ and matrix $\Phi_{s,t} = \sum_{j=1}^{s} \Phi_{s-j,t} A_{j,t}$ for horizons $s = 1, 2, \ldots$ As for the constant model, $\Phi_{s,t}$ are the time-varying coefficient matrices of the impulse response function (TVIRF). Its element (t, i, j) may be interpreted as the expected response of $y_{i,t+s}$ to an exogenous shock of $y_{j,t}$ ceteris paribus lags of y_t when the innovations are orthogonal. Otherwise, an orthogonal TVIRF can be found as $\Psi_{j,t} = \Phi_{j,t} P_t$ for $\Sigma_t = P_t P_t^{\top}$, the Cholesky decomposition of Σ_t at time t.

In the macroeconomic literature, the Bayesian estimation of process (20) has attracted a lot of attention in recent years driven by results in Cogley and Sargent (2005); Primiceri (2005) and Kapetanios, Mumtaz, Stevens, and Theodoridis (2012). In their approach, the coefficients are assumed to follow a random walk. Recently, Kapetanios, Marcellino, and Venditti (2017) study the inference of the local constant estimator of a TVVAR(p) for large sets, and they find an increase in the forecast accuracy in comparison to the forecast accuracy of the VAR(p).

5. tvReg for multi-equation linear models

The function `tvPLM` in **tvReg** fits time-varying coefficient panel data models. Its main arguments are `formula`, describing the dependent variable and regressors which are common for all cross-sections, and `data`. It returns an object of the class attribute `tvplm`. The function `tvSURE` fits TVSURE models whose main argument is `formula`, which now is a list of objects of the class attribute `formula`, one formula for each equation. Equations may have a different number of regressors, and these do not have to be the same. An object of the class attribute `tvsure` is returned. In addition, the function `tvVAR` fits TVVAR models to given data. Its two main arguments are a matrix of dependent variables, `y`, with as many columns as equations and the number its lagged values, `p`, involved in the problem. It returns an object of the class attribute `tvvar`. Coefficients of TVVAR models are not interpretable. Instead, the TVIRF is coded in the function `tvIRF`, which takes an object of the class attribute `tvvar` and returns an object of the class attribute `tvirf`.

5.1. Standard usage of tvSURE

The main argument of this function is a list of formulas, one for each equation. The `formula` follows the format of `formula` in the package **systemfit**, which implements estimators of parametric multi-equation models with constant coefficients. In particular, the function `systemfit` can estimate the coefficients of SURE models by OLS, where the estimation is done equation by equation assuming no correlation in the error variance-covariance matrix, and by FGLS, which uses the information from the error variance-covariance matrix in the estimation.

The `tvSURE` wraps the `tvOLS` and `tvGLS` methods to estimate the coefficients of TVSURE models. The `tvOLS` method is used by default, calculating estimates for each equation independently with different bandwidths, `bw`. The user is able to enter a set of bandwidths or a single bandwidth to be used in the estimation instead. The `tvGLS` method has argument `Sigma` where the variance-covariance matrix of the error may be entered. If this matrix is known, then the estimated procedure is the TVGLS using Equations (3) or (13). Otherwise, if `Sigma = NULL`, the variance-covariance matrix Σ_t is estimated using the function `tvCov`, which is discussed in Section 6. The estimate of this matrix is used in Equation (13), named as the time-varying feasible GLS (TVFGLS) estimator.

The user can provide additional optional arguments to modify the default estimation. See section "Standard usage of `tvLM`" for details on arguments `z`, `data`, `bw`, `est`, `tkernel` and `singular.ok`. Note that the current version only allows one single smoothing random variable, `z`, for all equations and balanced panels. The user can restrict certain coefficients in the TVSURE model using arguments `R` and `r`. Note that the restriction is done by setting those coefficients to a constant. Furthermore, argument `method` defines the type of estimator to be used. The possible choices in argument `method` are:

1. `"tvOLS"` for a line by line estimation, meaning that the error variance-covariance matrix, Σ, is the identity matrix.

2. `"tvGLS"` to estimate the coefficients of the system using Σ_t, for which the user must enter it in argument `Sigma`. Argument `Sigma` takes either a symmetric `matrix` or an `array`. If `Sigma` is a `matrix` (constant over time) then it must have dimensions `neq` $\times$ `neq`, where `neq` is the number of equations in the system. If Σ_t is considered to change with time, then argument `Sigma` is an `array` of dimension `neq` $\times$ `neq` $\times$ `obs`, where the last dimension measures the number of time observations. Note that if the user enters a diagonal variance-covariance matrix with diagonal values different from one, then a time-varying weighted least squares is performed. If `method = "tvGLS"` is entered but `Sigma = NULL`, then `tvSURE` is fitted as if `method = "tvOLS"` and a warning is issued.

3. `"tvFGLS"` to estimate the coefficients of the system using an estimated time-varying variance-covariance matrix. By default, only one iteration is performed in the estimation of Σ, unless argument `control` indicates otherwise. The user can choose the maximum number of iterations or the level of tolerance in the estimation of Σ. See example the below for details.

The package **systemfit** contains the `Kmenta` dataset, which was first described in Kmenta (1986), to example the usage of the function `systemfit` to fit SURE models. This example has two equations: i) a demand equation, which explains how food consumption per capita, `consump`, depends on the ratio of food price, `price`; and disposable income, `income`; and ii) a supply equation, which shows how consumption depends on `price`, ratio prices received by farmers to general consumer prices, `farmPrice`; and a possible time trend, `trend`. Mathematically, this SURE model is

$$\begin{aligned} consump_t =& \beta_{10} + \beta_{11} price_t + \beta_{12} income_t + u_{1t} \\ consump_t =& \beta_{20} + \beta_{21} price_t + \beta_{22} farmPrice_t + \beta_{23} t + u_{2t}. \end{aligned} \tag{22}$$

The code below defines the system of equations using two `formula` calls which are put into a `list`.

```
> data("Kmenta", package = "systemfit")
> eqDemand <- consump ~ price + income
> eqSupply <- consump ~ price + farmPrice + trend
> system <- list(demand = eqDemand, supply = eqSupply)
```

Two parametric models are fitted to the data using the function `systemfit`: one assuming that there is no correlation of the errors setting (the default), `OLS.fit` below; and another one assuming the existence of correlation in the system error term setting `method = "SUR"`, `FGLS1.fit` below. Arguing that the coefficients in (22) may change over time, the corresponding TVSUREs are fitted by using the the function `tvSURE` with the default in the argument `method` and by `method = "tvFGLS"`, respectively. They are denoted by `tvOLS.fit` and `tvFGLS1.fit`.

```
> OLS.fit <- systemfit::systemfit(system, data = Kmenta)
> FGLS1.fit <- systemfit::systemfit(system, data = Kmenta, method = "SUR")
> tvOLS.fit <- tvSURE(system, data = Kmenta)

Calculating regression bandwidth...

> tvFGLS1.fit <- tvSURE(system, data = Kmenta, method = "tvFGLS")

Calculating regression bandwidth...
Calculating variance-covariance estimation bandwidth...
```

In the previous chunk, the FGLS and TVFGLS estimators use only one iteration. However, the user can choose the iterative FGLS and the iterative TVFGLS models, which estimate the coefficients iteratively until convergence. The convergence level can be chosen with the argument `tol` (1e-05 by default) and the argumenter `maxiter` with the maximum number of iterations. The following chunk illustrates its usage:

```
> FGLS2.fit <- systemfit::systemfit(system, data = Kmenta, method = "SUR",
+                                   maxiter = 100)
> tvFGLS2.fit <- tvSURE(system, data = Kmenta, method = "tvFGLS",
+                       control = list(tol = 0.001, maxiter = 100))

Calculating regression bandwidth...
Calculating variance-covariance estimation bandwidth...
```

Some of the coefficients can be restricted to have a certain constant value in `tvSURE`. This can aid statistical inference to test certain conditions. See an example of this below. Matrix `R` has as many rows as restrictions in `r` and as many columns as regressors in the model. In this case, Model (22) has 7 coefficients which are ordered as they appear in the list of formulas. Note that the time-varying coefficient of the variable `trend` is redundant when an intercept is included in the second equation of the TVSURE. Therefore, we want to restrict its coefficient two zero. We would also like to impose $\beta_{11,t} - \beta_{21,t} = 0.5$:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \beta_{10,t} \\ \beta_{11,t} \\ \beta_{12,t} \\ \beta_{20,t} \\ \beta_{21,t} \\ \beta_{22,t} \\ \beta_{23,t} \end{pmatrix} = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$$

This is done with the `tvSURE` as follows:

```
> Rrestr <- matrix(0, 2, 7)
> Rrestr[1, 7] <- 1
> Rrestr[2, 2] <- 1
> Rrestr[2, 5] <- -1
> qrestr <- c(0, 0.5)
> TVFGLS.rest <- tvSURE(system, data = Kmenta, method = "tvFGLS",
+                        R = Rrestr, r = qrestr,
+                        bw = tvFGLS1.fit$bw, bw.cov = tvFGLS1.fit$bw.cov)
```

5.2. Application to asset management

Recently, several studies have argued that the three-factor model by Fama and French (1993) does not explain the whole variation in average returns. In this line, Fama and French (2015) added two new factors that measure the differences in profitability (robust and weak) and investment (conservative and aggressive), creating their five-factor model (FF5F). This model has been applied in Fama and French (2017) to analyse the international markets. A time-varying coefficients version of the FF5F has been studied in Casas *et al.* (2019a), whose dataset is included in the **tvReg** under the name of `FF5F`. The TVFF5F model is

$$\begin{aligned} R_{it} - RF_{it} =& a_{it} + b_{it}\ (RM_{it} - RF_{it}) + s_{it}\ SMB_{it} + h_{it}\ HML_{it} \\ & + r_{it}\ RMW_{it} + c_{it}\ CMA_{it} + u_{it}, \end{aligned} \tag{23}$$

where R_{it} refers to the price return of the asset of certain portfolio for market i at time t, RF_t is the risk free return rate, RM_t represents the return of the market portfolio, SMB_t stands for "small minus big" measuring the historic excess returns of small caps over big caps, and HML_t stands for "high minus low" and measures the historic excess returns of value stocks over growth stocks. The recently added factors are RMW_t and CMA_t. The former is the profitability factor and it is computed as the difference between the returns of firms with robust and weak operating profitability. The latter accounts for the investment capabilities of the company and it is calculated as the difference between the returns of firms that invest conservatively and firms that invest aggressively. Finally, the error term structure is

$$E(u_{it}u_{js}) = \begin{cases} \sigma_{iit} = \sigma^2_{it} & i = j, \quad t = s \\ \sigma_{ijt} & i \neq j, \quad t = s \\ 0 & t \neq s. \end{cases}$$

The FF5F dataset has been downloaded from the Kenneth R. French (2016) data library. It contains the five factors from four different international markets: North America (NA) with the United States and Canada; Japan (JP); Europe (EU) consisting of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, the Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, and the United Kingdom; and Asia Pacific (AP) with Australia, New Zealand, Hong Kong and Singapore. For the dependent variable, the excess returns of portfolios formed on size and book-to-market have been selected. The period runs from July 1990 to August 2016 and it has a monthly frequency. The data contains the Small/Low, Small/High, Big/Low and Big/High portfolios. The factors in the TVFF5F model explain the variation in returns well if the intercept is statistically zero. The lines of code below illustrate how to fit a TVSURE to the Small/Low portfolio.

```
> data("FF5F")
> eqNA <- NA.SMALL.LoBM - NA.RF ~ NA.Mkt.RF + NA.SMB + NA.HML + NA.RMW +
+     NA.CMA
> eqJP <- JP.SMALL.LoBM - JP.RF ~ JP.Mkt.RF + JP.SMB + JP.HML + JP.RMW +
+     JP.CMA
> eqAP <- AP.SMALL.LoBM - AP.RF ~ AP.Mkt.RF + AP.SMB + AP.HML + AP.RMW +
+     AP.CMA
> eqEU <- EU.SMALL.LoBM - EU.RF ~ EU.Mkt.RF + EU.SMB + EU.HML + EU.RMW +
+     EU.CMA
> system2 <- list(NorthA = eqNA, JP = eqJP, AP = eqAP, EU = eqEU)
> tvFF5F <- tvSURE(system2, data = FF5F, method = "tvFGLS",
+                  bw = c(0.56, 0.27, 0.43, 0.18), bw.cov = 0.12)
```

The coefficient 95% confidence intervals of the intercept for the North American, Japanese, Asia Pacific and European markets are calculated in the code below. They are plotted using the plot method for the class attribute tvsure. The plot statement below produces four independent plots of the first variable in each equation due to argument vars = 1.

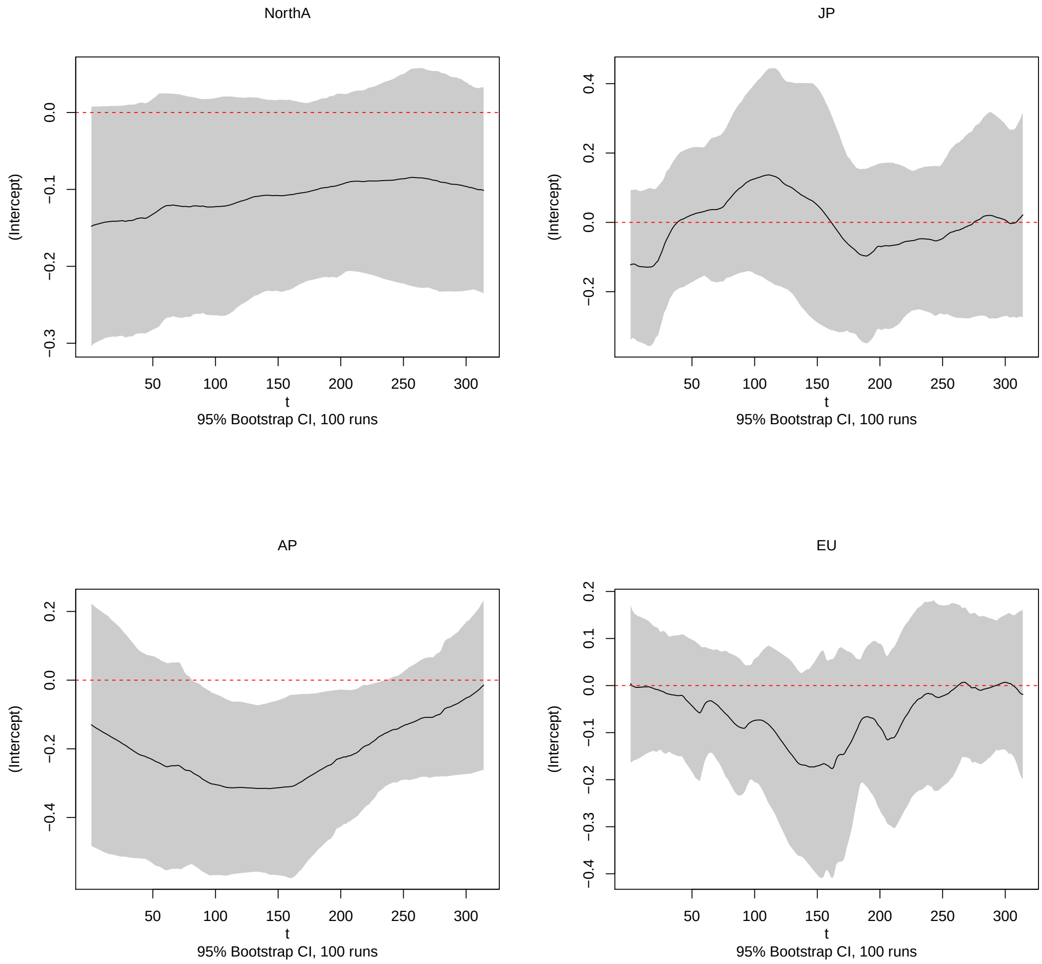
```
> tvFF5F <- confint(tvFF5F)
> plot(tvFF5F, vars = 1)
```

Figure 4 shows that, in the Asia Pacific market, the value of the intercept and its 95% confidence interval are different from zero during the mid period of the sample. This means that during this period, these factors do not explain the Asia Pacific market returns well. On the other hand, the rest of Japanese and European markets have been explained well by these five factors during the whole period with intercepts statistically equal to zero. The North American market intercept is very borderline different from zero during the early period.

The user can also choose to plot the coefficients of different variable(s) and/or equation(s). Plots will be grouped by equation, with a maximum of three variables per plot. The piece of code below show how to plot the second and third variables from the second equation, which results can be seen in Figure 5.

```
> plot(tvFF5F, vars = c(2, 3), eqs = 2)
```

Figure 4: Intercept estimates of a Small/Low portfolio in the four markets (left to right, top to bottom: North America, Japan, Asia Pacific and Europe).



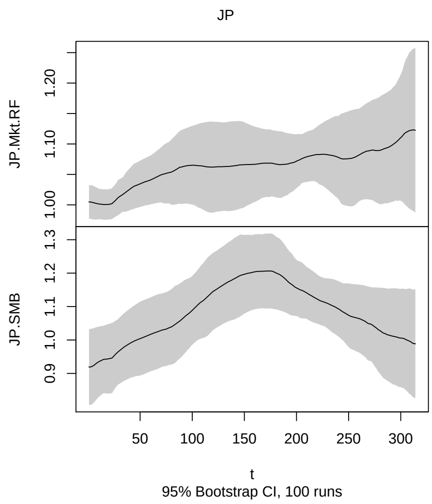
5.3. Standard usage of tvPLM

The `tvPLM` method is inspired by the `plm` method from the package **plm**. It converts `data` into an object of the class attribute `pdata.frame` using argument `index` to define the cross-section and time dimensions. If `index = NULL` (default), the two first columns of `data` define the two dimensions.

The `tvPLM` wraps the `tvRE` and `tvFE` methods to estimate the coefficients of time-varying panel data models.

The user can provide additional optional arguments to modify the default estimation. See section "Standard usage of `tvLM`" for details on arguments `z`, `data`, `bw`, `est` and `tkernel`. Note that the current version only allows one single smoothing random variable, `z`, for all equations. Furthermore, argument `method` defines the estimator used. The possible choices based on package **plm** choices are: `"pooling"` (default), `"random"` and `"within"`.

Figure 5: Coefficient estimates of market return and SMB variables for a Small/Low portfolio in the Japan market.



5.4. Application to health policy

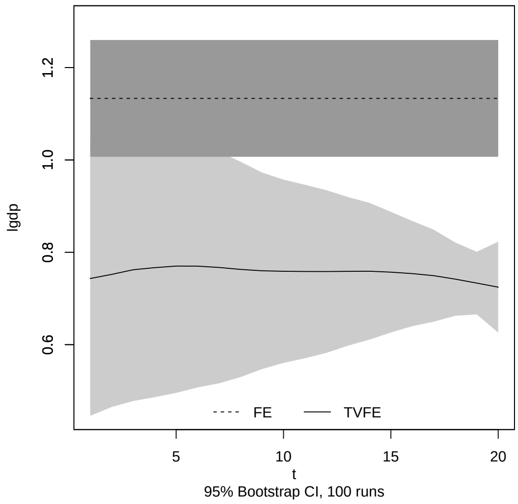
The income elasticity of healthcare expenditure is defined as the percentage change in healthcare expenditure in response to the percentage change in income per capita. If this elasticity is greater than one, then healthcare expenditure grows faster than income, as luxury goods do, and is driven by market forces alone. The heterogeneity of health systems among countries and time periods have motivated the use of models like (14) for example in Gerdtham, Soegaard, Andersson, and Jönsson (1992) who use a FE model. Recently, Casas *et al.* (2019b) have investigated the problem from the time-varying panel models perspective using the TVFE estimation. In addition to the income per capita, measured by the log GDP, the authors use the ratio of population over 65 years old, population under 15 years old and the ratio of public funding of healthcare. The income elasticity estimate with a FE developed in the **plm** is greater than 1, a counter-intuitive result. This issue is resolved using the TVFE developed in the **tvReg**. Find the code below:

```
> data("OECD")
> elast.fe <- plm::plm(lhe ~ lgdp + pop65 + pop14 + public, data = OECD,
+                      index = c("country", "year"), model = "within")
> elast.tvfe <- tvPLM (lhe ~ lgdp + pop65 + pop14 + public, data = OECD,
+                      index = c("country", "year"), method = "within", bw = 0.6)
> elast.fe <- confint(elast.fe)
> elast.tvfe <- confint(elast.tvfe)
> #Comparison plot of the within estimators
> plot(elast.tvfe, vars = 1, ylim = c(0.45, 1.3))
> graphics::par(mfrow = c(1, 1),
+                mar = c(4, 4, 2, 1), oma = c(0, 0, 0, 0))
> x.axis <- 1:elast.tvfe$obs
> par(new = TRUE)
> plot(x.axis, rep(mean(elast.fe[1,]), elast.tvfe$obs), ylim = c(0.45, 1.3),
```

```
+      ylab ="", xlab ="", type = "l", xaxt ="n", lty = 2)
> graphics::polygon(c(rev(x.axis), x.axis),
+                    c(rep(rev(elast.fe[1, 2]), elast.tvfe$obs),
+                      rep(elast.fe[1, 1], elast.tvfe$obs)),
+                    col = "grey60", border = NA, fillOddEven = TRUE)
> lines(x.axis, rep(mean(elast.fe[1,]), elast.tvfe$obs), lty=2)
> legend("bottom", c("FE", "TVFE"), lty = 2:1, col = 1, ncol = 2, bty = "n")
```

Figure 6 show the elasticity estimates using the FE and TVFE estimators. Results from the constant coefficient model are plotted with a white line within a 95% confidence intervals in black, and the time-varying estimates are plotted with a black line within a 95% confidence intervals in grey. The constant coefficients model suggest that healthcare is a luxury good, while the time-varying coefficients models suggest it is a value under 0.8. The confidence intervals decrease around the global financial crisis (value 15 corresponds to year 2009), showing that the variability in the elasticity among countries decreased during that period.

Figure 6: Comparison of income elasticity of healthcare expenditure in OECD countries. The continuous line with light bands corresponds to the TVFE estimates and their 95% confidence intervals. The dash line with darker bands corresponds to the FE estimates and their 95% confidence bands.



5.5. Standard usage of tvVAR and tvIRF

A TVVAR(p) model is a system of TVAR(p) equations. The dependent variable, `y`, is of the class attribute `matrix` or `data.frame` with as many columns as equations. Regressors are the same for all equations and they contain an intercept if the argument `type` is set to `"const"`; lagged values of `y`; and other exogenous variables in `exogen`. Econometrically, the `tvOLS` method is called to calculate the estimates for each equation independently using one bandwidth per equation. The user can enter one value in `bw`, meaning that all equations will be estimated with the same bandwidth; or a vector of bandwidths, one for each equation. If `bw = NULL`, the default, one bandwidth for each equation is automatically selected by leave-one-out cross-validation.

As in the other models, the kernel and the estimation methodology can be chosen with arguments `tkernel` and `est`. The `tvVAR` returns a `list` of the class attribute `tvvar`, which can be used to estimate the TVIRF model with the function `tvIRF`. The smoothing variable is τ by default, but another random variable can be used in argument `z`.

The example below uses the macroeconomic dataset `Canada` from the package **vars**. It consists of quarterly Canadian macrodata indicators of employment (`e`), labour productivity (code-prod), unemployment rate (`U`) and real wages (`rw`). Model (24) is a TVVAR(2) as an extension of the classical VAR(2).

$$
\begin{aligned}
\mathrm{e}_t =& a_t^1 + \sum_{i=1}^{2} b_{it}^1\ \mathrm{e}_{t-i} + \sum_{i=1}^{2} c_{it}^1\ \mathrm{prod}_{t-i} + \sum_{i=1}^{2} d_{it}^1\ \mathrm{rw}_{t-i} + \sum_{i=1}^{2} f_{it}^1\ \mathrm{U}_{t-i} + u_t^1 \\
\mathrm{prod}_t =& a_t^2 + \sum_{i=1}^{2} b_{it}^2\ \mathrm{e}_{t-i} + \sum_{i=1}^{2} c_{it}^2\ \mathrm{prod}_{t-i} + \sum_{i=1}^{2} d_{it}^2\ \mathrm{rw}_{t-i} + \sum_{i=1}^{2} f_{it}^2\ \mathrm{U}_{t-i} + u_t^2 \\
\mathrm{rw}_t =& a_t^3 + \sum_{i=1}^{2} b_{it}^3\ \mathrm{e}_{t-i} + \sum_{i=1}^{2} c_{it}^3\ \mathrm{prod}_{t-i} + \sum_{i=1}^{2} d_{it}^3\ \mathrm{rw}_{t-i} + \sum_{i=1}^{2} f_{it}^3\ \mathrm{U}_{t-i} + u_t^3 \\
\mathrm{U}_t =& a_t^4 + \sum_{i=1}^{2} b_{it}^4\ \mathrm{e}_{t-i} + \sum_{i=1}^{2} c_{it}^4\ \mathrm{prod}_{t-i} + \sum_{i=1}^{2} d_{it}^4\ \mathrm{rw}_{t-i} + \sum_{i=1}^{2} f_{it}^4\ \mathrm{U}_{t-i} + u_t^4 \qquad (24)
\end{aligned}
$$

The code below fits a VAR(2) model using the function `VAR` from the package **vars** and a TVVAR(2) with function `tvVAR`. Both models contain an intercept and no exogenous variables. Bandwidths of `tvVAR.fit` are pretty large with values 1.49, 2.29, 20 and 1.19 for each equation in (24), respectively.

```
> data(Canada, package = "vars")
> VAR.fit <- vars::VAR(Canada, p = 2, type = "const")
> tvVAR.fit <- tvVAR(Canada, p = 2, type = "const")

Calculating regression bandwidths... bandwidth(s)  0.8170732 2.287034 20 0.7195122

> print(tvVAR.fit$bw)

     bw.e     bw.prod       bw.rw        bw.U
0.8170732  2.2870339 19.9999995  0.7195122
```

The user can provide additional optional arguments to modify the default estimation. See section 3.2 to understand the usage of the arguments `p`, `z`, `bw`, `type`, `exogen`, `est`, `tkernel` and `singular.ok`. The variance-covariance matrix from the residuals of a TVVAR(p) can be used to calculate the orthogonal TVIRF.

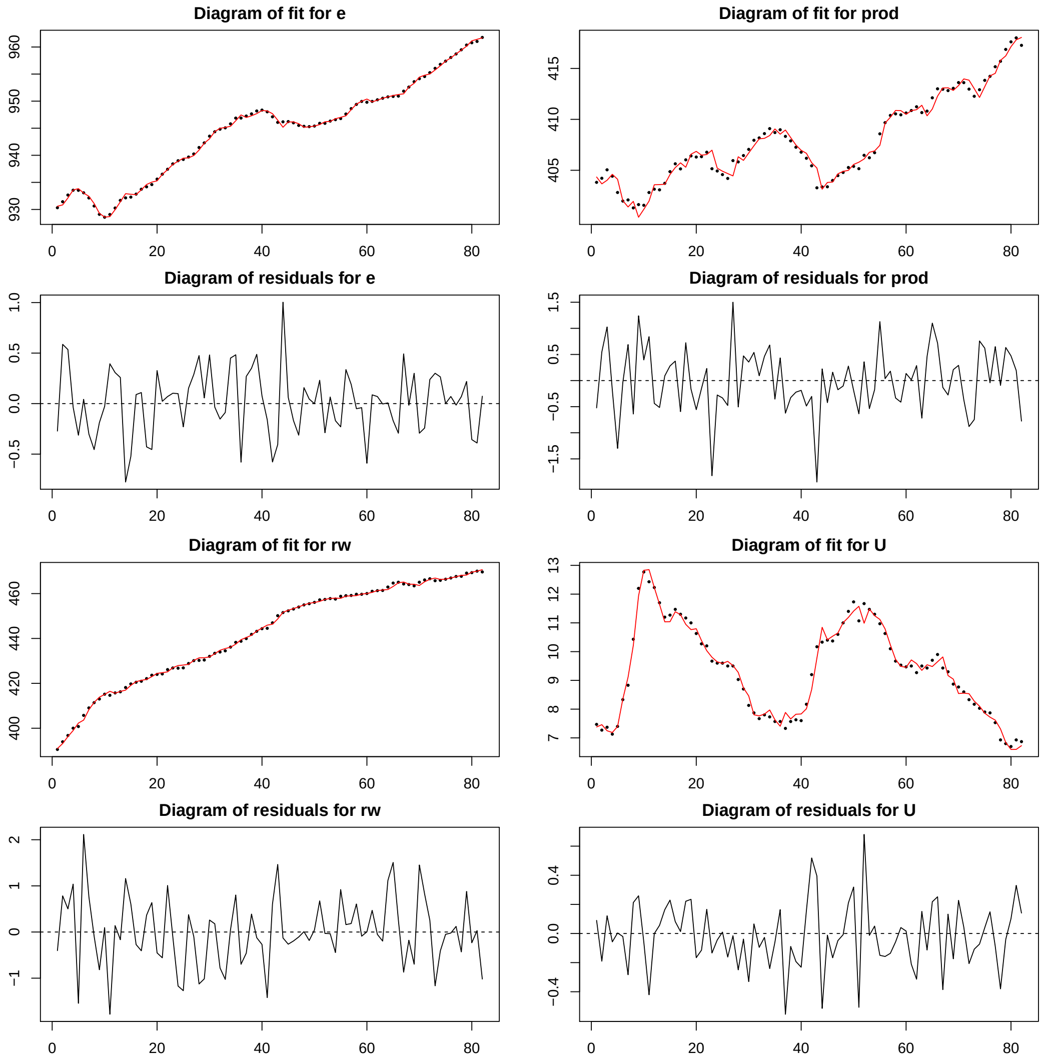
Note that the current version only allows one single smoothing variable for every variable coefficients. So the argument `z` must be either `NULL` if the coefficients are functions of time or a vector if they are function of another random variable.

The `plot` method for object of class attribute `tvvar` displays as many plots as equations, each plot with the fitted and residuals values as it is shown in Figure 7.

```
> plot(tvVAR.fit)
```

Figure 7 show that the residuals of all equations have a mean close to zero and the fitted values are fitting the data points closely.

Figure 7: Plot return of object `tvVAR.fit` for the employment (`e`) and unemployment rate (`U`) equations.



The `summary` method displays a summary of all coefficient values over the whole time period, the value of the bandwidth(s), and a measure of goodness-of-fit, pseudo-R^2. The latter is only printed for the class attributes `tvlm`, `tvar`, `tvplm`, `tvsure` and `tvvar` and it is calculated with the classical equation,

$$R^2 = 1 - \frac{\sum_{t=1}^{T}(y_t - \hat{y})^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2},$$

where y_t is the dependent variable, $\bar{y}$ is its mean and $\hat{y}_t$ are the fitted values. For multiple equation models, one pseudo-R^2 is calculated for each equation.

As an example, the output of the summary from tvVAR.fit is shown in the chunk below.

```
> summary(tvVAR.fit)

Call:
tvVAR(y = Canada, p = 2, type = "const")

Summary of tvVAR for equation e:
==================================
         e.l1 prod.l1    rw.l1   U.l1    e.l2  prod.l2     rw.l2    U.l2
Min.    1.447  0.1254 -0.10348 0.2364 -0.4970 -0.17044 -0.069437 0.04553
1st Qu. 1.591  0.1509 -0.07113 0.2579 -0.4916 -0.13574 -0.021167 0.07868
Median  1.634  0.1733 -0.05510 0.2815 -0.4705 -0.10667 -0.008469 0.11816
Mean    1.617  0.1860 -0.05827 0.3381 -0.4293 -0.10669 -0.018390 0.15320
3rd Qu. 1.663  0.2079 -0.04175 0.3603 -0.3862 -0.08083 -0.003808 0.19082
Max.    1.703  0.2844 -0.02375 0.6275 -0.2452 -0.01535 -0.002961 0.37671
        (Intercept)
Min.        -411.83
1st Qu.     -206.96
Median      -138.56
Mean        -179.68
3rd Qu.     -108.87
Max.         -88.54

Bandwidth:  0.8171
Pseudo R-squared:  0.9987


Summary of tvVAR for equation prod:
=====================================
           e.l1 prod.l1   rw.l1    U.l1   e.l2 prod.l2   rw.l2  U.l2
Min.    -0.1832   1.146 0.04797 -0.4971 0.3806 -0.1767 -0.1216 1.015
1st Qu. -0.1795   1.148 0.04975 -0.4918 0.3842 -0.1740 -0.1201 1.019
Median  -0.1758   1.150 0.05140 -0.4873 0.3873 -0.1715 -0.1187 1.023
Mean    -0.1757   1.150 0.05140 -0.4876 0.3871 -0.1715 -0.1186 1.023
3rd Qu. -0.1721   1.152 0.05306 -0.4832 0.3901 -0.1690 -0.1172 1.028
Max.    -0.1678   1.154 0.05484 -0.4793 0.3928 -0.1664 -0.1155 1.033
        (Intercept)
Min.         -166.7
1st Qu.      -166.4
Median       -166.1
Mean         -166.0
3rd Qu.      -165.7
Max.         -165.1

Bandwidth:  2.287
Pseudo R-squared:  0.9788
```

```
Summary of tvVAR for equation rw:
==================================
           e.l1  prod.l1  rw.l1    U.l1   e.l2   prod.l2   rw.l2    U.l2
Min.    -0.2690 -0.08108 0.8954 0.01201 0.3678 -0.005160 0.05258 -0.1279
1st Qu. -0.2689 -0.08107 0.8955 0.01212 0.3678 -0.005160 0.05261 -0.1278
Median  -0.2689 -0.08107 0.8955 0.01223 0.3679 -0.005160 0.05265 -0.1277
Mean    -0.2689 -0.08107 0.8955 0.01223 0.3679 -0.005160 0.05265 -0.1277
3rd Qu. -0.2688 -0.08106 0.8955 0.01234 0.3680 -0.005159 0.05268 -0.1276
Max.    -0.2687 -0.08105 0.8956 0.01245 0.3680 -0.005159 0.05272 -0.1275
        (Intercept)
Min.         -33.26
1st Qu.      -33.24
Median       -33.22
Mean         -33.22
3rd Qu.      -33.21
Max.         -33.19

Bandwidth:  20
Pseudo R-squared:  0.9989


Summary of tvVAR for equation U:
=================================
           e.l1  prod.l1   rw.l1   U.l1    e.l2   prod.l2   rw.l2     U.l2
Min.    -0.6675 -0.16927 0.01675 0.1608 0.07907 -0.007052 0.01126 -0.41430
1st Qu. -0.6098 -0.14253 0.01988 0.3802 0.21754  0.065928 0.02222 -0.25436
Median  -0.5787 -0.10036 0.02744 0.5759 0.35654  0.067302 0.03804 -0.04808
Mean    -0.5600 -0.11570 0.03355 0.4970 0.30874  0.066092 0.05780 -0.11284
3rd Qu. -0.5300 -0.09412 0.04661 0.6266 0.41519  0.081135 0.09267  0.02115
Max.    -0.4059 -0.09170 0.06166 0.6530 0.44166  0.088026 0.14656  0.07134
        (Intercept)
Min.          81.28
1st Qu.      108.40
Median       149.11
Mean         223.05
3rd Qu.      341.30
Max.         505.89

Bandwidth:  0.7195
Pseudo R-squared:  0.9807
```

The function `tvIRF` estimates the TVIRF with main argument, `x`, which is an object of class attribute `tvvar` returned by the function `tvVAR`. The user can provide additional optional arguments to modify the default estimation as explained below.

Impulse and response variables
The user has the option to pick a subset of impulse variables and/or response variables using arguments `impulse` and `response`.

Horizon
The horizon of the TVIRF coefficients can be chosen by the user with argument `n.ahead`, the default is 10.

Orthogonal TVIRF
The orthogonalised impulse response function is computed by default (`ortho = TRUE`). In the orthogonal case, the estimation of the variance-covariance matrix of the errors is estimated as time-varying (`ortho.cov = "tv"`) by default (see 6 for theoretical details). Note that the user can enter a value of the bandwidth for the variance-covariance matrix estimation in `bw.cov`. It is possible to use a constant variance-covariance matrix by setting `ortho.cov = "const"`.

Cumulative TVIRF
If the user desires to obtain the cumulative TVIRF values, then argument `cumulative` must be set to `TRUE`.

Following the previous example, the lines of code below estimate the IRF using the package **vars** and the TVIRF of the previous TVVAR model. In both cases, the orthogonal (default) impulse response function is returned. Furthermore, `tvIRF2.fit` illustrates the use of argument `cumulative`.

```
> IRF.fit <- vars::irf(VAR.fit)
> tvIRF.fit <- tvIRF(tvVAR.fit)
> tvIRF2.fit <- tvIRF(tvVAR.fit, cumulative = TRUE)
```

5.6. Application to monetary policy

The assessment and forecast of the effects of monetary policy on macroeconomic variables, such as inflation, economic output and employment is commonly modelled using the econometric framework of VAR and interpreted by the IRF. In recent years, scholars of macroeconometrics have searched intensely for a way to include time variation in the coefficients and covariance matrix of the VAR model. The reason for this is that the macroeconomic climate evolves over time and effects of monetary policy must be identified locally rather than globally. In the Bayesian framework, Primiceri (2005) used the Carter and Kohn (1994) algorithm to fit the TVP-VAR to this monetary policy problem. Results of the latter can be replicated with the functions in the package `bvarsv` and compared with results in the **tvReg**. The mathematical model follows the expression of a TVVAR(4):

$$\text{inf}_t = a_t^1 + \sum_{i=1}^{4} b_{it}^1 \ \text{inf}_{t-i} + \sum_{i=1}^{4} c_{it}^1 \ \text{une}_{t-i} + \sum_{i=1}^{4} d_{it}^1 \ \text{tbi}_{t-i} + u_t^1$$

$$\text{une}_t = a_t^2 + \sum_{i=1}^{4} b_{it}^2 \ \text{inf}_{t-i} + \sum_{i=1}^{4} c_{it}^2 \ \text{une}_{t-i} + \sum_{i=1}^{4} d_{it}^2 \ \text{tbi}_{t-i} + u_t^2$$

$$\text{tbi}_t = a_t^3 + \sum_{i=1}^{4} b_{it}^3 \,\text{inf}_{t-i} + \sum_{i=1}^{4} c_{it}^3 \,\text{une}_{t-i} + \sum_{i=1}^{4} d_{it}^3 \,\text{tbi}_{t-i} + u_t^3.$$

Central banks commonly regulate the money supply by changing the interest rates to keep a stable inflation growth. The R code below uses macroeconomic data from the United States, exactly the one used in Primiceri (2005), with the following three variables: inflation rate (`inf`), unemployment rate (`une`) and the three months treasury bill interest rate (`tbi`). For illustration, a VAR(4) model is estimated using the function `VAR` from the package **vars**, a TVVAR(4) model is estimated using the function `tvVAR` from the package **tvReg** and a TVP-VAR(4) model is estimated using the function `bvar.sv.tvp` from the package `bvarsv`. Furthermore, their corresponding impulse response functions with horizon 20 are calculated to forecast how the inflation responds to a positive shock in interest rates.

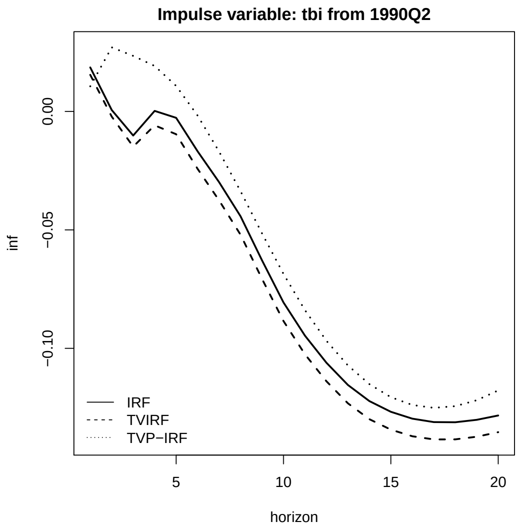
```
> data(usmacro, package = "bvarsv")
> VAR.usmacro <- vars::VAR(usmacro, p = 4, type = "const")
> IRF.usmacro <- vars::irf(VAR.usmacro, impulse = "tbi", response ="inf",
+                           n.ahead = 20)
> tvVAR.usmacro <- tvVAR(usmacro, p = 4, bw = c(1.14, 20, 20),
+                         type = "const")
> tvIRF.usmacro <- tvIRF(tvVAR.usmacro, impulse = "tbi", response="inf",
+                         n.ahead = 20)
> tvpVAR.usmacro <- bvarsv::bvar.sv.tvp(usmacro, p = 4, pdrift = TRUE,
+                                       nrep = 1000, nburn = 1000,
+                                       save.parameters = TRUE)
> tvpIRF.usmacro <- bvarsv::impulse.responses(tvpVAR.usmacro,
+                                             impulse.variable = 3,
+                                             response.variable = 1,
+                                             draw.plot = FALSE)
```

A comparison of impulse response functions from the three estimations is plotted in Figure 8, whose R code is shown below. Note that while the IRF is constant for all time points, the TVIRF and TVP-IRF may change over time.

```
> irf1 <- IRF.usmacro$irf[["tbi"]]
> irf2 <- tvIRF.usmacro$irf[["tbi"]]
> irf3 <- tvpIRF.usmacro$irf
> par (mar = c(4, 4, 2, 1))
> ylim <- range(irf1, irf2[150,,], irf3[50,])
> plot(1:20, irf1[-1], ylim = ylim, main = "Impulse variable: tbi from 1990Q2",
+      xlab ="horizon", ylab ="inf", type ="l", lwd = 2)
> lines(1:20, irf2[150,,-1], lty = 2, lwd = 2)
> lines(1:20, irf3[50,], lty = 3, lwd = 2)
> legend("bottomleft", c("IRF", "TVIRF", "TVP-IRF"), lty = 1:3, col = 1,
+        bty ="n")
```

Figure 8 displays the IRF for horizons 1 to 20 and the TVIRF and TVP-IRF at time 150 in our dataset, which corresponds to the second quarter of 1990. The IRF and TVIRF follow a similar pattern: a positive shock of one unit in the short-term interest rates (`tbi`) during 1990 results in a slight drop in inflation during the first five or six months and then in a steady decrease until it plateaus one year after for the TVIRF and about six quarters after for the IRF. The TVP-IRF estimates show a continuous inflation drop after the first eight months.

Figure 8: Estimated response of inflation to an increase in interest rates of one unit during 1990Q2.



The `confint` method is not implemented for the class attribute `tvvar` because it is not common to interpret the coefficients of model TVVAR. It is, however, implemented for the class attribute `tvirf`. For illustration, the 90% confidence intervals of object `tvIRF.usmacro` is computed in the code below. In addition, the response of the variable `inf` to a shock of one unit increase in `tbi` is plotted in Figure 9. Remember that the TVIRF model contains one impulse response function for data time record. So, the full plot of TVIRF would have as many lines as the number of rows in the dataset. Instead, the `plot` method displays only one line by default, the mean value of all those impulse response functions and it issues a warning. The user can enter one or several values into argument `obs.index` to plot the IRF at the desire point(s) in time.

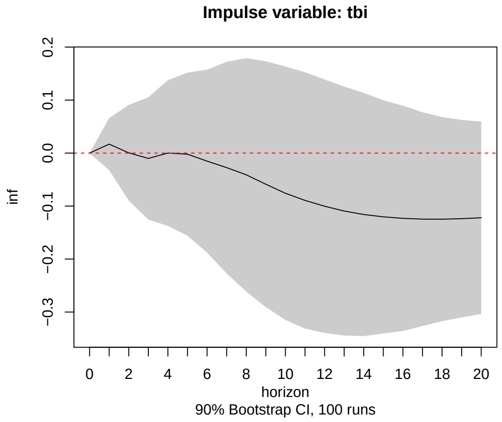
```
> tvIRF.90 <- confint(tvIRF.usmacro, level = 0.90)
> plot(tvIRF.90, impulse = "tbi", response ="inf")
```

Figure 9 shows that the inflation is expected to decrease during the first year (12 horizons) in by an average of 0.1 when the short-term interest rates increase by one unit. Then its value will stabilise. The 90% confidence intervals contain the value zero though, so we cannot assure that this

6. Estimating a time-varying variance-covariance matrix

The time-varying variance-covariance matrix of two or more series is estimated nonparametrically in `tvReg`. Given a random process $y_i = (y_{i1}, \dots, y_{iT})^\top$, such that $\mathsf{E}(y_{it}) = 0$ and

Figure 9: Coefficients of object `tvIRF.90` with a 90% confidence intervals for a shock in employment (e) and a shock in unemployment rate (U).



$\mathsf{E}\left(y_{it}y_{i't'}\right) = \sigma_{ii't}$ if $t = t'$ and zero otherwise. Thus, the variance-covariance matrix is allowed to be time-varying with the following expression for each time t:

$$\Sigma_t = \begin{pmatrix} \sigma_{11t} & \sigma_{12t} & \dots & \sigma_{1Nt} \\ \sigma_{21t} & \sigma_{22t} & \dots & \sigma_{2Nt} \\ \vdots & \ddots & & \vdots \\ \sigma_{N1t} & \sigma_{N2t} & \dots & \sigma_{NNt} \end{pmatrix}$$

Given that the matrix is locally stationary, its local linear estimator is defined by

$$\text{vech}(\tilde{\Sigma}_\tau) = \sum_{t=1}^{T} \text{vech}(y_t^\top y_t) K_h(t-\tau) \frac{s_2 - s_1(\tau - t)}{s_0 s_2 - s_1^2} \tag{25}$$

where $s_j = \sum_{t=1}^{T}(\tau - t)^j K_b(\tau - t)$ for j = 0, 1, 2. As shown previously, $K_b(\cdot)$ is a symmetric kernel function heavily concentrated around the origin, $\tau = t/T$ is the focal point and b is the bandwidth parameter. Note that a single bandwidth is used for all co-movements, which ensures that $\tilde{\Sigma}_\tau$ is positive definite.

The user must be aware that the local linear estimator can return non-positive definite matrices for small samples. Although the local constant estimator, calculated when $s_1 = s_2 = 1$ in (25), does not have as good asymptotic properties in the boundaries as the local linear estimator, it always provides positive definite matrices, which is a desirable property of an estimator of a variance-covariance matrix. Therefore, it is the default estimator in the function `tvCov`.

The function `tvCov` is called by the function `tvIRF` to calculate the orthogonal TVIRF, and by the function `tvSURE` for `method = "tvFGLS"` to estimate the variance-covariance matrix of the error term. The function `tvCov` can generally be used to estimate the time-varying covariance matrix of any two or more series.

6.1. Application to portfolio management

Aslanidis and Casas (2013) consider a portfolio of daily US dollar exchange rates of the Australian dollar (AUS), Swiss franc (CHF), euro (EUR), British pound (GBP), South African rand (RAND), Brazilian real (REALB) and Japanese yen (YEN), over the period from January 6, 1999 until May 7, 2010 (T = 2856 observations). This dataset contains the standarised rates after "devolatilisation"; i.e., after standarising the rates using the GARCH(1,1) estimates of the volatility and it is available in the **tvReg** under the names of CEES. A portfolio consisting of these currencies is well diversified containing some safe haven currencies, active and liquid currencies and currencies that perform well in times of high interest rates. The estimation of the correlation matrix among these currencies is essential for portfolio management. The model is

$$
\begin{aligned}
r_{p,t} =& \omega_t^\top r_t \\
h_{p,t} =& \omega_t^\top H_t \omega_t
\end{aligned}
$$

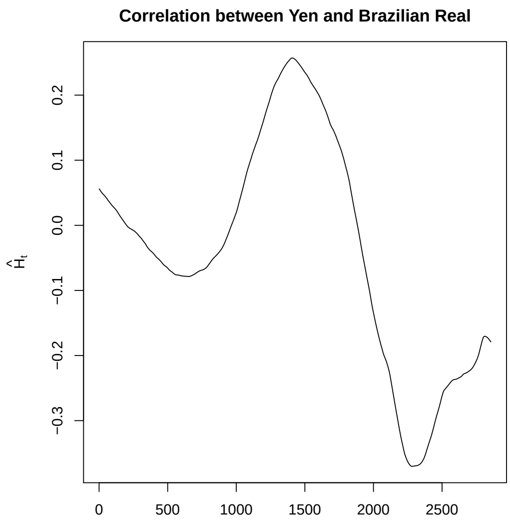
where $r_{p,t}$ and $h_{p,t}$ are the return and variance of the portfolio at time t. Variable ω_t is a vector with the weight of each currency in the portfolio strategy at time t. The portfolio variance-covariance matrix is denoted by H_t, and it may vary with time for a dynamic investment strategy. This matrix can be estimated using the function `tvCov` as shown in the R piece of code below.

```
> data(CEES)
> obs <- nrow(CEES)
> P <- ncol(CEES)
> Ht <- tvCov(CEES, bw = 0.12)
> par(mar = c(2, 5, 3, 1))
> plot(Ht[6, 7, ], type = "l", ylab = expression(hat(H)[t]), xlab = "",
+      main ="Correlation between Yen and Brazilian Real")
```

Figure 10 shows the estimated dynamic correlation between the Japanese yen and the Brazilian real. During the early period of the sample, the correlation is close to zero, it is positive during the mid period and negative number at the end. This makes sense because the Japanse yen is a safe-haven currency, which means that it is especially attractive to investors during times of uncertaintly. On the other hand, the Brazilian real is attractive for investment during the times of financial booming and it tends to pummel down during times of financial crisis. The mid period of our sample corresponds to the mid 2000s during which the financial markets were very profitable, while the final period of our sample includes the global financial crisis and its aftermath.

Value-at-risk, which is denoted by VaR in the financial literature and must not be confused with the VAR, measures the level of financial risk of a portfolio, asset or firm. The VaR of an asset X, with distribution function F_X, at the confidence level α is defined as $\mathrm{VaR}_\alpha = \inf\{x : F_X(x) > \alpha\}$. Commonly, the distribution function of X is assumed to be Gaussian with unknown variance. In a portfolio framework, the variance-covariance matrix is estimated to calculate the VaR of a portfolio together with the portfolio weights (`omega` in the code below). The portfolio weights are the percentage of the total portfolio investment in each asset and can be chosen to be constant or changing over time. In the code below, weights are calculated

Figure 10: Variance-covariance dynamic relationship between the Japanese yen and the Brazilian real).



by minimum variance at each point in time. The estimated VaR of this example portfolio is shown in Figure 11.

```
> e <- rep(1, P)
> alpha <- 0.05
> VaR <- numeric(obs)
> for (t in 1:obs)
+ {
+   omega <- solve(Ht[,,t])%*%e/((t(e)%*%solve(Ht[,,t])%*%e)[1])
+         VaR[t] <- abs(qnorm(alpha))*sqrt(max(t(omega)%*%Ht[,,t]%*%omega,0))
+ }
> plot(VaR, type ="l", xlab = "time", ylab ="VaR", main="VaR of CEES over time")
```

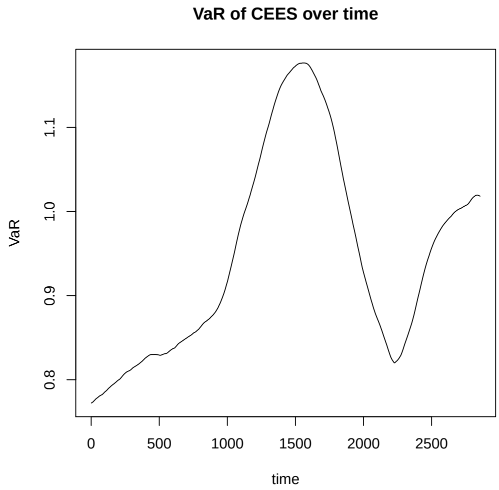
7. Prediction and forecast in time-varying coefficient models

Estimation is a useful tool to understand the patterns and processes hidden in known data. Prediction and forecasting are the mechanisms to extend this understanding to unknown data.

In a classical linear model like (1), the *prediction* of the dependent variable at time $T+h$ is $\hat{y}_{T+h} = x_{T+h}^\top \hat{\beta}$ for $h \geq 1$. All predictors, x_{T+h}, are known.

In autoregressive models, the prediction of future values has a slightly different nature and it is commonly referred to as *forecast*. The predictors (regressors) in the 1-step-ahead forecast are known, but they are unknown for longer horizons. For example, given $y_t = 5 - 0.5y_{t-1} + u_t$ for $t = 1, \ldots, T$, the 1-step-ahead forecast is $\hat{y}_{T+1} = 5 - 0.5y_T$ with known y_T. However, the 2-step-ahead forecast is $\hat{y}_{T+2} = 5 - 0.5\hat{y}_{T+1}$, which requires the previously forecasted $\hat{y}_{T+1}$ as a predictor. This means a greater uncertainty in the forecast error as the forecast horizon increases.

Figure 11: Dynamics of the Value-at-risk of the CEES exchange rates portfolio.



In our time-varying coefficient models, we refer to prediction when the smoothing variable, z_t, is a random variable and to forecast when $z_t = \tau$. Thus, the `predict` and `forecast` methods in **tvReg** are slightly different.

Given a TVLM model (2), with random variable z_t, from which we would like to predict the dependent variable at times $T+1, T+2$ and $T+3$, the focal points z_{T+h} must be known for $h = 1, 2, 3$. Coefficients $\beta(z_{T+h})$ are predicted using the TVOLS estimator in Equation (3) and using the known variables $\{y_t, x_t^\top, z_t\}_{t=1}^T$ as training sample on the focal points z_{T+h}. The predicted values are $\hat{y}_{T+h} = x_{T+h}^\top \hat{\beta}(z_{T+h})$.

If $z_t = \tau$ instead, then the TVLM can be forecasted for times $T+1, T+2$ and $T+3$ one step at the time. The training sample in the 1-step-ahead forecast is $F_T = \{y_t, x_t^\top, z_t\}_{t=1}^T$, where the smoothing variable grid is redefined to $z_t = t/(T+3), t = 1, \ldots, T$. The focal point is $z_{T+1} = (T+1)/(T+3)$. The forecasted value is $\hat{y}_{T+1} = x_{T+1}^\top \hat{\beta}(z_{T+1})$. The training sample in the 2-step-ahead forecast is $F_{T+1} = \{\hat{y}_{T+1}, x_{T+1}^\top, z_{T+1}\} \cup F_T$, which contains the previoulsy forecasted values. The focal point is $z_{T+2} = (T+2)/(T+3)$. Similarly, the training sample in the 3-step-ahead forecast is $F_{T+2} = \{\hat{y}_{T+2}, x_{T+2}^\top, z_{T+2}\} \cup F_{T+1}$, and the focal point is 1. This algorithm is described in Chen *et al.* (2017) for the TVAR model in detail.

7.1. Standard usage of predict and forecast

The `forecast` method is implemented for the class attributes `tvlm, tvar, tvvar` and `tvsure`. As an example, the three days ahead forecast of model `tvHAR`, evaluated in Section 3.3 using the first 2000 values of the dataset `RV`, is provided in the lines of code below. This is a TVAR(1) model with two exogenous variables, `RV_week` and `RV_month`. The argument `newexogen` requires three values of these exogenous variables and variable `n.ahead = 3`.

```
> newexogen <- cbind(RV$RV_week[2001:2003], RV$RV_month[2001:2003])
> forecast(tvHAR, n.ahead = 3, newexogen = newexogen)

[1] 2.200921e-05 2.566854e-05 2.466637e-05
```

The `forecast` method requires the argument `object`. In addition, other arguments are necessary, some of them depending on the class attribute of `object`.

Forecast horizon The argument `n.ahead` is a scalar with the forecast horizon. By default, it is set to 1.

Type of forecast It is possible to run either an increase window forecast (default), when the argument `winsize = 0` or a rolling window forecast with a window size defined in the argument `winsize`.

newx, newdata
These arguments belong to the `forecast` methods for the class attributes `tvlm` and `tvsure`, respectively. They are a `vector`, `data.frame` or `matrix` containing the new values of the regressors in the model. It is not necessary to enter the intercept.

newexogen
This argument appears in the `forecast` method for the class attributes `tvar` and `tvvar` and it must be entered when the initial model contains `exogen` variables. It is a `vector`, `data.frame` or `matrix`.

The `predict` method is implemented for the same four class attributes than the `forecast`. It does not require arguments `n.ahead` and `winsize`, but arguments `newx, newdata` and `newexogen` are defined as in `forecast`. In addition, new values of the smoothing variable must be entered into the argument `newz`. This must be of the class attribute `vector` or `numeric`. The code below, predicts three future values of the `tvHARQ` model fitted above.

```
> newdata <- RV$RV_lag[2001:2003]
> newexogen <- cbind(RV$RV_week[2001:2003], RV$RV_month[2001:2003])
> newz <- RV$RQ_lag_sqrt[2001:2003]
> predict(tvHARQ, newdata, newz, newexogen = newexogen)

[1] 1.739950e-05 2.398599e-05 2.086776e-05
```

The example below shows the usage of the `forecast` and `predict` methods for the class attribute `tvsure`.

The lines of code below forecast three values for model `tvOLS.fit` evaluated in Section 5.1. The method needs a set of new values in the argument `newdata`, which must have the same number of columns as the original dataset.

```
> newKmenta <- data.frame(consump = c(95, 100, 102), price = c(90, 100, 103),
+                         farmPrice = c(70, 95, 103), income = c(82, 94, 115),
+                         trend = c(21:23))
> forecast(tvOLS.fit, newdata = newKmenta, n.ahead = 3)

        demand    supply
[1,]  97.92300  95.32852
[2,]  98.94076 103.48589
[3,] 105.36951 106.26576
```

In case the smoothing variable in the model is a random variable, the `predict` method for the class attribute `tvsure` requires also a new set of values in argument `newz`. The chunk below first fits a TVSURE model, `tvOLS.z.fit`, to the `Kmenta` data with the same system of equations as in the `tvOLS.fit`, but with random variable as the smoothing variable, which is generated as an ARMA(2,2) process. Three values of the dependent variable are predicted with the `predict` method. In addition to new values in the argument `newdata`, it requires a set of new smoothing values in the argument `newz`. It returns the predicted values as a `matrix` with as many columns as equations in the system.

```
> nobs <- nrow (Kmenta)
> smoothing <- arima.sim(n = nobs + 3, list(ar = c(0.8897, -0.4858),
+                                           ma = c(-0.2279, 0.2488)),
+                        sd = sqrt(0.1796))
> smoothing <- as.numeric(smoothing)
> tvOLS.z.fit <- tvSURE(system, data = Kmenta,  z = smoothing[1:nobs])

Calculating regression bandwidth...

> newSmoothing <- tail(smoothing, 3)
> predict(tvOLS.z.fit, newdata = newKmenta, newz = newSmoothing)

        demand    supply
[1,]  98.86607  95.28646
[2,]  99.71995 103.34590
[3,] 105.79936 106.05852
```

The `forecast` and `predict` methods for the rest of the class attributes in the package follow similar patterns, and further examples can be found in the documentation of the `tvReg`.

8. Recap and further development

The **tvReg** functions and methods to estimate the coefficients of six common single-equation and multi-equation linear models with time-varying coefficients, which have seen the object of study in the nonparametric literature in the last two decades. The objective of this package is to allow empirical researchers to use this type of models, models that allow a great deal of flexibility but that are not easily programmed with existing computing tools. The `confint`, `fitted`, `predict`, `plot`, `print`, `resid` and `summary` methods are developed for the five class attributes in the `tvReg` and will allow the user to conveniently produce their research output. In any case, the user is able to produce customised plots and summaries from the returns of the functions, whose elements are accessible in the same manner as other R `list` objects.

Confidence intervals in this package are calculated with bootstrap and are the main tool for statistical inference in the **tvReg** at the moment. The addition of statistical tests is left for future development.

Acknowledgements

We thank the unknown referee for the constructive suggestions and comments on an earlier version of this paper. The first author acknowledges financial support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 657182).

References

Andersen TG, Bollerslev T (1998). "Answering the skeptics: Yes, standard volatility models do provide accurate forecasts." *International Economic Review*, **39**, 885–905.

Aslanidis N, Casas I (2013). "Nonparametric correlation models for portfolio allocation." *Journal of Banking & Finance*, **37**, 2268–2283.

Bollerslev T, Patton AJ, Quaedvlieg R (2016). "Exploiting the errors: A simple approach for improved volatility forecasting." *Journal of Econometrics*, **192**(1), 1–18.

Bollerslev T, Tauchen G, Zhou H (2009). "Expected Stock Returns and Variance Risk Premia." *The Review of Financial Studies*, **22**(11), 44–63.

Cai Z (2007). "Trending time-varying coefficient time series with serially correlated errors." *Journal of Econometrics*, **136**, 163–188.

Cai Z, Fan J, Yao Q (2000). "Functional-Coefficient Regression Models for Nonlinear Time Series." *Journal of the American Statistical Association*, **95**, 941–956.

Cai Z, Li Q, Park JY (2009). "Functional-coefficient models for nonstationary time series data." *Journal of Econometrics*, **148**, 101–113.

Carter CK, Kohn R (1994). "On Gibbs Sampling for State Space Models." *Biometrika*, **81**, 541–553.

Casas I, Ferreira E, Orbe S (2019a). "Time-Varying Coefficient Estimation in SURE Models. Application to Portfolio Management*." *Journal of Financial Econometrics*. Nbz010, https://academic.oup.com/jfec/advance-article-pdf/doi/10.1093/jjfinec/nbz010/28201109/nbz010.pdf.

Casas I, Gao J, Peng B, Xie S (2019b). "Time-Varying Income Elasticities of Healthcare Expenditure for the OECD and Eurozone." *Monash Econometrics and Business Statistics Working Papers 28/19*, Monash University, Department of Econometrics and Business Statistics. URL https://ideas.repec.org/p/msh/ebswps/2019-28.html.

Casas I, Mao X, Veiga H (2018). "Reexamining financial and economic predictability with new estimators of realized variance and variance risk premium." *Working papers*, CREATES, Aarhus University. URL http://econ.au.dk/fileadmin/site_files/filer_oekonomi/Working_Papers/CREATES/2018/rp18_10.pdf.

Chang Y, Martinez-Chombo E (2003). "Electricity Demand Analysis Using Cointegration and Error-Correction Models with Time Varying Parameters: The Mexican Case." *Working papers*, Rice University, Department of Economics. URL https://ideas.repec.org/p/ecl/riceco/2003-08.html.

Chen R, Liu LM (2001). "Functional Coefficient Autoregressive Models: Estimation and Tests of Hypotheses." *Journal of Time Series Analysis*, **22**(2), 151–173.

Chen R, Tsay R (1993). "Functional-coefficient autoregressive models." *Journal of the American Statistical Association*, **88**, 298–308.

Chen XB, Gao J, Li D, Silvapulle P (2017). "Nonparametric Estimation and Forecasting for Time-Varying Coefficient Realized Volatility Models." *Journal of Business & Economic Statistics*, **online**, 1–13.

Cogley T, Sargent TJ (2005). "Drift and Volatilities: Monetary Policies and Outcomes in the Post WWII U.S." *Review of Economic Dynamics*, **8**, 262–302.

Corsi F (2009). "A Simple Approximate Long-Memory Model of Realized Volatility." *Journal of Financial Econometrics*, **7**(2), 174–196.

Croissant Y, Millo G (2008). "Panel Data Econometrics in R: The plm Package." *Journal of Statistical Software*, **27**(2), 1–43. doi:10.18637/jss.v027.i02.

Croissant Y, Millo G (2018). *Panel Data Econometrics with R: the plm package*. Wiley.

Dahlhaus R (1997). "Fitting time series models to nonstationary processes." *Annals of Statistics*, **25**, 1–37.

Das M (2005). "Instrumental variables estimators of nonparametric models with discrete endogenous regressors." *Journal of Econometrics*, **124**, 335–361.

Dong C Jiti Gao J, Peng B (2018). "Varying-coefficient panel data models with partially observed factor structure." *Monash econometrics and business statistics working papers*, Monash University, Department of Econometrics and Business Statistics. URL https://ideas.repec.org/p/msh/ebswps/2018-1.html.

Dong, C Jiti Gao, J and Peng, B (2015). "Semiparametric single-index panel data models with cross-sectional dependence." *Journal of Econometrics*, **188**, 301–312.

Ellahie A, Ricco G (2017). "Government purchases reloaded: Informational insufficiency and heterogeneity in fiscal VARs." *Journal of Monetary Economics*, **90**, 13–27. ISSN 0304-3932. doi:https://doi.org/10.1016/j.jmoneco.2017.06.002. URL http://www.sciencedirect.com/science/article/pii/S0304393217300624.

Fama E, French K (1993). "Common risk factors in the returns on stocks and bonds." *Journal of Financial Economics*, **33**, 3–56.

Fama EF, French KR (2015). "A five-factor asset pricing model." *Journal of Financial Economics*, **116**, 1–22.

Fama EF, French KR (2017). "International tests of a five-factor asset pricing model." *Journal of Financial Economics*, **123**, 441–463.

Fan J, Gijbels I (1996). *Local Polynomial Modeling and Its Applications*. hapman and Hall, London.

French KR (2016). "Data Library." http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html. Accessed: 2016-09-01.

Gao J, Phillips P (2013). "Functional coefficient nonstationary regression." *Cowles Foundation Discussion Papers 1911*, Cowles Foundation for Research in Economics, Yale University. URL https://ideas.repec.org/p/cwl/cwldpp/1911.html.

Gerdtham UG, Soegaard J, Andersson F, Jönsson B (1992). "An Econometric Analysis of Health Care Expenditure: A Cross-Section Study of the OECD Countries." *Journal of Health Economics*, **11**, 63–84.

Haslbeck JM, Waldorp LJ (2016). "mgm: Structure estimation for time-varying mixed graphical models in high-dimensional data." *arXiv preprint:1510.06871v2*. URL http://arxiv.org/abs/1510.06871v2.

Hastie T, Tibshirani R (1993). "Varying-Coefficient Models." *Journal of the Royal Statistical Society. Series B (Methodological)*, **55**, 757–796.

Henderson DJ, Kumbhakar SC, Li Q, Parmeter CF (2015). "Smooth coefficient estimation of a seemingly unrelated regression." *Journal of Econometrics*, **189**, 148–162.

Henningsen A, Hamann JD (2007). "systemfit: A Package for Estimating Systems of Simultaneous Equations in R." *Journal of Statistical Software*, **23**(4), 1–40. URL http://www.jstatsoft.org/v23/i04/.

Holmes EE, Ward EJ, Wills K (2012). "MARSS: Multivariate autoregressive state-space models for analyzing time-series data." *The R Journal*, **4**(1), 30.

Iwata S, Wu S (2006). "Estimating monetary policy effects when interest rates are close to zero." *Journal of Monetary Economics*, **53**, 1395–1408.

Kapetanios G, Marcellino M, Venditti F (2017). "Large time-varying parameter VARs: a non-parametric approach." *Temi di discussione (economic working papers)*, Bank of Italy, Economic Research and International Relations Area. URL https://ideas.repec.org/p/bdi/wptemi/td_1122_17.html.

Kapetanios G, Mumtaz H, Stevens I, Theodoridis K (2012). "Assessing the Economy-wide effects of quantitative easing." *The Economic Journal*, **122**, 316–347.

Kmenta J (1986). *Elements of Econometrics*. 2nd edition. New York: Macmillan.

Krueger F (2015). *bvarsv: Bayesian analysis of a vector autoregressive model with stochastic volatility and time-varying parameters*. R package version 1.1, URL https://CRAN.R-project.org/package=bvarsv.

Lütkepohl H (2005). *New Introduction to Multiple Time Series Analysis*. Springer.

Mammen E (1993). "Bootstrap and Wild Bootstrap for High Dimensional Linear Models." *The Annals of Statistics*, **21**, 255–285.

Orbe S, Ferreira E, Rodriguez-Poo J (2005). "Nonparametric estimation of time varying parameters under shape restrictions." *Journal of Econometrics*, **126**, 53–77.

Pfaff B (2008). "VAR, SVAR and SVEC models: Implementation within R package vars." *Journal of Statistical Software, Articles*, **27**(4), 1–32. URL https://www.jstatsoft.org/v027/i04.

Primiceri GE (2005). "Time varying structural vector autoregressions and monetary policy." *Review of Economic Studies*, **72**, 821–852.

Robinson P (1989). "Nonparametric estimation of time-varying parameters." In P Hackl (ed.), *Statistical Analysis and Forecasting of Economic Structural Change*. Springer, Berlin.

Sims CA (1980). "Macroeconomics and reality." *Econometrica*, **48**, 1–48.

Store HSMD (2017). "Intraday Products." http://download-stock-data.webs.com/. Accessed: 2017-11-01.

Sun Y, Cai Z, Li Q (2013). "Semiparametric functional coefficient models with integrated covariates." *Econometric Theory*, **29**, 659–672.

Sun Y, Carroll R, Li D (2009). "Semiparametric estimation of fixed-effects panel data varying coefficient models." *Advances in Econometrics*, **25**, 101–129.

Vargas-Silva C (2008). "Monetary policy and the US housing market: A VAR analysis imposing sign restrictions." *Journal of Macroeconomics*, **30**, 977–990.

Wooldridge JM (2010). *Econometric Analysis of Cross Section and Panel Data*. MIT Press Books, 2nd edition. The MIT Press.

Xiao Z (2009). "Functional-coefficient cointegration models." *Journal of Econometrics*, **152**, 81–92.

Zellner A (1962). "An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias." *Journal of the American Statistical Association*, **57**, 348–368.

Affiliation:

Isabel Casas
Department of Business and Economics
University of Southern Denmark
Campusvej 55, 5230 Odense M, Denmark
Email: casasis@gmail.com
URL: www.icasasweb.com

Rubén Fernández-Casal
Department of Mathematics
University of A Coruña