BVAR: Bayesian Vector Autoregressions with Hierarchical Prior Selection in R

Kuschnig, Nikolas; Vashold, Lukas

DOI:
10.57938/e693f953-029b-4281-9867-4ac97a8777dc

Published: 01/01/2019

Document Version:
Publisher's PDF, also known as Version of record

Document License:
Unspecified

Department of Economics
Working Paper No. 296

BVAR: Bayesian Vector Autoregressions with Hierarchical Prior Selection in R

Nikolas Kuschnig
Lukas Vashold

October 2019

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

BVAR: Bayesian Vector Autoregressions with Hierarchical Prior Selection in R

Nikolas Kuschnig
WU Vienna University
of Economics and Business

Lukas Vashold
WU Vienna University
of Economics and Business

Abstract

Vector autoregression (VAR) models are widely used for multivariate time series analysis in macroeconomics, finance, and related fields. Bayesian methods are employed to deal with their dense parameterization, imposing structure on model coefficients via prior information. The optimal choice of the degree of informativeness implied by these priors is subject of much debate and can be approached via hierarchical modeling. This paper introduces **BVAR**, an R package dedicated to the estimation of Bayesian VAR models with hierarchical prior selection. It implements functionalities and options that permit addressing a wide range of research problems, while retaining an easy-to-use and transparent interface. Features include structural analysis of impulse responses, forecasts, the most commonly used conjugate priors, as well as a framework for defining custom dummy-observation priors. **BVAR** makes Bayesian VAR models user-friendly and provides an accessible reference implementation.

Keywords: vector autoregression (VAR), multivariate, time series, macroeconomics, econometrics, structural analysis, hierarchical model, forecast, impulse respose function, identification, Minnesota prior, FRED-MD, dataset, R package, free software, Bayesian inference.

1. Introduction

Vector autoregression (VAR) models, popularized by Sims (1980), have become a staple of empirical macroeconomic research (Kilian and Lütkepohl 2017). They are widely used for multivariate time series analysis and have been applied to evaluate DSGE models (Del Negro, Schorfheide, Smets, and Wouters 2007), investigate the effects of monetary policy (Bernanke, Boivin, and Eliasz 2005; Sims and Zha 2006), and conduct forecasting exercises (Litterman 1986; Koop 2013). The large number of parameters and limited temporal availability of macroeconomic datasets often lead to over-parameterization problems (Koop and Korobilis 2010) that can be mitigated by introducing prior information within a Bayesian approach. Informative priors are used to impose additional structure on the model and shrink it towards proven benchmarks. The result are models with reduced parameter uncertainty and significantly enhanced out-of-sample forecasting performance (Koop 2013). However, the specific choice and parameterization of these shrinkage priors pose a challenge that remains the fulcrum of discussion and critique. A number of heuristics for prior selection have been proposed in the literature. Giannone, Lenza, and Primiceri (2015) tackle this problem by setting prior informativeness in a data-based fashion, in the spirit of hierarchical modeling. Their flexible

approach alleviates the subjectivity of setting prior parameters and explicitly acknowledges uncertainty surrounding these choices. The conjugate setup allows for efficient estimation and has been shown to perform remarkably well in common analyses (see Miranda-Agrippino and Rey 2015; Baumeister and Kilian 2016).

Amidst the rise of Markov chain Monte Carlo (MCMC) methods, Bayesian statistical software has evolved rapidly. Established software provides flexible and extensible tools for Bayesian inference, which are available cross-platform. This includes **BUGS** (Lunn, Thomas, Best, and Spiegelhalter 2000; Lunn, Spiegelhalter, Thomas, and Best 2009) and **JAGS** (Plummer 2003), which build on the Gibbs sampler, **Stan** (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Riddell 2017), which builds on the Hamiltonian Monte Carlo algorithm, as well as **R-INLA** (Lindgren and Rue 2015), for approximate inference. Domain-specific inference is facilitated by specialized packages, such as **MCMCglmm** (Hadfield 2010) and **brms** (Bürkner 2018) for the R language (R Core Team 2020). In the domain of multivariate time series analysis, the R package **vars** (Pfaff 2008) represents a cornerstone. It offers a comprehensive set of frequentist VAR-related functionalities, including the calculation and visualization of forecasts, impulse responses, and forecast error variance decompositions. Other related packages include **MTS** (Tsay and Wood 2018), **BigVAR** (Nicholson, Matteson, and Bien 2019), and **tsDyn** (Di Narzo, Aznarte, Stigler, and Tsung-Wu 2020), for a powerful and mature assortment of software.

Currently there exists no equivalent to **vars**, as an all-purpose tool for Bayesian VAR models in R. Applied work is often performed via ad hoc scripts, compromising reproducibility. Some R packages provide specialized implementations of Bayesian VAR models, but lack flexibility and accessibility. The **bvarsv** package (Krueger 2015) implements estimation of a model with time-varying parameters and stochastic volatility by Primiceri (2005). **mfbvar**, by Ankargren and Yang (2019), implements estimation of mixed-frequency VAR models and provides forecasting routines. Several common prior distributions as well as stochastic volatility methods are available, but functions for structural analysis and inference are lacking. Another approach is taken by the **bvartools** package (Mohr 2019), which provides functions to assist with Bayesian inference in VAR models, but does not include routines for estimation. Despite the popularity of Bayesian VAR models, there is a considerable gap between specialized Bayesian and accessible, all-purpose implementations.

In this paper, we present **BVAR** (Kuschnig and Vashold 2020), a comprehensive and user-friendly R package for the estimation and analysis of Bayesian VAR models. It implements a hierarchical modeling approach to prior selection in the fashion of Giannone *et al.* (2015). Functionalities to facilitate most common analyses are provided. Standard methods and interfaces to existing frameworks ensure accessibility and extensibility. **BVAR** is free software, licensed under the GNU General Public License 3, openly available and developed online.[1]

The remainder of this paper is structured as follows. Section 2 describes the econometric framework used in the package. Section 3 provides an overview of **BVAR** and its functionalities, with their usage demonstrated by means of an example in Section 4. Section 5 concludes.

[1]See the Comprehensive R Archive Network (CRAN, https://CRAN.R-project.org/package=BVAR) and GitHub (https://github.com/nk027/bvar).

2. Econometric framework

BVAR takes a Bayesian hierarchical modeling approach to VAR models. This section introduces the model, prior specification, and the hierarchical prior selection procedure proposed by Giannone *et al.* (2015). For further information on VAR models, the Bayesian approach to them, as well as Bayesian estimation, and inference in general we refer the interested reader to Kilian and Lütkepohl (2017), Koop and Korobilis (2010), and Gelman, Carlin, Stern, Dunson, Vehtari, and Rubin (2013) respectively.

2.1. Model specification

VAR models are a generalization of univariate autoregressive (AR) models, based on the notion of interdependencies between lagged values of all variables in a given model. They are commonly resorted to as tools for investigating dynamic effects of shocks and perform well in forecasting exercises. A VAR model of finite order p, referred to as VAR(p) model, can be expressed as:

$$\boldsymbol{y}_t = \boldsymbol{a}_0 + \boldsymbol{A}_1 \boldsymbol{y}_{t-1} + \cdots + \boldsymbol{A}_p \boldsymbol{y}_{t-p} + \boldsymbol{\epsilon}_t, \text{ with } \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}), \tag{1}$$

where $\boldsymbol{y}_t$ is an $M \times 1$ vector of endogenous variables, $\boldsymbol{a}_0$ is an $M \times 1$ intercept vector, $\boldsymbol{A}_j$ $(j = 1, \ldots, p)$ are $M \times M$ coefficient matrices, and $\boldsymbol{\epsilon_t}$ is an $M \times 1$ vector of Gaussian exogenous shocks with zero mean and variance-covariance (VCOV) matrix $\boldsymbol{\Sigma}$. The number of coefficients to be estimated is $M + M^2 p$, rising quadratically with the number of included variables and linearly in the lag order. Such a dense parameterization often leads to inaccuracies with regard to out-of-sample forecasting and structural inference, especially for higher-dimensional models. This phenomenon is commonly referred to as the curse of dimensionality.

The Bayesian approach to estimation of VAR models tackles this limitation by imposing additional structure on the model. Informative conjugate priors have been shown to be effective in mitigating the curse of dimensionality and allow for large models to be estimated (see Bańbura, Giannone, and Reichlin 2010; Doan, Litterman, and Sims 1984). They push the model parameters towards a parsimonious benchmark, reducing estimation error and improving out-of-sample prediction accuracy (see Koop 2013). This type of shrinkage is related to frequentist regularization approaches (Hoerl and Kennard 1970; Tibshirani 1996), which is discussed in detail by De Mol, Giannone, and Reichlin (2008), among others. The flexibility of the Bayesian framework allows for the accommodation of a wide range of economic issues, naturally involves prior information, and can account for layers of uncertainty through hierarchical modeling (Gelman *et al.* 2013).

2.2. Prior selection and specification

Properly informing prior beliefs is critical and hence the subject of much research. In the multivariate context, flat priors, which attempt not to impose a certain belief, yield inadmissible estimators (Stein 1956) and poor inference (Sims 1980; Bańbura *et al.* 2010). Other uninformative or informative priors are necessary. Early contributions (Litterman 1980) set the priors and their parameters in a way that maximizes out-of-sample forecasting performance over a pre-sample. Del Negro and Schorfheide (2004) choose values that maximize the marginal data density. Bańbura *et al.* (2010) use the in-sample fit as decision criterion and control for overfitting. Economic theory is a preferred source of prior information, but

is lacking in many settings – in particular for high-dimensional models. Acknowledging this, Villani (2009) reformulates the model and places priors on the steady state, which is better understood theoretically by economists.

Giannone *et al.* (2015) propose setting prior parameters in a data-based fashion, i.e., by treating them as additional parameters to be estimated. In their hierarchical approach, prior parameters are assigned their own hyperpriors with hyperparameters. Uncertainty surrounding the choice of prior parameters is acknowledged explicitly. This can be expressed by invoking Bayes' law as:

$$p(\boldsymbol{\gamma}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}|\boldsymbol{\gamma})\ p(\boldsymbol{\gamma}), \tag{2}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \int p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}|\boldsymbol{\gamma}) d\boldsymbol{\theta}, \tag{3}$$

where $\boldsymbol{y} = (\boldsymbol{y}_{p+1}, \ldots, \boldsymbol{y}_T)^\top$, the autoregressive and variance parameters of the VAR model are denoted by $\boldsymbol{\theta}$, and the set of hyperparameters with $\boldsymbol{\gamma}$. The first part of Equation 2 is marginalized with respect to the parameters $\boldsymbol{\theta}$ in Equation 3. This yields a density of the data as a function of the hyperparameters $p(\boldsymbol{y}|\boldsymbol{\gamma})$, also called marginal likelihood (ML). This quantity is marginal with respect to the parameters $\boldsymbol{\theta}$, but conditional on the hyperparameters $\boldsymbol{\gamma}$. The ML can be used as a decision criterion for the hyperparameter choice; maximization constitutes an empirical Bayes method, with a clear frequentist interpretation (Giannone *et al.* 2015). In the Bayesian hierarchical approach, the ML is used to explore the full posterior hyperparameter space, acknowledging uncertainty surrounding them. This yields robust inference, is theoretically grounded, and can be implemented in an efficient manner (Giannone *et al.* 2015). The authors demonstrate the high accuracy of impulse response functions and forecasts, with the model performing competitively compared to factor models. Since then, their approach has been used extensively in applied research (see e.g., Miranda-Agrippino and Rey 2015; Baumeister and Kilian 2016; Altavilla, Boucinha, and Peydró 2018; Nelson, Pinter, and Theodoridis 2018; Altavilla, Pariès, and Nicoletti 2019).

The contribution of Giannone *et al.* (2015) focuses on conjugate prior distributions, specifically of the Normal-inverse-Wishart (NIW) family.[2] Conjugacy implies that the ML is available in closed form, enabling efficient computation. The NIW family includes many of the most commonly used priors (Koop and Korobilis 2010; Karlsson 2013), with some notable exceptions. These include the steady-state prior (Villani 2009), the Normal-Gamma prior (Griffin and Brown 2010; Huber and Feldkircher 2019), and the Dirichlet-Laplace prior (Bhattacharya, Pati, Pillai, and Dunson 2015). Many recent contributions focus on accounting for heteroskedastic error structures (Clark 2011; Kastner and Frühwirth-Schnatter 2014; Carriero, Clark, and Marcellino 2016). This may improve model performance, but is not possible within the conjugate setup and furthermore, would complicate inference. In the chosen NIW framework we approach the model in Equation 1 by letting $\boldsymbol{A} = [\boldsymbol{a}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_p]'$ and $\boldsymbol{\beta} = vec(\boldsymbol{A})$. Then the conjugate prior setup reads as:

$$\begin{aligned} \boldsymbol{\beta}|\boldsymbol{\Sigma} &\sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Omega}), \\ \boldsymbol{\Sigma} &\sim \mathcal{IW}(\boldsymbol{\Psi}, \boldsymbol{d}), \end{aligned} \tag{4}$$

where $\boldsymbol{b}$, $\boldsymbol{\Omega}$, $\boldsymbol{\Psi}$ and $\boldsymbol{d}$ are functions of a lower-dimensional vector of hyperparameters $\boldsymbol{\gamma}$. In their paper, Giannone *et al.* (2015) consider three specific priors – the so-called Minnesota

[2] De Mol *et al.* (2008) note that this setting is similar to ridge penalized estimation in frequentist terms.

(Litterman) prior, which is used as a baseline, the sum-of-coefficients prior and the single-unit-root prior (also see Sims and Zha 1998).

The Minnesota prior (Litterman 1980) imposes the hypothesis that individual variables all follow random walk processes. This parsimonious specification typically performs well in forecasts of macroeconomic time series (Kilian and Lütkepohl 2017) and is often used as a benchmark to evaluate accuracy. The prior is characterized by the following moments:

$$\mathbb{E}[(\boldsymbol{A}_s)_{ij}|\boldsymbol{\Sigma}] = \begin{cases} 1 & \text{if } i = j \text{ and } s = 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$cov\left[(\boldsymbol{A}_s)_{ij}, (\boldsymbol{A}_r)_{kl}|\boldsymbol{\Sigma}\right] = \begin{cases} \lambda^2 \frac{1}{s^\alpha} \frac{\boldsymbol{\Sigma}_{ik}}{\psi_j/(d-M-1)} & \text{if } l = j \text{ and } r = s, \\ 0 & \text{otherwise.} \end{cases}$$

The key parameter λ controls the tightness of the prior, i.e., it weighs the relative importance of prior and data. For $\lambda \to 0$ the prior outweighs any information in the data; the posterior approaches the prior. As $\lambda \to \infty$ the posterior distribution mirrors the sample information. Governing the variance decay with increasing lag order, α controls the degree of shrinkage for more distant observations. Finally, ψ_j controls the prior's standard deviation on lags of variables other than the dependent.

Refinements of the Minnesota prior are often implemented as additional priors trying to "reduce the importance of the deterministic component implied by VAR models estimated conditioning on the initial observations" (Giannone *et al.* 2015, p. 440). This component is defined as the expectation of future observations, given initial conditions and estimated coefficients. The sum-of-coefficients prior (Doan *et al.* 1984) is one example for such an additional prior. It imposes the notion that a no-change forecast is optimal at the beginning of a time series. The prior can be implemented by adding artificial dummy-observations on top of the data matrix. They are constructed as follows:

$$\underset{M\times M}{\boldsymbol{y}^+} = diag\left(\frac{\bar{\boldsymbol{y}}}{\mu}\right),$$
$$\underset{M\times(1+Mp)}{\boldsymbol{x}^+} = [\boldsymbol{0}, \boldsymbol{y}^+, \ldots, \boldsymbol{y}^+],$$

where $\bar{\boldsymbol{y}}$ is a $M \times 1$ vector of averages over the first p – denoting the lag order – observations of each variable. The key parameter μ controls the variance and hence, the tightness of the prior. For $\mu \to \infty$ the prior becomes uninformative, while for $\mu \to 0$ the model is pulled towards a form with as many unit roots as variables and no cointegration. The latter imposition motivates the single-unit-root prior (Sims 1993; Sims and Zha 1998), which allows for cointegration relationships in the data. The prior pushes the variables either towards their unconditional mean or towards the presence of at least one unit root. Its associated dummy observations are:

$$\underset{1\times M}{\boldsymbol{y}^{++}} = \frac{\bar{\boldsymbol{y}}}{\delta},$$
$$\underset{1\times(1+Mp)}{\boldsymbol{x}^{++}} = \left[\frac{1}{\delta}, \boldsymbol{y}^{++}, \ldots, \boldsymbol{y}^{++}\right],$$

where $\bar{\boldsymbol{y}}$ is again defined as above. Similarly to before, δ is the key parameter and governs the tightness of the prior. The sum-of-coefficients and single-unit-root dummy-observation

priors are commonly used in the estimation of VAR models in levels and fit the hierarchical approach to prior selection. Note however, that the approach is applicable to all priors from the NIW family in Equation 4, yielding a flexible and readily extensible framework.

3. The BVAR package

BVAR implements a hierarchical approach to prior selection (Giannone *et al.* 2015) into R (R Core Team 2020) and hands the user an easy-to-use and flexible tool for Bayesian VAR models. Its primary use cases are in the field of macroeconomic multivariate time series analysis. **BVAR** is ideal for a broad range of economic analyses (in the spirit of Baumeister and Kilian 2016; Altavilla *et al.* 2018; Nelson *et al.* 2018). It may be consulted as a reference for similar models, where the hierarchical prior selection serves as a safeguard against unreasonable parameter choices. The accessible and user-friendly implementation make it a suitable tool for introductions to Bayesian multivariate time series modeling and for quick, versatile analysis.

The package is available cross-platform and on minimal installations, with no dependencies outside base R, and imports from **mvtnorm** (Genz, Bretz, Miwa, Mi, Leisch, Scheipl, and Hothorn 2020). It is implemented in native R for transparency and in order to lower the bar for contributions and/or adaptations. A functional approach to the package structure facilitates optimization of computationally intensive steps, including ports to e.g., C++, and ensures extensibility. The complete documentation, helper functions to access the multitude of settings, and use of established methods for analysis make the package easy to operate, without sacrificing flexibility.

BVAR features extensive customization options with regard to the elicited priors, their parameters, and their hierarchical treatment. The Minnesota prior is used as baseline; all of its parameters are adjustable and can be treated hierarchically. Users can easily include the sum-of-coefficients and single-unit-root priors of Sims and Zha (1998) and Giannone *et al.* (2015). The flexible implementation also allows users to construct custom dummy-observation priors. Further options are devoted to the MCMC method and the Metropolis-Hastings (MH) algorithm, which is used to explore the posterior hyperparameter space. The number of burned and saved draws are adjustable; thinning may be employed to reduce memory requirements and serial correlation. Proper exploration of the posterior is facilitated by options to manually scale individual proposals for the MH step, or to enable automatic scaling until a target acceptance rate is achieved. The customization options can be harnessed for flexible analysis with a number of established and specialized methods.

A major function and common application of VAR models are predictions. VAR-based forecasts have proven to be superior to many other methods (Bańbura *et al.* 2010; Koop 2013). They do not rely on inducing particular restrictions on model parameters, as is the case for structural models. **BVAR** can be used to conduct both classic unconditional as well as conditional forecasts. Unconditional forecasts are implemented to mirror base R for straightforward use. They rival those obtained from factor models in accuracy (Giannone *et al.* 2015) and can be used for a variety of analyses. Conditional forecasts allow for elaborate scenario analyses, where the future path of one or more variables is assumed to be known. They are a handy tool for analyzing possible realizations of policy-relevant variables. The algorithmic implementation of conditional forecasts follows Waggoner and Zha (1999) and is closely linked to structural analysis.

Impulse response functions (IRF) are a central tool for structural analysis. They provide insights into the behavior of economic systems and are another cornerstone of inference with VAR models. IRF serve as a representation of shocks hitting the economic system and are used to analyze the reactions of individual variables. The exact propagation of these shocks is of great interest, but meaningful interpretation relies on proper identification. **BVAR** features a framework for identification schemes, with two of the most popular schemes currently available – namely short-term zero restrictions and sign restrictions. The former is also known as recursive identification and is achieved via Cholesky decomposition of the VCOV matrix $\mathbf{\Sigma}$ (see Kilian and Lütkepohl 2017, Chapter 8). This approach is computationally cheap and achieves exact identification without the need for detailed assumptions about variable behavior. Only the contemporaneous reactions of certain variables are limited, making the order of variables pivotal. Sign restrictions (see Kilian and Lütkepohl 2017, Chapter 13) are another popular means of identification that is implemented following the approach of Rubio-Ramirez, Waggoner, and Zha (2010). This scheme requires some presumptions about the behavior of variables following a certain shock. With increasing dimension of the model theoretically grounding such presumptions becomes increasingly challenging. Additionally, identification via sign restrictions comes at the cost of increased uncertainty and a loss of precision for the resulting IRF. Another related tool for structural analysis are forecast error variance decompositions (FEVD). They are used to investigate which variables drive the paths of others after a given shock. FEVD can easily be computed in **BVAR** and allow for a more detailed structural analysis of the processes determining the behavior of an economic system.

BVAR packages the popular FRED-MD and FRED-QD databases (McCracken and Ng 2016, 2020). They constitute two of the largest macroeconomic databases, featuring more than 200 macroeconomic indicators on a monthly and a quarterly basis, respectively. The databases describe the US economy, starting from 1959 and are updated regularly. The databases are distributed in **BVAR** under a permissive modified Open Data Commons Attribution License (ODC-BY 1.0). Together with helper functions to aid with transformations, they allow users to start using the package hassle-free. FRED-MD and FRED-QD lend themselves to the study of a wide range of economic phenomena and are regularly used in benchmarking exercises for newly developed models and methods (see inter alia Carriero, Clark, and Marcellino 2018; Koop, Korobilis, and Pettenuzzo 2019; Huber, Koop, and Onorante 2020).

As detailed above, **BVAR** makes estimation of and inference in Bayesian VAR models accessible and user-friendly. Extensive customization options are available, with sensible default settings allowing for a step-by-step adoption. This is further facilitated by lucid helper functions and comprehensive documentation. Analysis of estimated VAR models is readily accessible – functions for summarizing and plotting model parameters, forecasts, IRF, traces, densities, and residuals are available. Use of established procedures and standard methods, including `plot()`, `predict()`, `coef()`, and `summary()`, set a low entry barrier for R users. Final and intermediate outputs are provided in an idiomatic format and feature `print()` methods for quick access and a transparent research process. Existing frameworks may be used for further analysis – an interface to **coda** for checking outputs, analysis, and diagnostics is provided. The available FRED-MD and FRED-QD datasets allow hassle-free exploration of macroeconomic research questions. These features make **BVAR** an ideal tool for macroeconomic analysis.

4. An applied example

In this section we demonstrate the functionalities of **BVAR** via an applied example. We use a subset of the included data and go through a typical workflow of (1) preparing the data, (2) configuring priors and other aspects of the model, (3) estimating the model, and finally (4) analyzing outputs, including IRF and forecasts. Further possible applications and examples are available in the Appendix. We start by setting a seed for reproducibility and loading **BVAR**.

```
R> set.seed(42)
R> library("BVAR")
```

4.1. Data preparation

The main function `bvar()` expects input data to be coercible to a rectangular numeric matrix without any missing values. For this example, we use six variables from the included FRED-QD dataset (McCracken and Ng 2020), akin to the medium VAR considered by Giannone *et al.* (2015). The six variables are real gross domestic product (GDP), real personal consumption expenditures, real gross private domestic investment (all three in billions of 2012 dollars), as well as the number of total hours worked in the non-farm business sector, the GDP deflator index as a means to measure price inflation, and the effective federal funds rate in percent per year. The currently covered time period ranges from Q1 1959 to Q1 2020. We follow Giannone *et al.* (2015) in transforming all variables except the federal funds rate to log-levels, in order to demonstrate aforementioned dummy priors. Transformation can be performed manually or with the helper function `fred_transform()`. The function supports transformations listed by McCracken and Ng (2016, 2020), which can be accessed via their transformation codes, and automatic transformation. See Appendix A for a demonstration of this and related functionalities. For our example, we specify a log-transformation for the corresponding variables with code `4` and no transformation for the federal funds rate with code `1`. Figure 1 provides an overview of the transformed time series.

```
R> data("fred_qd")
R> x <- fred_qd[, c("GDPC1", "PCECC96", "GPDIC1",
+    "HOANBS", "GDPCTPI", "FEDFUNDS")]
R> x <- fred_transform(x, codes = c(4, 4, 4, 4, 4, 1))
```

4.2. Prior setup and further configuration

After preparing the data, we are ready to specify priors and configure our model. Functions related to estimation setup and configuration share the prefix `bv_`. They are grouped in this way to make them easily discernible and their documentations accessible, facilitating their use. This contrasts methods and functions for analysis, which stick closely to idiomatic R.

Priors are set up using `bv_priors()`, which holds arguments for the Minnesota and dummy-observation priors as well as their hierarchical treatment. We start by adjusting the Minnesota prior using `bv_minnesota()`. The prior parameter λ has a Gamma hyperprior and is handed upper and lower bounds for its Gaussian proposal distribution in the MH step. For this

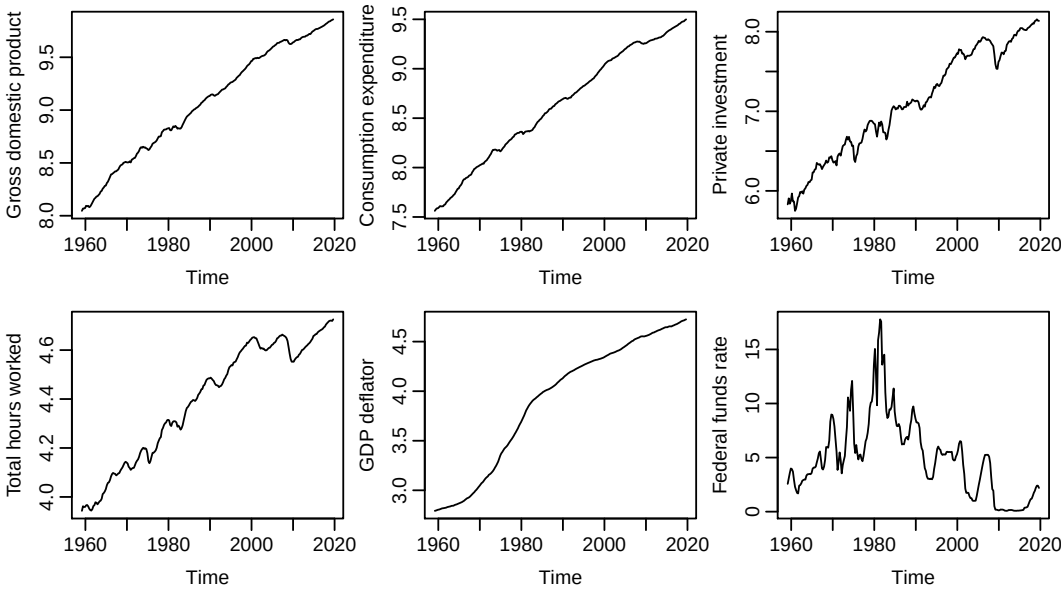

Figure 1: Transformed time series under consideration.

example, we do not treat α hierarchically, meaning the parameter can be fixed via the `mode` argument. The prior variance on the constant term of the model (`var`) is dealt a large value, for a diffuse prior. We leave $\mathbf{\Psi}$ to be set automatically – i.e., to the square root of the innovations variance, after fitting AR(p) models to each of the variables.

```
R> mn <- bv_minnesota(
+    lambda = bv_lambda(mode = 0.2, sd = 0.4, min = 0.0001, max = 5),
+    alpha = bv_alpha(mode = 2), var = 1e07)
```

We also include the sum-of-coefficients and single-unit-root priors – two pre-constructed dummy-observation priors. The hyperpriors of their key parameters are assigned Gamma distributions, with specification working in the same way as for λ. Custom dummy-observation priors can be set up similarly via `bv_dummy()` and require an additional function to construct the observations (see Appendix B for a demonstration).

```
R> soc <- bv_soc(mode = 1, sd = 1, min = 1e-04, max = 50)
R> sur <- bv_sur(mode = 1, sd = 1, min = 1e-04, max = 50)
```

Once the priors are defined, we provide them to `bv_priors()`. The dummy-observation priors are captured by the ellipsis argument (`...`) and need to be named. Via `hyper` we choose which prior parameters should be treated hierarchically. Its default setting (`"auto"`) includes λ and the key parameters of all provided dummy-observation priors. In our case, this is equivalent to providing the character vector `c("lambda", "soc", "sur")`. Prior parameters that are not treated hierarchically, e.g., α, are treated as fixed and set equal to their `mode`.

```
R> priors <- bv_priors(hyper = "auto", mn = mn, soc = soc, sur = sur)
```

As a final step before estimation, we adjust the the MH algorithm via `bv_metropolis()`. The function allows for fine-tuning the exploration of the posterior space – a vital prerequisite for proper inference. The primary argument is `scale_hess`, a scalar or vector. It allows scaling the inverse Hessian, which is used as VCOV matrix of the Gaussian proposal distribution of the hierarchically treated hyperparameters. This affords us the flexibility of individually tweaking the posterior exploration of hyperparameters. Scaling can be complemented by setting `adjust_acc = TRUE`, which enables automatic scale adjustment. This happens during an initial share of the burn-in period, adaptable via `adjust_burn`. Automatic adjustment is performed iteratively by `acc_change` percent, until an acceptance rate between `acc_lower` and `acc_upper` is reached.

```
R> mh <- bv_metropolis(scale_hess = c(0.05, 0.0001, 0.0001),
+    adjust_acc = TRUE, acc_lower = 0.25, acc_upper = 0.45)
```

After configuring the model's priors and the MH step we are ready for estimation. Further available configuration options for the MCMC method, IRF, FEVD, and forecasts are described in the following paragraphs. On the one hand, the available settings permit users to tailor models and specific components to their individual needs. This enables them to address an extensive set of research questions. On the other hand, much simpler and quicker utilization is possible – the default settings provide a suitable point of departure for many applications and the hierarchical approach brings additional parameter flexibility. This enables users to (1) focus on critical parts of their model and (2) use **BVAR** with ease, facilitating gradual fine-tuning of models.

4.3. Estimation of the model

Models are estimated using the core function `bvar()`. As the bare minimum, we need to provide our prepared data and a lag order p as arguments. Additionally, we pass on our customization objects from above to their respective arguments. We also define the total number of iterations with `n_draw`, the number of initial iterations to discard with `n_burn`, and a denominator for the fraction of draws to store via `n_thin`. We increase the number of total and burnt iterations, while retaining all draws. Note that arguments for computing IRF, FEVD, and forecasts are also available and work similarly to the ex-post calculations that are demonstrated below. When estimating the model, `verbose = TRUE` prompts printing of intermediate results and enables a progress bar during the MCMC step.

```
R> run <- bvar(x, lags = 5, n_draw = 15000, n_burn = 5000, n_thin = 1,
+    priors = priors, mh = mh, verbose = TRUE)

Optimisation concluded.
Posterior marginalised likelihood: 3637.405
Parameters: lambda = 1.51378; soc = 0.12618; sur = 0.47674
|================================================| 100%
Finished MCMC after 16.89 secs.
```

The return value is an object of class `'bvar'` – a named list with several outputs. These include the parameters of interest, i.e., posterior draws of the VAR coefficients, the VCOV

matrix, and hierarchically treated hyperparameters. Other content includes the values of the marginal likelihood for each draw, starting values of the prior hyperparameters obtained from `optim`, prior settings provided, as well as ones set automatically, and the original call to the `bvar()` function. A variety of meta information is included as well – e.g., the number of accepted draws, variable names, and time spent calculating. In addition, there are slots for the outputs of IRF, FEVD, and forecasts. They are filled automatically if calculated, or can be appended ex-post via replacement functions. Outputs can be accessed manually or via a multitude of functions and methods for analysis.

4.4. Analyzing outputs

BVAR provides a range of standard methods for objects of type `'bvar'` and derivatives, which facilitate cursory assessments and detailed analysis. These include `print()`, `plot()`, and `summary()` methods, as well as a `predict()` method and an `irf()` generic. The `print()` method provides some meta information, details on hierarchically treated prior hyperparameters, and their starting values obtained from optimization via `optim()`. The `summary()` method mimics its counterpart in **vars**, including information on the VAR model's log-likelihood, coefficients, and the VCOV matrix. These are also available via the methods `logLik()`, `coef()` and `vcov()`. Other established methods, such as `fitted()`, `density()` and `residuals()`, are provided. They operate on all posterior draws and include clear and concise `print()` methods.

For our example, we access an overview of our estimation using `print()`. Then we use `plot()` to assess convergence of the MCMC algorithm, which is essential for its stability. By default, the method provides trace and density plots of the ML and the hierarchically treated hyperparameters (see Figure 2). Burnt draws are not included and parameter boundaries are plotted as dashed gray lines. The plot can be subset to specific hyperparameters or autoregressive coefficients via the `vars` argument (see Figure 3). The arguments `var_response` and `var_impulse` provide a concise alternative way of retrieving autoregressive coefficients. We can also use the `type` argument to choose a specific type of plot.

```
R> print(run)

  Bayesian VAR consisting of 238 observations, 6 variables and 5 lags.
  Time spent calculating: 16.89 secs
  Hyperparameters: lambda, soc, sur
  Hyperparameter values after optimisation: 1.51378, 0.12618, 0.47674
  Iterations (burnt / thinning): 15000 (5000 / 1)
  Accepted draws (rate): 3933 (0.393)

R> plot(run)
R> plot(run, type = "dens",
+    vars_response = "GDPC1", vars_impulse = "GDPC1-lag1")
```

Visual inspection of trace and density plots suggests convergence of the key hyperparameters. The chain appears to be exploring the posterior rather well; no glaring outliers are recognizable. However, as a supplement to this examination, one might want to employ additional convergence diagnostics. The **coda** package provides, among many other useful functionalities,

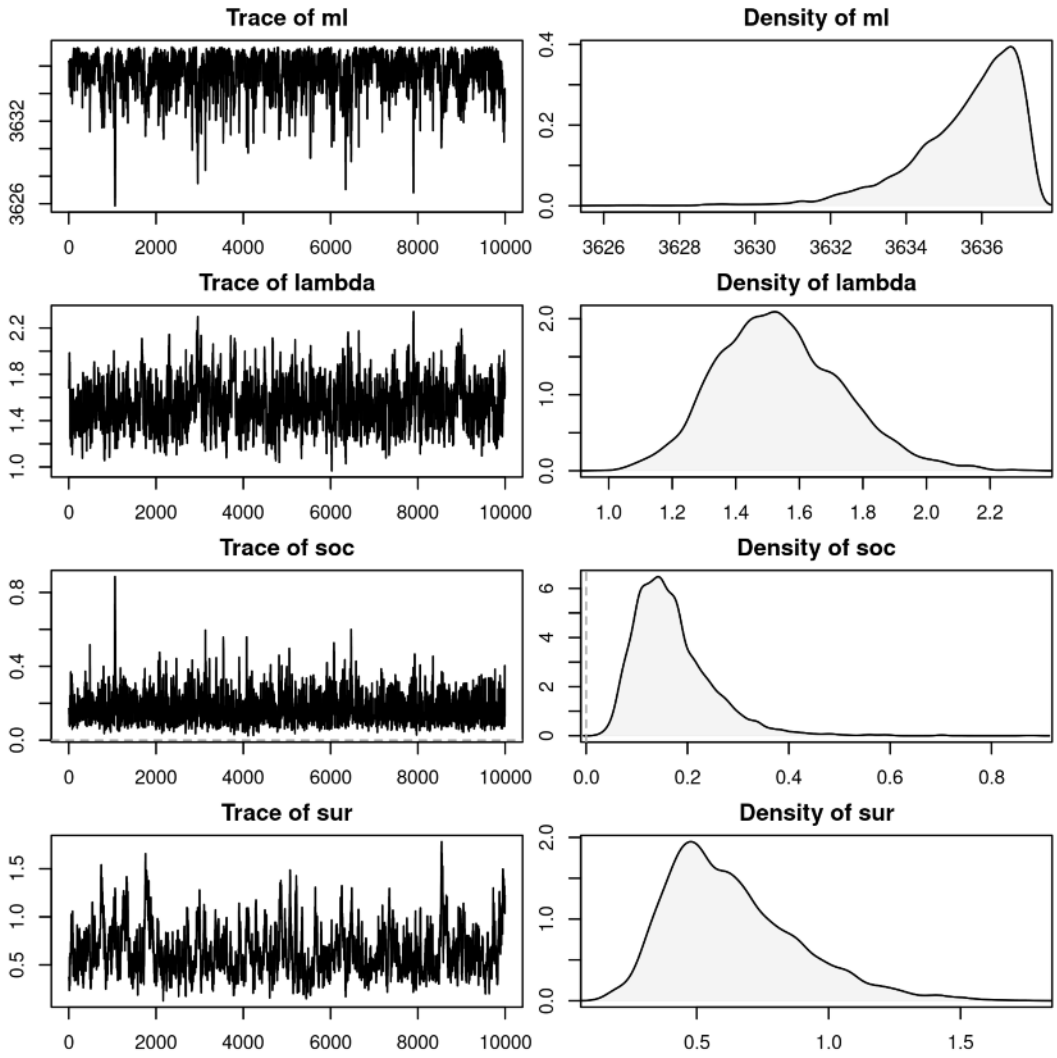

Figure 2: Trace and density plots of all hierarchically treated hyperparameters and the ML.

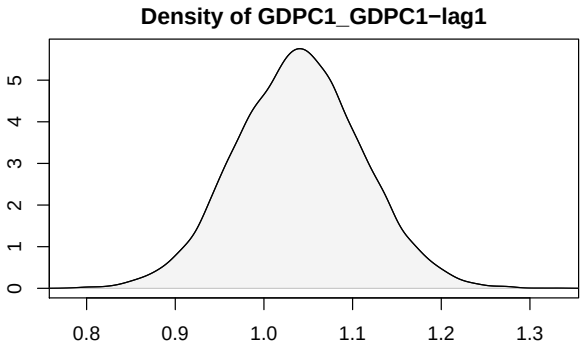

Figure 3: Density plot for the autoregressive coefficient corresponding to the first lag of GDP in the GDP equation.

several such statistics that can be accessed using **BVAR**'s `as.mcmc()` method. An illustration of the interface, the use of diagnostics, and parallel execution of `bvar()` is provided in Appendix C. Given proper convergence, we are interested in fitted and residual values (see Figure 4). We set `type = "mean"` to use the mean of posterior draws. Alternatively, credible bands can be computed via the `conf_bands` argument.

```
R> fitted(run, type = "mean")

Numeric array (dimensions 238, 6) with fitted values from a BVAR.
Average values:
        GDPC1: 8.1, 8.1, 8.1, [...], 9.85, 9.85, 9.86
        PCECC96: 7.61, 7.62, 7.61, [...], 9.48, 9.49, 9.5
        GPDIC1: 5.97, 5.89, 5.85, [...], 8.15, 8.15, 8.14
        HOANBS: 3.97, 3.97, 3.96, [...], 4.72, 4.72, 4.72
        GDPCTPI: 2.81, 2.81, 2.82, [...], 4.72, 4.72, 4.73
        FEDFUNDS: 4.05, 3.64, 2.45, [...], 2.3, 2.32, 2.3

R> plot(residuals(run, type = "mean"), vars = c("GDPC1", "PCECC96"))
```

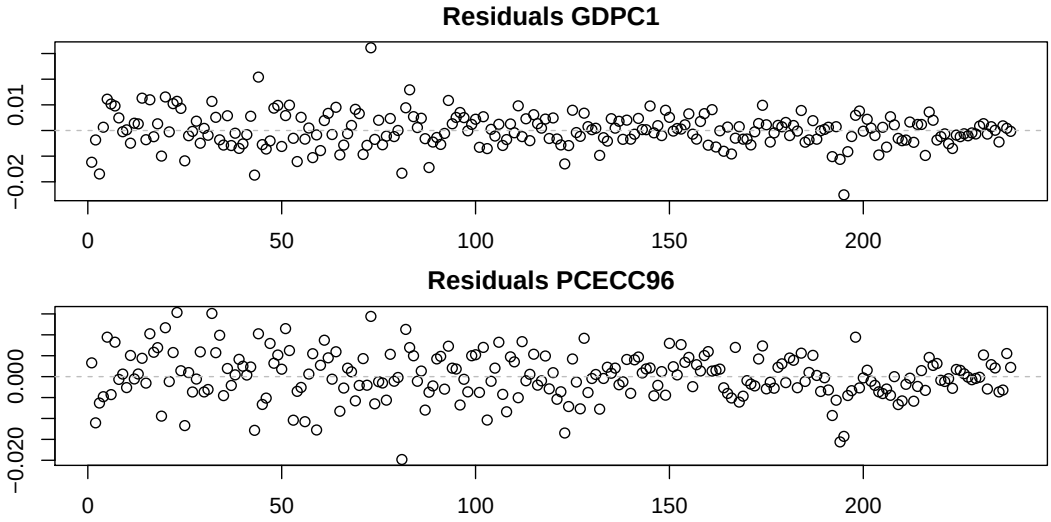

Figure 4: Residual plots of GDP and private consumption expenditure.

Structural analysis with **BVAR** works in a straightforward fashion. Impulse response functions are handled in a specific object, with the associated generic function `irf()`. The function is used for computing, accessing, and storing IRF in the respective slot of a `'bvar'` object as well as updating credible bands. Forecast error variance decompositions rely on IRF and are nested in the respective object. They can be accessed directly with the generic function `fevd()`. Configuration options for IRF and FEVD are available via the ellipsis argument of `irf()`, or the helper function `bv_irf()`. They include the horizon to be considered, whether or not FEVD should be computed, and further settings regarding identification. By default, the shocks under scrutiny are identified via short-term zero restrictions. Identification can also be achieved in other ways (see Appendix D for an example of imposing sign restrictions)

or skipped entirely. IRF objects feature methods for plotting, printing, and summarizing. The `plot()` method has options to subset the plots to specific impulses and/or responses via name or position of the variable. Credible bands are visualized as lines or shaded areas with the `area` argument toggled. In the example below, we customize our IRF using `bv_irf()`, then compute and store them with `irf()`. We plot the IRF of certain shocks and variables by specifying them with `vars_impulse` and `vars_response` in Figure 5.

```
R> opt_irf <- bv_irf(horizon = 16, identification = TRUE)
R> irf(run) <- irf(run, opt_irf, conf_bands = c(0.05, 0.16))

R> plot(irf(run), area = TRUE,
+    vars_impulse = c("GDPC1", "FEDFUNDS"), vars_response = c(1:2, 6))
```

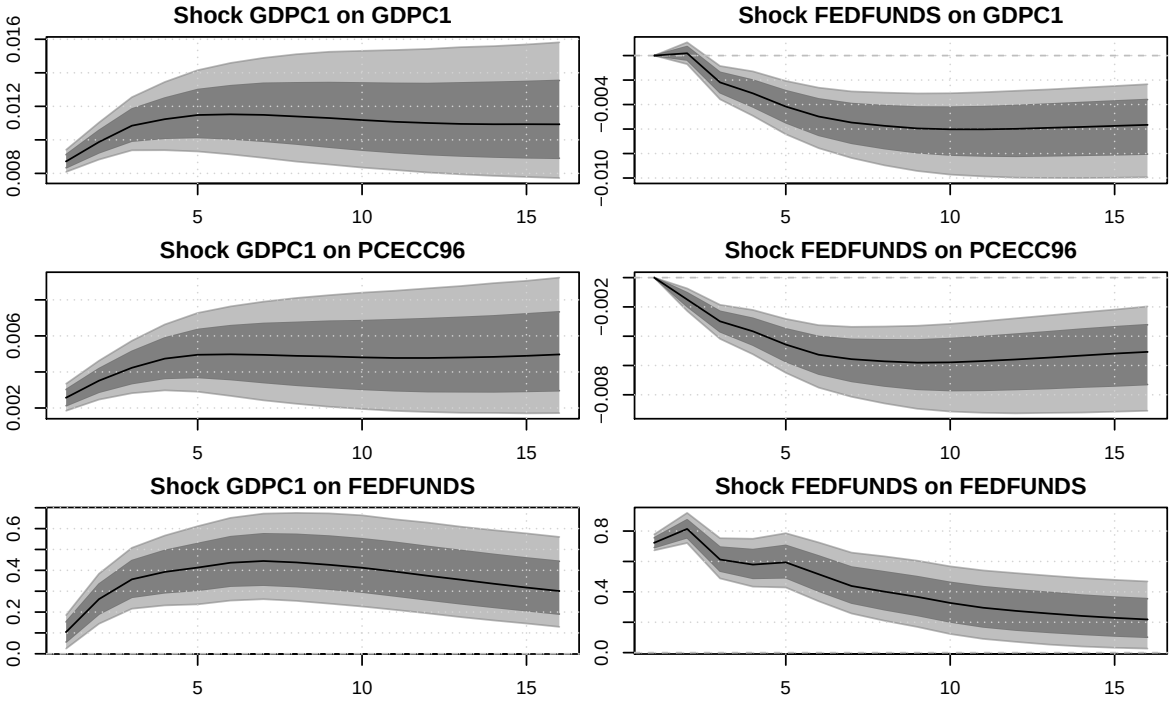

Figure 5: Impulse responses of GDP, private consumption expenditure and the federal funds rate to an aggregate demand shock (left panels) and a monetary policy shock (right panels). Shaded areas refer to the 90% and the 68% credible sets.

Forecasting with **BVAR** is facilitated by the `predict()` method and a specific object for forecasts. The method works similarly to `irf()`, with functionality to compute, access, and store outputs in the respective slot of a `'bvar'` object and to update credible bands. Settings can also be accessed via the ellipsis argument or the helper function `bv_fcast()`. For unconditional forecasts only the forecasting horizon is required. In order to conduct scenario analyses based on conditional forecasts, further settings have to be passed on (see Appendix E for a demonstration). Forecast objects feature methods for plotting, printing, and summarizing. The `vars` argument of the `plot()` method can be used to subset plots to certain variables.

The visualization can include a number of realized values before the forecast, which is set with the argument `t_back` (see Figure 6).

```
R> predict(run) <- predict(run, horizon = 16, conf_bands = c(0.05, 0.16))

R> plot(predict(run), area = TRUE, t_back = 32,
+    vars = c("GDPC1", "GDPCTPI", "FEDFUNDS"))
```

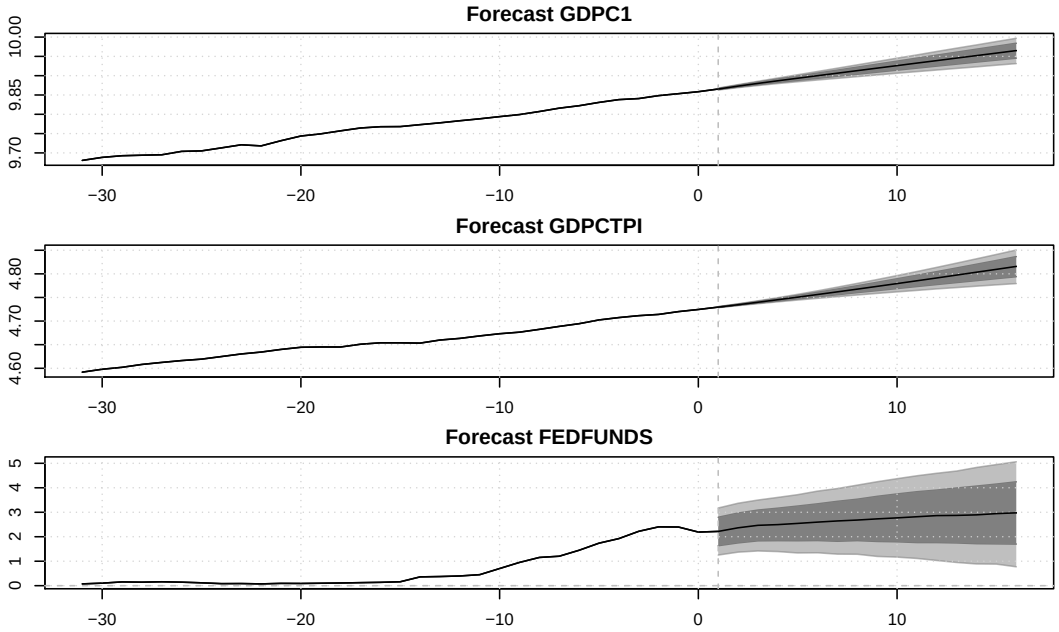

Figure 6: Unconditional forecasts for GDP, the GDP deflator and the federal funds rate. Shaded areas refer to the 90% and the 68% credible sets.

This concludes the demonstration of setup, estimation, and analysis with **BVAR**. A comprehensive description of the package and available functionalities is provided in the package documentation. Some more advanced and specific features, including identification via sign restrictions, conditional forecasts, and the interface to **coda**, are demonstrated in the Appendix.

5. Conclusion

This article introduced **BVAR**, an R package that implements Bayesian vector autoregressions with hierarchical prior selection. It offers a flexible, but structured and transparent tool for multivariate time series analysis and can be used to assess a wide range of research questions. By means of an applied example, we illustrated the usage of the package and explained its implementation and configuration. The hierarchical prior selection mitigates subjective choices, improving flexibility and counteracting a common criticism of Bayesian methods. Accessible helper functions for customization as well as comprehensive methods and generic functions for analysis top off an extensive set of features. The functional style and idiomatic implementation in R make the package easy to use, extensible, and transparent.

BVAR bridges a gap as an accessible all-purpose tool for Bayesian VAR models, but leaves plenty of potential for extensions and novel libraries. One evident extension is the implementation of additional structural identification schemes, such as sign and zero restrictions (Arias, Rubio-Ramírez, and Waggoner 2018). The interface to **coda** demonstrates the potential of such extensions – interfaces to further packages facilitating analysis and visualization, e.g., **tidybayes** (Kay 2020), would be useful. A major area for future work is support for a broader spectrum of prior families. This would help enrich analysis and could incorporate e.g., heteroskedastic error structures in various forms. A powerful library implementing these features would be a valuable asset and complement **BVAR** and similar R packages.

Computational details

The results in this paper were obtained using R 4.0.0 with **BVAR** 1.0.0, **mvtnorm** 1.1-0, and **coda** 0.19-3. The machine used to generate the results is running Windows 10 with an i5-4670k and 16 GB RAM. The scripts were also tested on a machines running Debian Sid 11 with an i7-7500U and 16 GB RAM. and a machine running Ubuntu 18.04 LTS with a Ryzen-3500U and 16 GB RAM. R and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgments

We would like to thank Victor Maus and Florian Huber for helpful comments and encouragement, the staff at the St. Louis Federal Reserve for their work on FRED-QD and FRED-MD, and the JSS editorial team for improvement suggestions. Our thanks also go out to Gregor Kastner, Casper Engelen, Daran Demirol, Maximilian Böck, Michael Pfarrhofer, Niko Hauzenberger, Sebastian Luckeneder, and two anonymous referees.

References

Altavilla C, Boucinha M, Peydró JL (2018). "Monetary Policy and Bank Profitability in a Low Interest Rate Environment." *Economic Policy*, **33**(96), 531–586. doi:10.1093/epolic/eiy013.

Altavilla C, Pariès MD, Nicoletti G (2019). "Loan Supply, Credit Markets and the Euro Area Financial Crisis." *Journal of Banking & Finance*, **109**, 105658. doi:10.1016/j.jbankfin.2019.105658.

Ankargren S, Yang Y (2019). *Mixed-Frequency Bayesian VAR Models in R: The **mfbvar** Package*. R package version 0.5.3, URL https://CRAN.R-project.org/package=mfbvar.

Arias JE, Rubio-Ramírez JF, Waggoner DF (2018). "Inference Based on Structural Vector Autoregressions Identified with Sign and Zero Restrictions: Theory and Applications." *Econometrica*, **86**(2), 685–720. doi:10.3982/ECTA14468.

Bańbura M, Giannone D, Reichlin L (2010). "Large Bayesian Vector Auto Regressions." *Journal of Applied Econometrics*, **25**(1), 71–92. doi:10.1002/jae.1137.

Baumeister C, Kilian L (2016). "Understanding the Decline in the Price of Oil since June 2014." *Journal of the Association of Environmental and Resource Economists*, **3**(1), 131–158. doi:10.1086/684160.

Bernanke BS, Boivin J, Eliasz P (2005). "Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach." *The Quarterly Journal of Economics*, **120**(1), 387–422. doi:10.1162/0033553053327452.

Bhattacharya A, Pati D, Pillai NS, Dunson DB (2015). "Dirichlet–Laplace Priors for Optimal Shrinkage." *Journal of the American Statistical Association*, **110**(512), 1479–1490. doi:10.1080/01621459.2014.960967.

Bürkner PC (2018). "Advanced Bayesian Multilevel Modeling with the R Package **brms**." *The R Journal*, **10**(1), 395–411. doi:10.32614/RJ-2018-017.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1), 1–32. doi:10.18637/jss.v076.i01.

Carriero A, Clark TE, Marcellino M (2016). "Common Drifting Volatility in Large Bayesian VARs." *Journal of Business & Economic Statistics*, **34**(3), 375–390. doi:10.1080/07350015.2015.1040116.

Carriero A, Clark TE, Marcellino M (2018). "Measuring Uncertainty and its Impact on the Economy." *Review of Economics and Statistics*, **100**(5), 799–815. doi:10.1162/rest_a_00693.

Clark TE (2011). "Real-Time Density Forecasts From Bayesian Vector Autoregressions With Stochastic Volatility." *Journal of Business & Economic Statistics*, **29**(3), 327–341. doi:10.1198/jbes.2010.09248.

De Mol C, Giannone D, Reichlin L (2008). "Forecasting Using a Large Number of Predictors: Is Bayesian Shrinkage a Valid Alternative to Principal Components?" *Journal of Econometrics*, **146**(2), 318–328. doi:10.1016/j.jeconom.2008.08.011.

Del Negro M, Schorfheide F (2004). "Priors From General Equilibrium Models for VARs." *International Economic Review*, **45**(2), 643–673. doi:10.1111/j.1468-2354.2004.00139.x.

Del Negro M, Schorfheide F, Smets F, Wouters R (2007). "On the Fit of New Keynesian Models." *Journal of Business & Economic Statistics*, **25**(2), 123–143. doi:10.1198/073500107000000016.

Di Narzo AF, Aznarte JL, Stigler M, Tsung-Wu H (2020). ***tsDyn***: *Nonlinear Time Series Models with Regime Switching*. R package version 10-1.2, URL https://CRAN.R-project.org/package=tsDyn.

Doan T, Litterman R, Sims C (1984). "Forecasting and Conditional Projection Using Realistic Prior Distributions." *Econometric Reviews*, **3**(1), 1–100. doi:10.1080/07474938408800053.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. CRC Press. ISBN 978-1439840955. doi:10.1201/b16018.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472. doi:10.1214/ss/1177011136.

Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2020). ***mvtnorm***: *Multivariate Normal and t Distributions*. R package version 1.1-0, URL https://CRAN.R-project.org/package=mvtnorm.

Geweke J (1992). "Evaluating the Accuracy of Sampling-Based Approaches to the Calculations of Posterior Moments." In JM Bernardo, JO Berger, AP Dawid, AFM Smith (eds.), *Bayesian Statistics*, volume 4, pp. 641–649. Clarendon Press. ISBN 9780198522669.

Giannone D, Lenza M, Primiceri GE (2015). "Prior Selection for Vector Autoregressions." *Review of Economics and Statistics*, **97**(2), 436–451. doi:10.1162/REST_a_00483.

Griffin JE, Brown PJ (2010). "Inference with Normal-Gamma Prior Distributions in Regression Problems." *Bayesian Analysis*, **5**(1), 171–188. doi:10.1214/10-BA507.

Hadfield JD (2010). "MCMC Methods for Multi-Response Generalized Linear Mixed Models: The **MCMCglmm** R Package." *Journal of Statistical Software*, **33**(2), 1–22. doi:10.18637/jss.v033.i02.

Hoerl AE, Kennard RW (1970). "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics*, **12**(1), 55–67. doi:10.1080/00401706.1970.10488634.

Huber F, Feldkircher M (2019). "Adaptive Shrinkage in Bayesian Vector Autoregressive Models." *Journal of Business & Economic Statistics*, **37**(1), 27–39. doi:10.1080/07350015.2016.1256217.

Huber F, Koop GM, Onorante L (2020). "Inducing Sparsity and Shrinkage in Time-Varying Parameter Models." *Journal of Business & Economic Statistics*, (just-accepted), 1–48. doi:10.1080/07350015.2020.1713796.

Karlsson S (2013). "Forecasting with Bayesian Vector Autoregression." In *Handbook of Economic Forecasting*, volume 2, pp. 791–897. Elsevier. doi:10.1016/B978-0-444-62731-5.00015-4.

Kastner G, Frühwirth-Schnatter S (2014). "Ancillarity-Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Estimation of Stochastic Volatility Models." *Computational Statistics & Data Analysis*, **76**, 408 – 423. doi:10.1016/j.csda.2013.01.002.

Kay M (2020). ***tidybayes**: Tidy Data and Geoms for Bayesian Models*. doi:10.5281/zenodo.1308151. R package version 2.0.3, URL https://CRAN.R-project.org/package=tidybayes/.

Kilian L, Lütkepohl H (2017). *Structural Vector Autoregressive Analysis*. Cambridge University Press. doi:10.1017/9781108164818.

Koop GM (2013). "Forecasting with Medium and Large Bayesian VARs." *Journal of Applied Econometrics*, **28**(2), 177–203. doi:10.1002/jae.1270.

Koop GM, Korobilis D (2010). "Bayesian Multivariate Time Series Methods for Empirical Macroeconomics." *Foundations and Trends in Econometrics*, **3**(4), 267–358. doi:10.1561/0800000013.

Koop GM, Korobilis D, Pettenuzzo D (2019). "Bayesian Compressed Vector Autoregressions." *Journal of Econometrics*, **210**(1), 135–154. doi:10.1016/j.jeconom.2018.11.009.

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.

Kuschnig N, Vashold L (2020). ***BVAR**: Hierarchical Bayesian Vector Autoregression*. R package version 1.0.0, URL https://CRAN.R-project.org/package=BVAR.

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with **R-INLA**." *Journal of Statistical Software*, **63**(19), 1–25. doi:10.18637/jss.v063.i19.

Litterman RB (1980). "A Bayesian Procedure for Forecasting with Vector Autoregressions." *MIT Working Paper*, (275).

Litterman RB (1986). "Forecasting with Bayesian Vector Autoregressions – Five Years of Experience." *Journal of Business & Economic Statistics*, **4**(1), 25–38. doi:10.1080/07350015.1986.10509491.

Lunn DJ, Spiegelhalter D, Thomas A, Best N (2009). "The BUGS Project: Evolution, Critique and Future Directions." *Statistics in Medicine*, **28**(25), 3049–3067. doi:10.1002/sim.3680.

Lunn DJ, Thomas A, Best N, Spiegelhalter D (2000). "WinBUGS – A Bayesian Modelling Framework: Concepts, Structure, and Extensibility." *Statistics and Computing*, **10**(4), 325–337. doi:10.1023/a:1008929526011.

McCracken MW, Ng S (2016). "FRED-MD: A Monthly Database for Macroeconomic Research." *Journal of Business & Economic Statistics*, **34**(4), 574–589. doi:10.1080/07350015.2015.1086655.

McCracken MW, Ng S (2020). "FRED-QD: A Quarterly Database for Macroeconomic Research." *NBER Working Papers*, (26872). doi:10.3386/w26872.

Miranda-Agrippino S, Rey H (2015). "US Monetary Policy and the Global Financial Cycle." *NBER Working Papers*, (21722). doi:10.3386/w21722. URL https://www.nber.org/papers/w21722.

Mohr F (2019). ***bvartools***: *Bayesian Inference of Vector Autoregressive Models*. R package version 0.0.2, URL https://CRAN.R-project.org/package=bvartools.

Nelson B, Pinter G, Theodoridis K (2018). "Do Contractionary Monetary Policy Shocks Expand Shadow Banking?" *Journal of Applied Econometrics*, **33**(2), 198–211. doi:10.1002/jae.2594.

Nicholson W, Matteson D, Bien J (2019). ***BigVAR***: *Dimension Reduction Methods for Multivariate Time Series*. R package version 1.0.6, URL https://CRAN.R-project.org/package=BigVAR.

Pfaff B (2008). "VAR, SVAR and SVEC Models: Implementation Within R Package **vars**." *Journal of Statistical Software*, **27**(4), 1–32. ISSN 1548-7660. doi:10.18637/jss.v027.i04.

Plummer M (2003). "JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. ISSN 1609-395X. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/.

Primiceri GE (2005). "Time Varying Structural Vector Autoregressions and Monetary Policy." *The Review of Economic Studies*, **72**(3), 821–852. doi:10.1111/j.1467-937X.2005.00353.x.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rubio-Ramirez JF, Waggoner DF, Zha T (2010). "Structural Vector Autoregressions: Theory of Identification and Algorithms for Inference." *Review of Economic Studies*, **77**(2), 665–696. doi:10.1111/j.1467-937X.2009.00578.x.

Sims CA (1980). "Macroeconomics and Reality." *Econometrica*, pp. 1–48. doi:10.2307/1912017.

Sims CA (1993). "A Nine-Variable Probabilistic Macroeconomic Forecasting Model." In *Business Cycles, Indicators and Forecasting*, pp. 179–212. University of Chicago Press. URL https://www.nber.org/chapters/c7192.

Sims CA, Zha T (1998). "Bayesian Methods for Dynamic Multivariate Models." *International Economic Review*, pp. 949–968. doi:10.2307/2527347.

Sims CA, Zha T (2006). "Were There Regime Switches in US Monetary Policy?" *American Economic Review*, **96**(1), 54–81. doi:10.1257/000282806776157678.

Stein C (1956). "Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution." In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pp. 197–206. University of California Press, Berkeley, California. URL https://projecteuclid.org/euclid.bsmsp/1200501656.

Tibshirani R (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

Tsay R, Wood D (2018). ***MTS**: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models*. R package version 1.0, URL https://CRAN.R-project.org/package=MTS.

Villani M (2009). "Steady-State Priors for Vector Autoregressions." *Journal of Applied Econometrics*, **24**(4), 630–650. doi:10.1002/jae.1065.

Waggoner DF, Zha T (1999). "Conditional Forecasts in Dynamic Multivariate Models." *Review of Economics and Statistics*, **81**(4), 639–651. doi:10.1162/003465399558508.

A. Data transformation and stationarity

Econometric analysis often relies on data coming in a specific format, frequently requiring transformation. The helper function `fred_transform()` facilitates this work step. It can be used to apply common transformations, subset data to a rectangular format without missing values, and automatically transform the included FRED-QD and FRED-MD datasets. Both datasets are packaged in their raw format, but McCracken and Ng (2016, 2020) provide suggested transformations in their online appendices.[3] These transformations can also be looked up manually with `fred_code()`.

Below, we demonstrate this using a prototypical monetary VAR model – covering GDP, the GDP deflator and the federal funds rate. For this example, we are looking to transform the individual time series to be stationary. To automatically apply the suggested transformation of McCracken and Ng (2016) we simply specify the `type` – either `"fred_qd"` or `"fred_md"` – and call `fred_transform()` with our data. In a second attempt, we deviate from the suggestions and directly provide transformation codes to the `codes` argument. The function displays available codes when called without arguments. For this example, we choose `5` for log-differences and `1` for no transformation. We also set `lag = 4` for yearly differences of our quarterly data. Figure 7 shows the transformed time series, which will be used to illustrate additional features below.

```
R> data("fred_qd")
R> y <- fred_qd[, c("GDPC1", "GDPCTPI", "FEDFUNDS")]
R> fred_transform(y, type = "fred_qd")
R> y <- fred_transform(y, codes = c(5, 5, 1), lag = 4)
```

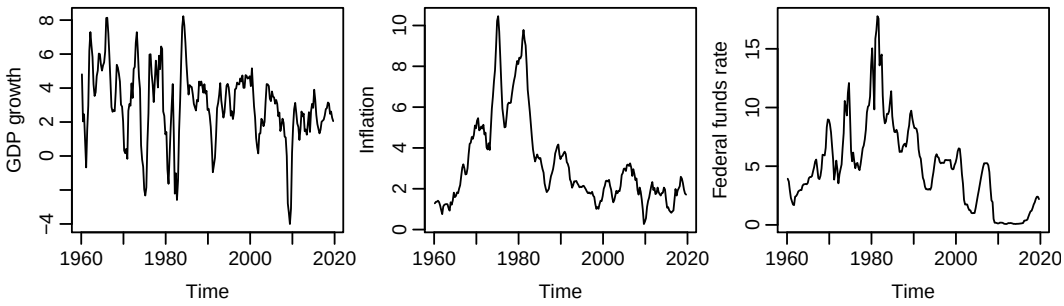

Figure 7: Transformed time series under consideration for the prototypical VAR model.

When estimating a VAR model in differences, implications on the priors need to be taken into account. The sum-of-coefficients prior and the single-unit-root prior are no longer applicable. For this example we only impose the Minnesota prior, which also needs to be adjusted slightly to still carry the notion that variables follow a random walk. This is done by setting the prior mean `b = 0` in `bv_mn()`, or its alias `bv_minnesota()`. The argument may also be provided in a vector or a matrix format, in case the variables differ in their order of integration.

```
R> priors_app <- bv_priors(mn = bv_mn(b = 0))
```

[3] See https://research.stlouisfed.org/econ/mccracken/fred-databases/.

```
R> run_app <- bvar(y, lags = 5, n_draw = 15000, n_burn = 5000,
+    priors = priors_app, mh = bv_mh(scale_hess = 0.5, adjust_acc = TRUE))
```

B. Construction of custom dummy priors

Here, we demonstrate the construction of custom dummy priors using `bv_dummy()`. As an example, the sum-of-coefficients prior is reconstructed manually.

A custom prior requires a function to construct artificial observations. This function takes three arguments – the data as a numeric matrix, an integer with the number of lags, and the value of the prior parameter. The return value is a `'list'`, containing two numeric matrices, `X` and `Y`, with artificial observations to stack on top of the data matrix and the lagged data matrix. For the sum-of-coefficients prior we follow the procedure outlined in Section 2.2.

```
R> add_soc <- function(Y, lags, par) {
+    soc <- if(lags == 1) {diag(Y[1, ]) / par} else {
+      diag(colMeans(Y[1:lags, ])) / par
+    }
+    X_soc <- cbind(rep(0, ncol(Y)), matrix(rep(soc, lags), nrow = ncol(Y)))
+    return(list("Y" = soc, "X" = X_soc))
+  }
```

This function is then passed to `bv_dummy()` via the argument `fun`. The remaining arguments work in the same way as for other prior constructors (see Section 4.2). They determine the hyperprior distribution and boundaries for the proposal distribution. Again, if not treated hierarchically, the prior parameter is set to its mode. The output of `bv_dummy()` is then passed to the ellipsis argument of `bv_priors()` and needs to be named. Further steps do not differ from standard procedure – posterior draws are stored and can be analyzed in the same way as for the Minnesota prior.

```
R> soc <- bv_dummy(mode = 1, sd = 1, min = 0.0001, max = 50, fun = add_soc)
R> bv_priors(soc = soc)
```

C. Convergence assessment and parallelization

Bayesian simulation relies heavily on the convergence of models, particularly so for hierarchical models. As demonstrated in Section 4.4, **BVAR** includes functionality to assess convergence visually. Many more tools are available through the interface to **coda**, which specializes in output assessment for Markov chain Monte Carlo (MCMC) simulations.

To access the interface we call the `as.mcmc()` method on the VAR model from Appendix A. The method works similarly to `plot()`, allowing users to subset hyperparameters or VAR coefficients with `vars`, `vars_response`, and `vars_impulse`. These are then converted to a `'mcmc'` object, which supports thorough analysis of convergence behavior. This includes the diagnostic statistics of Geweke (1992), which can be calculated with `geweke.diag()`.

```
R> library("coda")
R> run_mcmc <- as.mcmc(run_app, vars = "lambda")
R> geweke.diag(run_mcmc)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

lambda
0.9688
```

The value of the diagnostic statistic constitutes a standard z-score and indicates proper within-chain convergence of the hyperparameter λ. Still, one may also be interested in the behavior across chains. A suitable diagnostic to assess between-chain convergence was proposed by Gelman and Rubin (1992) and is implemented in **coda** as `gelman.diag()`. The need for multiple chains makes parallelization attractive, which is supported by the wrapper function `par_bvar()`. The wrapper uses `parLapply()` from the **parallel** package (R Core Team 2020) to run instances of `bvar()` on multiple threads. The output is a `'list'` of `'bvar'` objects that is supported by covergence-related methods. We can visualize the separate runs for different hyperparameters or the ML, which can be specified via the `vars` argument, with the `plot()` method (see Figure 8). We can also transform it for use with **coda**, by calling `as.mcmc()`, which now yields a `'mcmc_list'` object. This object is passed on to `gelman.diag()`, in order to compute a diagnostic statistics for between-chain convergence.

```
R> library("parallel")
R> n_cores <- 2
R> cl <- makeCluster(n_cores)
R> runs <- par_bvar(cl = cl, data = y, lags = 5,
+    n_draw = 15000, n_burn = 5000, n_thin = 1,
+    priors = priors_app, mh = bv_mh(scale_hess = 0.5, adjust_acc = TRUE))
R> stopCluster(cl)
R> runs_mcmc <- as.mcmc(runs, vars = "lambda")
R> gelman.diag(runs_mcmc, autoburnin = FALSE)

Potential scale reduction factors:

       Point est. Upper C.I.
lambda          1          1

R> plot(runs, type = "full", vars = "lambda")
```

D. Identification via sign restrictions

Identification is required for meaningful interpretation of impulse response functions. **BVAR** implements identification in a flexible manner, facilitating various schemes. Sign restrictions, one such scheme, are readily available and demonstrated here.

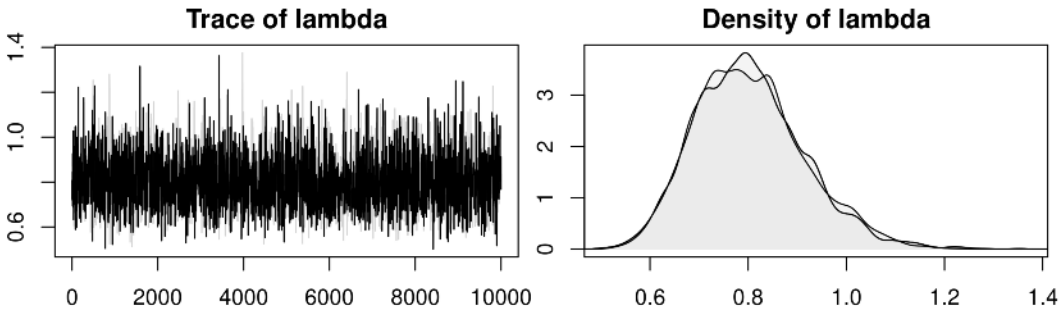

Figure 8: Plot of λ, the key hyperparameter of the Minnesota prior, for separate runs.

Identification via sign restrictions relies on forming expectations about response directions following certain shocks (Rubio-Ramirez *et al.* 2010). These expectations are usually derived from economic theory – a process that can become cumbersome for high-dimensional models. Sign identification of shocks in **BVAR** is set up in `bv_irf()`, which can be accessed directly, or over the ellipsis argument of `irf()`. We toggle `identification = TRUE` and provide a matrix **SR** with sign restrictions to the `sign_restr` argument. **SR** is constructed by setting all elements SR_{ij} equal to 1 (-1) if the contemporaneous response of variable i to a shock from variable j is expected to be an increase (decrease). For an agnostic view, we set the element to `NA`; no restrictions are imposed. Note that the shocks need to be uniquely identifiable (Kilian and Lütkepohl 2017). After running `bv_irf()`, we can print the chosen sign restrictions with its `print()` method. To calculate, we call `irf()` as usual, providing the settings to the ellipsis argument. IRF are then calculated based on suitable shocks, which are drawn following the algorithm by Rubio-Ramirez *et al.* (2010). Figure 9 provides a visualization of the restricted impulse responses.

```
R> sr <- matrix(c(1, 1, 1, -1, 1, NA, -1, -1, 1), ncol = 3)
R> opt_signs <- bv_irf(horizon = 16, fevd = TRUE,
+    identification = TRUE, sign_restr = sr)
R> print(opt_signs)

Object with settings for computing impulse responses.
Horizon: 16
Identification: Sign restrictions
Chosen restrictions:
                  Shock to
                  Var1  Var2  Var3
Response of Var1    +     -     -

            Var2    +     +     -

            Var3    +     NA    +
FEVD: TRUE

R> irf(run_app) <- irf(run_app, opt_signs)
```

```
R> plot(irf(run_app), vars_impulse = c(1, 3))
```

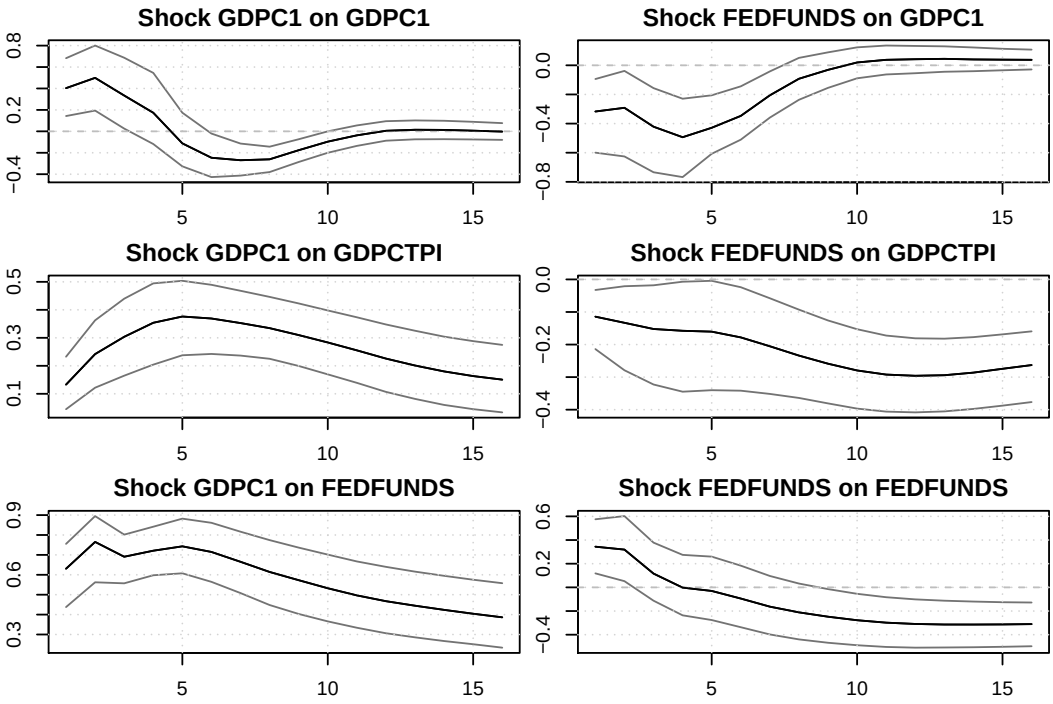

Figure 9: Set of impulse responses from a prototypical monetary VAR model using sign restrictions to achieve identification. Grey lines refer to the 68% credible set.

As can be discerned, the shocks identified via sign restrictions allow for contemporaneous relations between all variables – in contrast to ones obtained via recursive identification. The instantaneous impacts are in accordance with the imposed restrictions. Note that the credible sets surrounding the median impulse response are somewhat inflated. This stems from additional uncertainty that is introduced by drawing random orthogonal matrices in the algorithm.

E. Conditional forecasts

Conditional forecasts are a useful tool for scenario analysis, where the future path of one or multiple variable(s) is known, or assumed to be known. Here we demonstrate how to conduct such a scenario analysis using **BVAR** by means of a concise example.

Conditional forecasts rely on fixing the future paths of a number of variables, in order to gain insights into the development of the remaining unconstrained variables (Waggoner and Zha 1999). This allows researchers to compare different realizations of policy-relevant quantities and analyze impacts. In **BVAR**, we configure conditional forecasts with `bv_fcast()`, which can be accessed directly or over the ellipsis argument of `predict()`. The assumed future paths are provided to the `cond_path` argument as numeric vector or matrix; the names or positions of constrained variables to `cond_vars`. For our example, we constrain the federal funds rate to a sharp rise and subsequent drop. The forecasts are then calculated as usual,

using the `predict()` method. We use the `plot()` method to visualize the conditional forecast. Figure 10 shows the constrained federal funds rate together with unconstrained GDP growth and inflation dynamics.

```
R> path <- c(2.25, 3, 4, 5.5, 6.75, 4.25, 2.75, 2, 2, 2)
R> predict(run_app) <- predict(run_app, horizon = 16,
+    cond_path = path, cond_var = "FEDFUNDS")

R> plot(predict(run_app), t_back = 16)
```

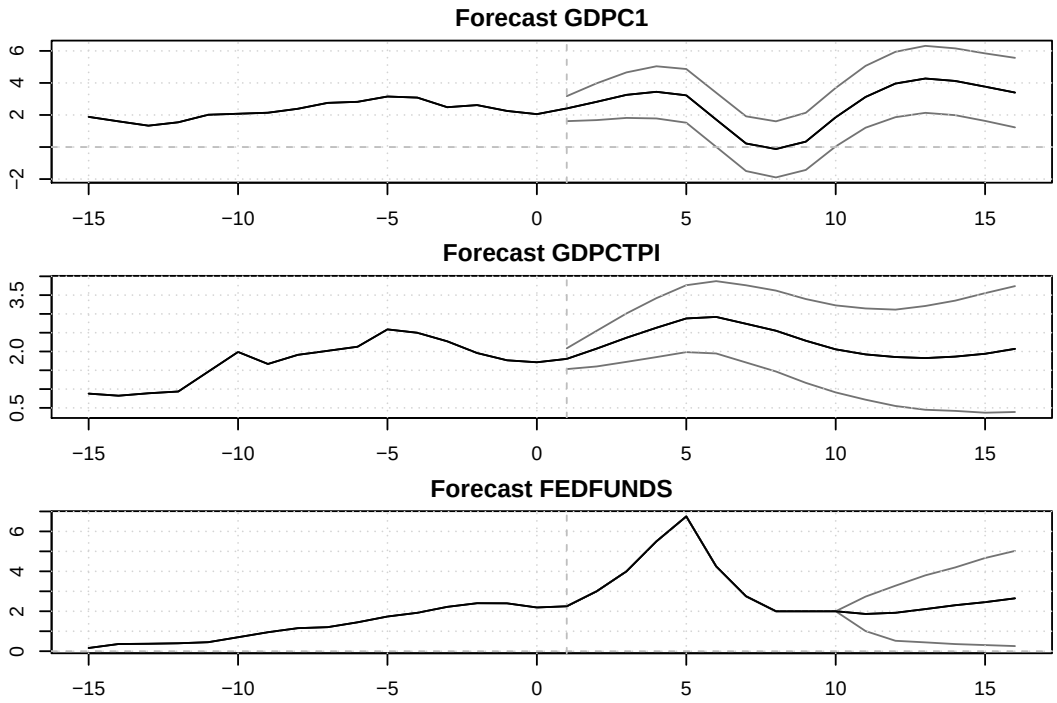

Figure 10: Conditional forecasts for GDP growth, inflation and the federal funds rate. Grey lines refer to the 68% credible set.

Affiliation:

Nikolas Kuschnig
Vienna University of Economics and Business
Institute for Ecological Economics
Global Resource Use
Welthandelsplatz 1
1020 Vienna, Austria
E-mail: nikolas.kuschnig@wu.ac.at
URL: https://kuschnig.eu/