



Federal Reserve  
Bank of Dallas

# BGVAR: Bayesian Global Vector Autoregressions with Shrinkage Priors in R

---

Maximilian Böck, Martin Feldkircher and Florian Huber

**Globalization Institute Working Paper 395**

**August 2020**

Research Department

<https://doi.org/10.24149/gwp395>

Working papers from the Federal Reserve Bank of Dallas are preliminary drafts circulated for professional comment. The views in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Dallas or the Federal Reserve System. Any errors or omissions are the responsibility of the authors.

# BGVAR: Bayesian Global Vector Autoregressions with Shrinkage Priors in R\*

Maximilian Böck<sup>†</sup>, Martin Feldkircher<sup>‡</sup> and Florian Huber<sup>§</sup>

June 24, 2020

## Abstract

This document introduces the R library **BGVAR** to estimate Bayesian global vector autoregressions (GVAR) with shrinkage priors and stochastic volatility. The Bayesian treatment of GVARs allows us to include large information sets by mitigating issues related to overfitting. This improves inference and often leads to better out-of-sample forecasts. Computational efficiency is achieved by using C++ to considerably speed up time-consuming functions. To maximize usability, the package includes numerous functions for carrying out structural inference and forecasting. These include generalized and structural impulse response functions, forecast error variance and historical decompositions as well as conditional forecasts.

**Keywords:** Global Vector Autoregressions, Bayesian inference, time series analysis, R.

**JEL Codes:** C30, C50, C87, F40.

---

\* The views in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Dallas or the Federal Reserve System. Any errors or omissions are the responsibility of the authors.

<sup>†</sup>Maximilian Böck, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria, E-mail: [maximilian.boeck@wu.ac.at](mailto:maximilian.boeck@wu.ac.at).

<sup>‡</sup>Martin Feldkircher, Oesterreichische Nationalbank, Otto-Wagner-Platz 3, 1090 Vienna, Austria, E-mail: [martin.feldkircher@oenb.at](mailto:martin.feldkircher@oenb.at) and Vienna School of International Studies, Favoritenstraße 15a, 1040 Vienna, Austria, E-mail: [martin.feldkircher@da-vienna.ac.at](mailto:martin.feldkircher@da-vienna.ac.at).

<sup>§</sup>Florian Huber, University of Salzburg, Mönchsberg 2a, 5020 Salzburg, Austria, E-mail: [florian.huber@sbg.ac.at](mailto:florian.huber@sbg.ac.at).

## 1. Introduction

Researchers in macroeconomics and finance increasingly wish to use large panel data sets. Such panels typically include a wide range of indicators per unit. And these indicators are frequently correlated within and across units. Additionally, cross-sectional heterogeneity in coefficients is often necessary. Vector autoregressions (VARs) might be adequate tools that allow for capturing rich dynamics within a panel data set.<sup>1</sup> But these models are tightly parameterized, leading to over-fitting and computational issues in large data sets. A potential solution is the global vector autoregressive (GVAR) modeling approach originally proposed in Pesaran, Schuermann, and Weiner (2004) and estimated within a Bayesian framework in Crespo Cuaresma, Feldkircher, and Huber (2016) and Feldkircher and Huber (2016).

Traditionally, these GVARs have been used in macroeconomic and financial applications (for a survey, see, Chudik and Pesaran 2016). In macroeconomic applications, interest often centers on modeling the relationship of a (potentially) large number of countries with a moderate number of endogenous variables. But unrestricted estimation of multivariate time series models such as VARs in this case is unfeasible. The GVAR approach solves the corresponding overparameterization and computation issues by assuming that relations across units (or countries in this case) are captured through cross-sectional weighted averages of other units endogenous variables. This allows for unit-by-unit estimation and reduces the number of parameters appreciably.

Estimation of GVAR models is frequently done using the classical GVAR framework of Pesaran *et al.* (2004), implemented in Matlab (Smith, L.V. and Galesi, A. 2014). In R (R Core Team 2019a), however, there exists no comparable package to estimate GVAR models easily.<sup>2</sup> To estimate Bayesian GVARs, there is no fully fledged software available. In this paper, we aim to fill this gap by proposing **BGVAR**. **BGVAR** is a highly efficient R package that entails fast and simple estimation of GVAR models within a Bayesian framework. Computational efficiency is achieved by coding up most functions in C++ using the **Rcpp** package. This enables estimating Bayesian GVAR models on a standard desktop computer in a reasonable amount of time.

In **BGVAR**, we provide recent shrinkage priors that avoid issues stemming from overparameterization and have been shown to work well for forecasting (see, e.g., Crespo Cuaresma *et al.* 2016; Dovern, Feldkircher, and Huber 2016; Huber 2016). All specifications can also be estimated with stochastic volatility (SV). Allowing for heteroscedasticity in the error variances can be particularly useful when the time period under study is volatile, leading to considerable improvements in terms of forecasting (see, e.g., Clark 2011). The Bayesian treatment of the unit-specific models also allows to handle larger models, either by including more variables per unit, a larger number of units or by increasing the number of lags.

The **BGVAR** package is designed to reduce necessary input by the researcher to a minimum. In fact, successful estimation that works for a wide range of data sets involves choosing a prior distribution and a few hyperparameters. This constitutes a significant advantage and improves usability substantially.

---

<sup>1</sup>For a recent survey on suitable methods for large cross-sectional data sets, see Feldkircher, Huber, and Pfarrhofer (2019).

<sup>2</sup>There are several R packages that allow for estimating VARs (**vars** by Pfaff (2008), and Bayesian VARs such as **bvarsv** by Krueger (2015), **bvartools** by Mohr (2019), **mfbvar** by Ankargren and Yang (2020), **BVAR** by Kuschnig and Vashold (2019) and **BHSBVAR** by (Richardson 2020)). However, to the best of our knowledge, none of these packages allows for estimating VARs with the corresponding GVAR restrictions.

To maximize its functionality, the library comes along with convenience functions such as print and plot functions that allow to summarize the output, carry out residual and model diagnostics and analyze the in-sample fit of the model. In addition to estimating a GVAR, **BGVAR** provides several functions to perform inference in multivariate time series models. These cover structural and generalized impulse responses, forecast error variance and historical decompositions as well as unconditional and conditional forecasting. Structural analysis can be done by using orthogonalization or zero and sign restrictions – the latter which can be applied either in a single unit or on the cross-section.

The remainder of the paper is structured as follows. We first provide a brief and intuitive overview of the GVAR approach in Section 2 and discuss the Bayesian priors included in the package. Section 3 introduces the main function of the package to estimate Bayesian GVARs. It also illustrates how to analyze GVAR output and perform diagnostic checks. In Section 4, we showcase the full functionality of the package, which includes structural and generalized impulse response analysis, forecast error variance and historical decompositions as well as unconditional and conditional forecasting. Finally, Section 5 concludes.

## 2. Econometric Framework

### 2.1. Model Framework and Estimation

The GVAR modeling approach consists of two stages. The first stage deals with specifying unit-specific models. The second stage combines these models to obtain a global representation of the model.

In this discussion, we remain very general and assume a panel structure where  $i = 1, \dots, N$  indicates unit  $i$  and the total number of units is  $N$ . For each  $i$ , we assume that a  $k_i$ -dimensional vector of endogenous variables  $\mathbf{y}_{it}$  follows a VAR model with  $P$  lags. To simplify notation, we assume that  $k_i = k_j$  for all  $i, j$ . These are complemented with contemporaneous and lagged exogenous regressors  $\mathbf{y}_{it}^*$ . For a pre-specified set of weights  $w_{ij}$  (which satisfy  $\sum_{j=1}^N w_{ij} = 1$  and  $w_{ii} = 0$ ), the  $\mathbf{y}_{it}^*$ 's are given by

$$\mathbf{y}_{it}^* = \sum_{j=1}^N w_{ij} \mathbf{y}_{jt}.^3$$

The resulting unit-specific model is given by:

$$\mathbf{y}_{it} = \mathbf{a}_{i0} + \mathbf{a}_{i1}t + \sum_{p=1}^P \Phi_{ip} \mathbf{y}_{it-p} + \sum_{q=0}^Q \Lambda_{iq} \mathbf{y}_{it-q}^* + \boldsymbol{\varepsilon}_{it}, \quad \boldsymbol{\varepsilon}_{it} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{it}), \quad (1)$$

where  $\mathbf{a}_{i0}$  and  $\mathbf{a}_{i1}$  denote  $k_i$ -dimensional coefficient vectors. These relate to the deterministic term of the model which we assume to consist of an intercept and a trend term.<sup>4</sup>  $\Phi_{ij}$  ( $j = 1, \dots, P$ ) denotes a  $k_i \times k_i$ -dimensional coefficient matrix associated with the lagged dependent variables.  $\Lambda_{ij}$  ( $j = 0, \dots, Q$ ) denotes a  $k_i \times k_i$  matrix of coefficients

<sup>3</sup>This implies that each element in  $\mathbf{y}_{it}^*$  is based on the same set of weights. The package allows to relax this restriction by using separate weights for different variable types.

<sup>4</sup>**BGVAR** allows for including additional deterministic components and exogenous variables.

associated with the (weakly) exogenous regressors in  $\mathbf{x}_{it}^*$  while  $\boldsymbol{\varepsilon}_{it}$  is a Gaussian error vector with zero mean and  $k_i \times k_i$ -dimensional variance-covariance matrix  $\boldsymbol{\Sigma}_{it}$ . To allow for unit-by-unit estimation, we assume that  $\boldsymbol{\varepsilon}_{it}$  and  $\boldsymbol{\varepsilon}_{jt}$  are independent for all  $i, j$  and hence  $\boldsymbol{\Sigma}_t = \text{diag}(\boldsymbol{\Sigma}_{1t}, \dots, \boldsymbol{\Sigma}_{Nt})$  is a block-diagonal matrix. This also implies that any co-movement between  $\mathbf{y}_{it}$  and  $\mathbf{y}_{jt}$  is exclusively driven by  $\mathbf{y}_{it}^*$ . And since  $\mathbf{y}_{it}^*$  is a deterministic function of the weights, this specification implies that contemporaneous relations are effectively driven by  $\boldsymbol{\Lambda}_{i0}$  and  $\mathbf{w}_i = (w_{i1}, \dots, \dots, w_{iN})'$ . The specification of the unit-specific models is completed by decomposing  $\boldsymbol{\Sigma}_{it} = \mathbf{A}_0 \mathbf{H}_t \mathbf{A}_0'$  with  $\mathbf{A}_0$  denoting a lower uni-triangular matrix and  $\mathbf{H}_t = \text{diag}(e^{h_{i1,t}}, \dots, e^{h_{ik_i,t}})$  with  $h_{ij,t}$  following an AR(1) process.

After estimating all unit-specific submodels, these can be combined using the weights  $w_{ij}$  and some straightforward algebra. Here, we omit the relevant derivations. These can be found in, e.g., Pesaran *et al.* (2004) or Feldkircher and Huber (2016). The GVAR representation is then easily obtained and given by:

$$\mathbf{y}_t = \mathbf{b}_0 + \mathbf{b}_1 t + \sum_{j=1}^{P^*} \mathbf{F}_j \mathbf{y}_{t-j} + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{et}), \quad (2)$$

with  $\mathbf{y}_t = (\mathbf{y}'_{1t}, \dots, \mathbf{y}'_{Nt})'$  is a  $k = \sum_{j=1}^N k_j$ -dimensional vector of endogenous variables and the corresponding parameters depend on the estimates of the submodels and the weights. The variance-covariance matrix  $\boldsymbol{\Sigma}_{et}$  is a full matrix. Equation (2) resembles a standard large-scale VAR model with  $P^* = \max(P, Q)$  lags.

## 2.2. Prior specification

Since the GVAR model is tightly parameterized we follow a Bayesian approach to estimation and inference. The package allows for estimating the GVAR model using three different priors. All these priors capture the notion that non-stationary variables follow a random walk process a priori. But they differ with respect to how shrinkage is introduced. In this paper, we only summarize the main features of these priors and refer to the relevant papers for more information.

To simplify prior implementation, we collect all unit-specific coefficients in a  $N_i = k_i K_i$  vector  $\boldsymbol{\Psi}_i = (\mathbf{a}_{i0}, \mathbf{a}_{i1}, \boldsymbol{\Phi}_{i1}, \dots, \boldsymbol{\Phi}_{iP}, \boldsymbol{\Lambda}_{i0}, \dots, \boldsymbol{\Lambda}_{iQ}^*)'$  (with  $K_i = k_i P + k_i^*(Q + 1) + 2$ ). Depending on the transformations of the time series in  $\mathbf{y}_{it}$ , we assume that the prior mean of  $\boldsymbol{\Phi}_{i1}$  is equal to an identity matrix (if  $\mathbf{y}_{it}$  is integrated of order one) or equal to a zero matrix (if  $\mathbf{y}_{it}$  is stationary). To capture a situation where some of the endogenous variables are stationary but persistent, the prior mean can also be centered on an AR(1) process. For the remaining coefficients, the prior mean is always set equal to zero. For further convenience, we capture these prior assumptions on the mean in a vector  $\underline{\boldsymbol{\Psi}}_i$ .

All priors on the unit-specific coefficients in  $\boldsymbol{\Psi}_i$  are Gaussian and have a prior mean equal to  $\underline{\boldsymbol{\Psi}}_i$ . However, they differ with respect to their treatment of the prior variance,  $\underline{\mathbf{V}}_i$ . More precisely, the prior on  $\boldsymbol{\Psi}_i$  is given by:

$$\boldsymbol{\Psi}_i \sim \mathcal{N}(\underline{\boldsymbol{\Psi}}_i, \underline{\mathbf{V}}_i). \quad (3)$$

Again, as stated above, we assume that  $\underline{\boldsymbol{\Psi}}_i$  is the same for all priors. The different priors differ in the precise specification of  $\underline{\mathbf{V}}_i$ .

The first prior we consider is a variant of the well-known non-conjugate Minnesota prior (for an overview, see [Koop and Korobilis 2010](#); [Koop 2013](#)). The prior variance of this setup differs across the different coefficient types in our model. First, for some equation  $j$  in unit  $i$  we assume that the prior variance on the  $k^{\text{th}}$  lag of  $x_{ij,t}$ , the  $j^{\text{th}}$  element of  $\mathbf{x}_{it}$ , is equal to  $\lambda_1^2/k^2$ .  $\lambda_1$  is a hyperparameter that is either set by the researcher or estimated. This specification implies that higher lag orders will be forced towards zero since the prior variance is decreasing in the lag length. The intuition is that the more recent past is likely to explain more variation in  $x_{ij,t}$  as compared to the more distant past.

Lags of the other endogenous variables  $x_{is,t}$  for  $s \neq j$  are treated differently. For those, the prior variance equals  $\sigma_j^2 \sigma_s^{-2} \lambda_2^2 / k^2$ . The first term serves to control for scaling differences with  $\sigma_j, \sigma_s$  denoting the OLS error standard deviations obtained by estimating univariate autoregressive models of order  $P$  and  $Q$ , respectively. The second term is similar to the prior variance we use on "own" lags of a given variable but assume that  $\lambda_1 > \lambda_2$ . This controls for the fact that lags of other variables tend to be less important than own lags of a given variable. Higher lag orders are, again, increasingly forced to zero.

The third block we consider are the coefficients on the weakly exogenous variables. For them, we specify the prior variance analogously to the case described in the previous paragraph but replace  $\lambda_2$  with  $\lambda_3$ . Finally, we set the prior variance on the deterministic part of the model equal to  $\lambda_4$ , with  $\lambda_4$  being a large value to render the prior effectively non-informative.

Since  $\lambda_1, \lambda_2$  and  $\lambda_3$  are crucial and might exert a powerful effect on the posterior estimates, we infer them from the data. Similar to [Giannone, Lenza, and Primiceri \(2015\)](#), we specify Gamma priors on  $\lambda_s (s = 1, \dots, 3)$ . These Gamma priors are set to be weakly informative. This procedure allows us to estimate the shrinkage hyperparameters alongside the remaining coefficients of the model.

The big advantage of the prior discussed above is that it effectively depends on a low number of hyperparameters. But this might be restrictive since it effectively introduces the same degree of shrinkage on, for example, the coefficients related to the contemporaneous values of  $\mathbf{x}_{it}^*$ . To allow for more flexibility and control for model uncertainty, we use the stochastic search variable selection (SSVS) prior put forward in [George and McCulloch \(1993\)](#) and subsequently introduced to the VAR literature by [George, Sun, and Ni \(2008\)](#). This prior imposes a scale mixture of two Gaussian distributions on each element in  $\Psi_i, \Psi_{ij}$ :

$$\Psi_{ij} \mid \delta_{ij} \sim (1 - \delta_{ij})\mathcal{N}(\Psi_{ij}, \tau_{0,ij}^2) + \delta_{ij}\mathcal{N}(\Psi_{ij}, \tau_{1,ij}^2), \quad \delta_{ij} \sim \text{Bernoulli}(\underline{p}), \quad (4)$$

where  $\delta_{ij}$  is a binary random variable and  $\tau_{1,ij}^2 \gg \tau_{0,ij}^2$  denote prior scaling parameters. If  $\delta_{ij} = 1$ , the prior variance is  $\tau_{1,ij}^2$ . This will be set to a large value, implying that relatively little information is introduced through the prior. By contrast, if  $\delta_{ij} = 0$ , the prior variance equals  $\tau_{0,ij}^2$  which will be very close to zero. In this case, the prior dominates the likelihood and the resulting posterior estimate will be close to zero (i.e. the  $j^{\text{th}}$  regressor will be excluded). [George et al. \(2008\)](#) modify this approach by setting  $\tau_{0,ij}^2 = c_0 \hat{\sigma}_{\psi_{ij}}^2$  and  $\tau_{1,ij}^2 = c_1 \hat{\sigma}_{\psi_{ij}}^2$ . Here, we let  $\hat{\sigma}_{\psi_{ij}}^2$  denote the OLS variance of  $\psi_{ij}$  and  $c_0$  is set to a number close to zero and  $c_1$  will be large. Good standard choices are  $c_0 = 0.1$  and  $c_1 = 10$ . Finally, we set the prior inclusion probability  $\underline{p} = 0.5$ . This implies that all covariates are equally likely to enter the model a priori.

Spike and slab priors such as the SSVS prior have excellent theoretical properties ([Bhattacharya, Pati, Pillai, and Dunson 2015](#)). However, in large models, estimating the discrete

indicators becomes an issue since any algorithm has to explore a vast model space of dimension  $2^{K_i}$ . Since  $K_i$  is typically large, this is clearly unfeasible and even carefully constructed MCMC algorithms might run into mixing issues. As a solution, one can approximate the behavior of a spike and slab prior by specifying an absolutely continuous shrinkage prior. Here, we use the Normal-Gamma (NG) global local shrinkage prior originally proposed in Griffin, Brown *et al.* (2010) and applied to the VAR context in Huber and Feldkircher (2019). The NG prior is given by:

$$\Psi_{ij} \mid \lambda_i^2, \theta_{ij} \sim \mathcal{N}(\underline{\Psi}_{ij}, 2\lambda_i^{-2}\theta_{ij}), \quad \theta_{ij} \sim G(\tau_\theta, \tau_\theta), \quad \lambda_i^2 \sim \mathcal{G}(d_\lambda, e_\lambda). \quad (5)$$

$\tau_{ij}$  denotes the *local* shrinkage parameter that is coefficient specific and  $\lambda_i$  is a *global* shrinkage term that pulls all elements in  $\Psi_i$  towards zero. This can be viewed as a common scaling factor with the  $\theta_{ij}$  allowing for coefficient-specific deviations in light of a large value of  $\lambda_i^2$ . On both the global and local parameters, we impose Gamma distributed priors with hyperparameters  $\tau_\theta, d_\lambda, e_\lambda$ .  $\tau_\theta$  controls the tail behavior of the prior with small values placing more prior mass on zero and leading to heavier tails. The remaining two hyperparameters  $d_\lambda$  and  $e_\lambda$  control the amount of global shrinkage with small values (i.e. of order 0.01) leading to heavy shrinkage towards the origin.

The key feature of this prior is that if the local scaling parameters are integrated out, the corresponding marginal prior is heavy tailed and effectively allows for separating signals and noise. Huber and Feldkircher (2019) show that this prior works well for US macroeconomic data and modify it to also capture features of the Minnesota prior such as increased penalty for higher lag lengths.

This concludes the prior setup for the coefficients. For the remaining quantities in the model we use standard priors in the literature that work reasonably well for a wide range of data sets. More precisely, we use the prior setup proposed in Kastner and Frühwirth-Schnatter (2014) on the coefficients in the log-volatility state equation.<sup>5</sup> Since we sample the free elements in  $\mathbf{A}_0$  using the algorithm outlined in Carriero, Clark, and Marcellino (2019), we can use a NG prior similar to the one discussed above.

### 3. Implementation: The BGVAR package

#### 3.1. Overview

The **BGVAR** library allows to estimate Bayesian Global Vector Autoregressions in R (R Core Team 2019b). To maximize its functionality, the library comes along with various tool functions to carry out predictions, impulse response functions, (generalized) forecast error variance decompositions, historical decompositions and conditional forecasts. Each function has its own class and helper functions are provided to summarize and plot the output. The classes and an overview of all functions contained in the package are presented in Table 1.

---

<sup>5</sup>This corresponds to using a Gaussian prior with zero mean and variance 10 on the unconditional mean of the log-volatility, a Beta prior on the persistence parameter  $\rho$ ,  $(\rho + 1)/2 \sim \mathcal{B}(25, 5)$  and a non-conjugate Gamma prior on the error variance of the log-volatility process  $\sigma_h^2 \sim \mathcal{G}(1/2, 1/2)$ . In case we do not use SV, we use weakly informative inverted Gamma priors on the innovation variances.

Table 1: Functionality of **BGVAR**.

Function	Class	Description
<i>Estimation function</i>		
<code>bgvar()</code>	<code>bgvar</code>	Estimates the Bayesian Global Vector Autoregression.
<i>BGVAR tool functions</i>		
<code>IRF()</code>	<code>bgvar.irf</code>	Computes impulse response functions offering various identification procedures by using an object fitted by <code>bgvar</code> .
<code>IRF.cf()</code>	<code>bgvar.irf</code>	Computes counterfactual analysis of an object fitted by <code>IRF</code> .
<code>predict()</code>	<code>bgvar.pred</code>	Computes predictions of an object fitted by <code>bgvar</code> .
<code>fevd.decomp()</code>	<code>bgvar.fevd</code>	Computes forecast error variance decompositions of an object fitted by <code>bgvar</code> .
<code>gfevd.decomp()</code>	<code>bgvar.fevd</code>	Computes generalized forecast error variance decompositions of an object fitted by <code>bgvar</code> .
<code>hd.decomp()</code>	<code>bgvar.hd</code>	Computes historical decompositions of an object fitted by <code>bgvar</code> .
<code>cond.pred()</code>	<code>bgvar.pred</code>	Computes conditional forecasts of an object fitted by <code>bgvar</code> and <code>predict.bgvar</code> .
<i>Helper functions</i>		
<code>AIC()</code>		Computes Akaike information criterion.
<code>avg.pair.cc()</code>		Average pairwise cross-sectional correlations of either the data or the unit-specific submodels' residuals.
<code>BIC()</code>		Computes Bayesian information criterion.
<code>coef()</code>		Extracts global model coefficients for certain posterior quantiles.
<code>coefficients()</code>		Extracts global model coefficients for certain posterior quantiles.
<code>conv.diag()</code>		Computes Geweke's convergence diagnostic.
<code>DIC()</code>		Computes Deviance information criterion.
<code>fitted()</code>		Extracts model fitted values.
<code>logLik()</code>		Computes log-likelihood of the model.
<code>lps()</code>		Computes log-predictive scores of an object of class <code>bgvar.pred</code> .
<code>matrix_to_list()</code>		Converts an input matrix to a list suitable for estimation with <code>bgvar</code> .
<code>plot()</code>		Plotting function available for class <code>bgvar</code> , <code>bgvar.resid</code> , <code>bgvar.irf</code> , <code>bgvar.fevd</code> .
<code>print()</code>		Summarizes model specification specified in <code>bgvar</code> .
<code>resid()</code>	<code>bgvar.resid</code>	Extract residuals of an object fitted with <code>bgvar</code> .
<code>residuals()</code>	<code>bgvar.resid</code>	Extract residuals of an object fitted with <code>bgvar</code> .
<code>residual.corr.test()</code>		Performs an F-test for serial autocorrelation in the residuals.
<code>rmse()</code>		Computes root mean squared error of an object of class <code>bgvar.pred</code> .
<code>summary()</code>		Computes summary and diagnostic statistics for <code>bgvar</code> .
<code>vcov()</code>		Extracts global variance covariance matrix.

The library depends on well established R packages. Most importantly, and to speed up computational time, the estimation of the unit-specific submodels in Equation 1 as well as the function to compute the global representation of the model in Equation 2 are implemented in C++. This implies a dependency on **Rcpp** (Eddelbuettel and François 2011) and **RcppArmadillo** (Eddelbuettel and Sanderson 2014). To further speed up computation, **BGVAR** allows the use of parallel computation for the estimation of the unit-specific submodels as well as the

sampling of rotation matrices when performing impulse response analysis identified through sign restrictions. This implies that **BGVAR** relies on **doParallel** (Microsoft Corporation and Weston 2019a) and **foreach** (Microsoft Corporation and Weston 2019b). Stochastic volatility is implemented using the **stochvol** package of Kastner (2016).<sup>6</sup>

### 3.2. Estimation

In this section, we start describing the data requirements and the main function `bgvar()` which allows to estimate Bayesian GVARs using three different prior set-ups and various specifications. To illustrate the function, we use the `eerData` data set of Feldkircher and Huber (2016), which contains 76 quarterly observations for 43 countries and the period from 1995Q1 to 2013Q4. The units in the following example hence refer to countries. The euro area (EA) is included as a regional aggregate.<sup>7</sup>

To get started, we load the **BGVAR** package and the data set by typing:

```
R> set.seed(571)
R> library("BGVAR")
R> data("eerData")
```

As a bare minimum, the estimation function needs two inputs from the researcher, a data set (`Data`) and a weight matrix (`W`). With the example data set, this implies `Data=eerData` and `W=W.0012`. In general, data can be provided in two forms. The first, as in this example, is to provide a list object of length  $N$  (i.e., the number of countries / units) with each slot containing a  $T \times k_i$  data matrix. The names of the data list should correspond to the countries / units and the `colnames` in a given slot of `Data` denote the variable names.

```
R> names(eerData)

[1] "EA" "US" "UK" "JP" "CN" "CZ" "HU" "PL" "SI" "SK" "BG" "RO" "EE"
[14] "LT" "LV" "HR" "AL" "RS" "RU" "UA" "BY" "GE" "AR" "BR" "CL" "MX"
[27] "PE" "KR" "PH" "SG" "TH" "IN" "ID" "MY" "AU" "NZ" "TR" "CA" "CH"
[40] "NO" "SE" "DK" "IS"
```

```
R> colnames(eerData$US)

[1] "y" "Dp" "rer" "stir" "ltir" "tb" "poil"
```

The library uses one important naming convention, which is that neither the unit names of `Data` nor the variable names should contain a dot (hence use `US` instead of `U.S.`). It is also possible to submit the data in form of a  $T \times k$  data matrix, with  $k = \sum_{i=1}^N k_i$  denoting the sum of endogenous variables in the system. The above mentioned naming convention implies that

<sup>6</sup>Further dependencies are **mvtnorm** (Genz, Bretz, Miwa, Mi, Leisch, Scheipl, and Hothorn 2020), **GIGrvg** (Leydold and Hormann 2017) to simulate from the conditional posterior distributions, packages **abind** (Plate and Heiberger 2016), **MASS** (Venables and Ripley 2002) and **Matrix** (Bates and Maechler 2019) for matrix manipulation and **coda** and **bayesm** (Rossi 2019) to inspect posterior output.

<sup>7</sup>For further examples with other datasets and showing additional possible specifications, we refer to the vignette of the **BGVAR** package (accessible here: <https://cran.r-project.org/web/packages/BGVAR/>).

the column names of `Data` have to include the name of the unit (country) and the variable name, separated by a dot (i.e., `US.gdp` to denote GDP in the US country model). The  $N \times N$  weight matrix `W` should be row-standardized and the diagonal elements should be zero.

Using these data, we can now estimate a toy model as follows:

```
R> model.1<-bgvar(Data=eerData,
+                 W=W.trade0012,
+                 saves=500,
+                 burns=500,
+                 plag=1,
+                 prior="NG",
+                 hyperpara=NULL,
+                 SV=TRUE,
+                 thin=1,
+                 trend=TRUE,
+                 eigen=1.05,
+                 h=8,
+                 save.country.store=FALSE,
+                 multithread=FALSE
+                 )
```

Start estimation of Bayesian Global Vector Autoregression.

Prior: Normal-Gamma prior.

Lag order: 1

Stochastic volatility: enabled.

Thinning factor: 1. This means every draw is saved.

Hyperparameter setup:

No hyperparameters are chosen, default setting applied.

Estimation of country models starts... took 1 min 52 seconds.

Start stacking:

Stacking finished.

Needed time for estimation of bgvar: 2 mins 1 second.

The default prior specification in **BGVAR** is to use the NG prior with stochastic volatility and one lag for both the endogenous and weakly exogenous variables (`plag=1`). The setting `hyperpara=NULL` implies that we use the standard hyperparameter specification for the NG prior; see the help files for more details. The slots `saves` and `burns` denote the posterior draws and the burn-in draws (i.e., the draws that are discarded). To ensure that the MCMC estimation has converged, a high-number of burn-ins is recommended (say 15,000 to 30,000). Saving the full set of posterior draws can eat up a lot of storage. To reduce this, we can use a thinning interval which stores only each  $\text{thin}^{\text{th}}$  draw of the global posterior output. For example, with `thin=10` and `saves=5000` posterior draws, the amount of MCMC draws stored

is 500. `TREND=TRUE` implies that the model is estimated using a trend. Note that regardless of the trend specification, each equation always automatically includes an intercept term. To speed up computation, it is possible to set `multithread=TRUE`, which invokes parallel computing using the `doParallel` package. The package then attempts to detect the number of available CPU cores which it can use for parallel computing. Ideally this number would be equal to the number of units  $N$ . In addition to storing the global posterior, we could also be interested in inspecting the output of the  $N$  unit-specific submodels in more detail. To do so, we could set `save.country.store=TRUE` which implies storing the whole posterior distribution of each single unit-specific submodel. Due to storage reasons, the default is set to `FALSE` and only the posterior medians of the submodels are reported. Note that `thin` also applies to `save.country.store=TRUE`.

The above model has been estimated with stochastic volatility (`SV=TRUE`). There are several reasons why one may want to let the residual variances change over time. First and foremost, data frequently used in macroeconometrics contain volatile periods, such as severe recessions and gradual recoveries. Hence accounting for time variation can considerably improve the fit of the model (Primiceri 2005; Sims and Zha 2006; Doornik *et al.* 2016; Huber 2016). Second, the specification implemented in this library nests the homoskedastic case. Note that in case the model has been estimated with stochastic volatility, the variance covariance matrix that is used for structural analysis corresponds to the one with median volatilities (calculated over the sample period) on its diagonal.<sup>8</sup> The volatilities of the first equation ( $y$ ) in the euro area country model can be accessed by:

```
R> stoch.vol.y<-model.1$cc.results$sig$EA[,"EA.y","EA.y"]
```

To discard explosive draws, we can compute the eigenvalues of the coefficient matrix  $F$  of the reduced form of the global model, written in its companion form. Note that this can only be done once the single models have been estimated and stacked (and hence not directly built into the MCMC algorithm for the unit-specific submodels). To discard draws that lead to higher eigenvalues than 1.05, set `eigen=1.05`. Last, note that since we are also going to demonstrate forecasting and forecast evaluation, we have set `h=8`. This implies that the last 8 observations are reserved as a hold-out sample and hence not used for estimation. The default setting is `h=0` which implies to use the full sample period.

### 3.3. Model diagnostics and output analysis

Having estimated the model, we can summarize the outcome in various ways. As such, we could invoke the print method `print(model.1)`. For the sake of brevity, we do not show the lengthy output of the print function in this document. What it does is to print the arguments submitted to **BGVAR** along with the model specification for each unit. For the latter, asterisks indicate weakly exogenous variables, double asterisks exogenous variables and variables without asterisks the endogenous variables per unit.

The `summary` method is a more enhanced way to analyze the output. It computes descriptive statistics like convergence properties of the MCMC chain, serial autocorrelation in the errors and the average pairwise autocorrelation of cross-unit residuals.

---

<sup>8</sup>Alternatively, one would have  $T$  variance covariance matrices and hence  $T$  impulse responses for each variable. Since the size of the shock (i.e., the residual variance) varies over time, the resulting impulses would be typically either up- or down-scaled, whereas the shapes of the impulse response functions are not affected.

```
R> summary(model.1)
```

```
-----
```

```
Model Info:
```

```
Prior: NG
```

```
Nr. of lags: 1
```

```
Nr. of posterior draws: 500/1=500
```

```
Number of stable posterior draws: 46
```

```
Number of countries: 43
```

```
-----
```

```
Convergence diagnostics
```

```
Geweke statistic: 1578 out of 10580 variables' z-values exceed the 1.96 threshold (14.91%)
```

```
-----
```

```
Global Likelihood: 41425.76
```

```
F-test, first order serial autocorrelation of cross-country residuals
```

```
Summary statistics:
```

	# p-values in %	
>0.1	"132"	"57.39%"
0.05-0.1	"21"	"9.13%"
0.01-0.05	"29"	"12.61%"
<0.01	"48"	"20.87%"

```
-----
```

```
Average pairwise cross-country correlation of country model residuals
```

```
Summary statistics:
```

	y	Dp	rer	stir
<0.1	"35 (81.4%)"	"43 (100%)"	"37 (92.5%)"	"42 (100%)"
0.1-0.2	"8 (18.6%)"	"0 (0%)"	"3 (7.5%)"	"0 (0%)"
0.2-0.5	"0 (0%)"	"0 (0%)"	"0 (0%)"	"0 (0%)"
>0.5	"0 (0%)"	"0 (0%)"	"0 (0%)"	"0 (0%)"

	ltir	tb
<0.1	"17 (89.47%)"	"42 (100%)"
0.1-0.2	"2 (10.53%)"	"0 (0%)"
0.2-0.5	"0 (0%)"	"0 (0%)"
>0.5	"0 (0%)"	"0 (0%)"

```
-----
```

We can now have a closer look at the information provided by `summary`. The header contains some basic information about the prior used to estimate the model, how many lags, posterior draws and countries. The next line shows Geweke's CD statistic ([Geweke 1992](#)), which is calculated using the `coda` package. Geweke's CD assesses practical convergence of the MCMC algorithm. In a nutshell, the diagnostic is based on a test for equality of the means of the first and last part of a Markov chain (by default we use the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution. The test statistic is a standard Z-score: the difference between the two sample means divided by its estimated standard error. The standard error is estimated from the spectral density at zero and so takes into account any autocorrelation. The test statistic shows that only a small fraction of all

coefficients did not convergence. Increasing the number of burn-ins can help decreasing this number further. The statistic can also be calculated by typing `conv.diag(model.1)`.

The next model statistic is the likelihood of the global model. This statistic can be used for model comparison. Next and to assess, whether there is first order serial autocorrelation present, we provide the results of a simple F-test. The table shows the share of p-values that fall into different significance categories. Since the null hypothesis is that of no serial correlation, we would like to have as many large ( $> 0.1$ ) p-values as possible. The statistics show that already with one lag, serial correlation is modest in most equations' residuals. To further decrease serial correlation in the errors, one could increase the number of lags via `plag`.

The last part of the summary output contains a statistic of cross-unit correlation of (posterior median) residuals. One assumption of the GVAR framework is that of negligible, cross-unit correlation of the residuals. Significant correlations prohibit structural and spillover analysis (Dees, di Mauro, Pesaran, and Smith 2007). In this example, correlation is reasonably small. Some other useful methods the library offers include the `coef` (or `coefficients` as its alias) methods to extract the  $k \times k \times plag$  matrix of reduced form coefficients of the global model. Via the `vcov` command, we can access the global variance covariance matrix and the `logLik` function allows us to gather the global log likelihood.

```
R> Fmat <- coef(model.1)
R> Smat <- vcov(model.1)
R> lik <- logLik(model.1)
```

Last, we can have a quick look at the in-sample fit using either the posterior median of the country models' residuals (`global=FALSE`) or those of the global solution of the GVAR (`global=TRUE`). The in-sample fit can also be extracted by using `fitted()`.

In [Figure 1](#) we show the in-sample fit of the euro area model (`global=FALSE`).

## 4. Extensions: Structural analysis and predictions

In this section, we illustrate functions of **BGVAR** to perform structural analysis and forecasting. Further technical details can be found in textbooks such as [Kilian and Lütkepohl \(2017\)](#).

### 4.1. Impulse response analysis

One of the main applications of a GVAR is to assess the domestic and international impulse responses to a structural / identified macroeconomic shock. To do so, we can invoke the function `IRF()` which can be used to either calculate generalized impulse response functions (GIRFs) as in [Pesaran and Shin \(1998\)](#), orthogonalized impulse response functions using a Cholesky decomposition or impulse response functions given a set of user-specified sign restrictions.

We provide two examples of the functionality of `IRF`, one where we make use of the Cholesky decomposition for identification and one where we apply sign-restrictions to identify the model. In both schemes we follow the bulk of the GVAR literature ([Dees et al. 2007](#); [Eickmeier](#)

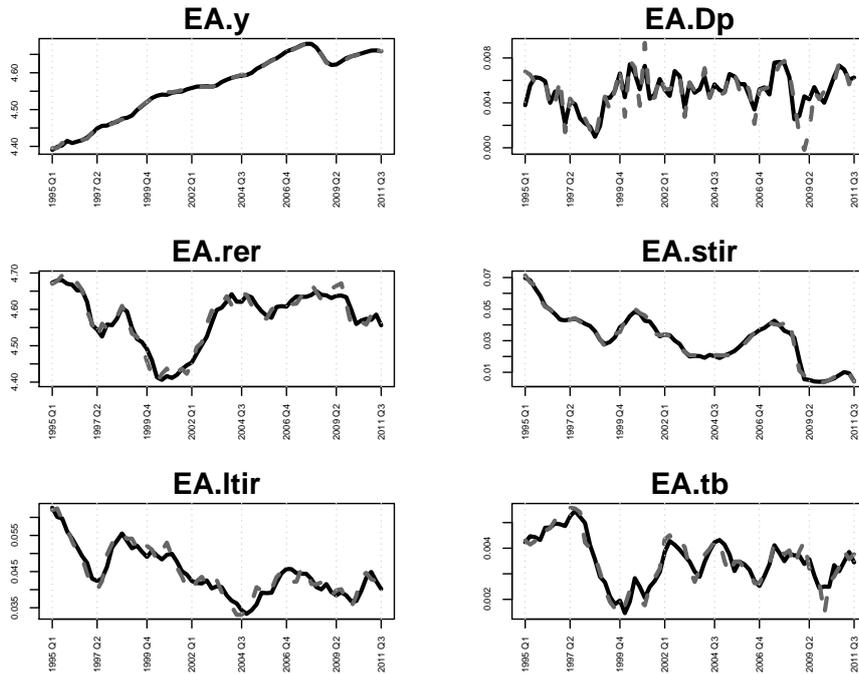


Figure 1: In-sample fit for euro area variables.

and Ng 2015; Feldkircher and Huber 2016) and identify the model only *locally*. This translates into applying the Cholesky decomposition or sign-restrictions only in the unit (country) of interest. For example, to identify a US monetary policy shock, it suffices to apply the identification scheme under consideration to  $\Sigma_{0t}$  (with  $i = 0$  denoting the US).

More specifically, local identification in model  $i$  – omitting the deterministic terms – boils down to

$$\mathbf{Q}_0^{-1} \mathbf{y}_{0t} = \mathbf{Q}_0^{-1} \sum_{j=1}^p \Phi_{0j} \mathbf{y}_{0,t-j} + \mathbf{Q}_0^{-1} \sum_{j=0}^{p^*} \Lambda_{ij} \mathbf{y}_{i,t-j} + \underbrace{\mathbf{Q}_0^{-1} \boldsymbol{\varepsilon}_{0t}}_{:= \mathbf{v}_{0t}},$$

with  $\mathbf{Q}_0^{-1}$  being the lower Cholesky factor of  $\Sigma_{0t}$ . To construct the corresponding global representation of the GVAR with a locally identified shock, we have to set up a matrix  $\mathbf{Q}$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_0 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{k_1} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{I}_{k_N} \end{pmatrix}.$$

The global vector of structural errors is then obtained by  $\mathbf{Q}^{-1} \boldsymbol{\varepsilon}_t = (\mathbf{v}'_{0t}, \boldsymbol{\varepsilon}'_{1t}, \dots, \boldsymbol{\varepsilon}'_{1t})'$ . Hence, the local identification yields orthogonal residuals in country model  $i$ , whereas the remaining residuals are potentially correlated, which implies that spillovers are of the generalized type. Identification via sign restrictions follows in the same fashion with the exception that instead of  $\mathbf{Q}_0$  we use  $\mathbf{Q}_0 \mathbf{R}_0$  with  $\mathbf{R}_0$  denoting a  $k_0 \times k_0$  rotation matrix. The rotation matrix has to

be sampled either using the algorithm of Rubio-Ramirez, Waggoner, and Zha (2010) or – in case also zero restrictions are placed – the algorithm of Arias, Rubio-Ramírez, and Waggoner (2018). **BGVAR** also allows to place sign restrictions on the cross-section or to examine responses to a joint / regional shock. This implies that the identity matrices on the diagonal of  $Q$  are replaced by suitable  $Q_i / Q_i R_i$  matrices.

We start illustrating the functionality of IRF by using a Cholesky decomposition to identify a US monetary policy shock. For that purpose, we have to set up a list `shock` that contains information about which variable we want to shock `shocks$var`, in which country `shocks$cN`, the forecast horizon (`nhor`) and whether we want a recursive identification, GIRFs or sign restrictions (`shocks$ident`). We can also scale the shock using `shocks$scal`. In the example below, we want the impulse responses calculated over 24 time periods (`nhor=24`) and decide to store the full posterior of the responses (`save.store=TRUE`).

```
R> shocks<-list();shocks$var="stir";shocks$cN<-"US"
R> shocks$ident="chol";shocks$scal=-0.5 # 50 bps
R> irf.chol.us.mp<-IRF(obj=model.1,shock=shocks,nhor=24,save.store=TRUE)
```

Start computing impulse response functions of Bayesian Global Vector Autoregression.

```
Identification scheme: Short-run identification via Cholesky decomposition.
Structural shock of interest: US.stir.
Start impulse response analysis on single core (46 stable draws in total).
```

```
Impulse response analysis took 0 mins 18 seconds.
```

```
Size of IRF object: 4.2 Mb
Needed time for impulse response analysis: 0 mins 27 seconds.
```

The results are stored in `irf.chol.us.mp$posterior`, which is a  $K \times nhor \times nr.ofshocks \times 7$  object. The last slot contains the 50%, 68% and 95% credible intervals along the posterior median. If `save.store=TRUE`, set of impulse response draws can be accessed by `irf.chol.us.mp$IRF_store`.

We can plot the set of all responses of a particular unit by typing:

```
R> plot(irf.chol.us.mp,resp="US")
```

The plot shows the posterior median response (solid, black line) along 50% (dark grey) and 68% (light grey) credible intervals.

In the next example, we show how to identify an aggregate demand and supply shock in the US country model using sign restrictions. A positive shock to aggregate demand is characterized by a parallel upward movement of output (`y`) and inflation. The positive supply shocks, by contrast, leads to a decrease in output. The opposite movements of output hence separate the two shocks. The signs are summarized in [Table 2](#).

The restrictions in [Table 2](#) can be passed on to IRF by setting up a list. In the example below we specify `sign.constr.eer` which has to contain the variable to shock (`shock`), which responses to restrict (`restrictions`), the direction of the restrictions (`sign`) and the horizon

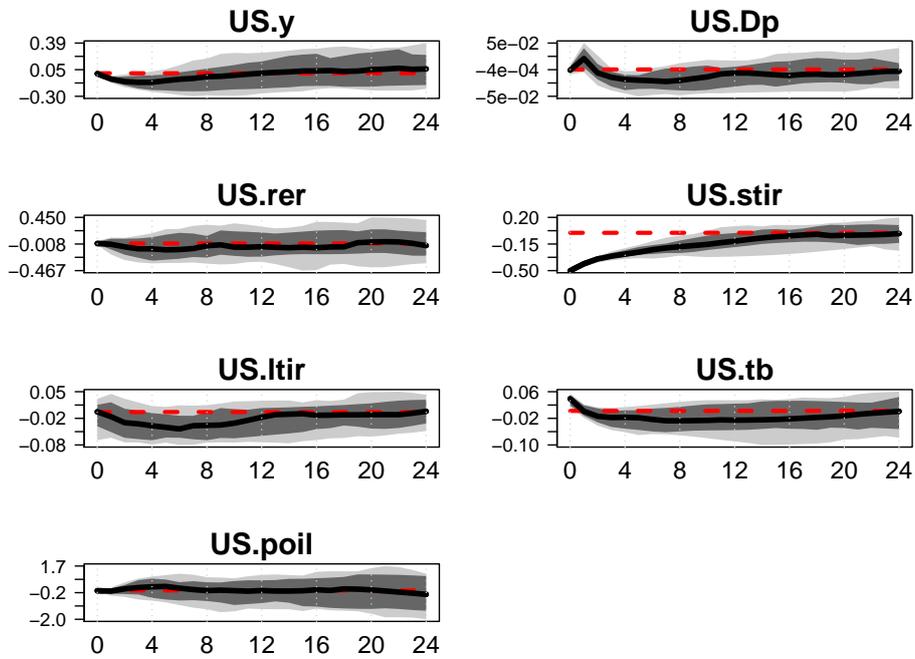


Figure 2: Impulse response functions in the US country model (Cholesky identification).

Shock	y	Dp
Aggregate demand	↑ (shock)	↑ (restriction)
Aggregate supply	↓ (restriction)	↑ (shock)

Table 2: Sign restrictions for the US country model.

how long these restrictions should hold (`rest.horz`). The maximum number of rotation matrices sampled per MCMC draw before we jump to the next draw can be specified by `MaxTries`. Increasing the number of restrictions (on the variables or the horizon) will lead to preciser inference; however, finding a suitable rotation matrix will become substantially harder in which case we want to increase `MaxTries`.

```
R> sign.constr.eer<-list()
R> sign.constr.eer$shock1$shock<-"US.y" #Positive AD Shock, GDP up,
R> sign.constr.eer$shock1$restrictions$res1<-"US.Dp" #infl. and int. rates up
R> sign.constr.eer$shock1$sign<-c(">", ">")
R> sign.constr.eer$shock1$rest.horz<-c(1,1)
R> sign.constr.eer$shock1$constr<-c(1,1) # no cc restr., set constr. to 1
R> sign.constr.eer$shock1$scal<-1 #+1% increase
R> sign.constr.eer$shock2$shock<-"US.Dp" # Negative AS shock, infl. up
R> sign.constr.eer$shock2$restrictions/res1<-"US.y"
R> sign.constr.eer$shock2$sign<-c(">", "<")
R> sign.constr.eer$shock2$rest.horz<-c(1,1)
R> sign.constr.eer$shock2$constr<-c(1,1) # no cc restr., set constr. to 1
R> sign.constr.eer$shock2$scal<-1 # +1% increase
R> names(sign.constr.eer)<-c("AD", "AS")
R> sign.constr.eer$MaxTries<-10000 # maximum number of rotations per draw
```

With the information from above, we can then proceed by invoking the IRF command to actually compute the impulse responses. By default, we use one CPU core (`multithread=FALSE`) and do not store the full set of responses (`save.store=FALSE`).

```
R> irf.sign<-IRF(obj=model.1,nhor=24,sign.constr=sign.constr.eer,save.store=TRUE)
```

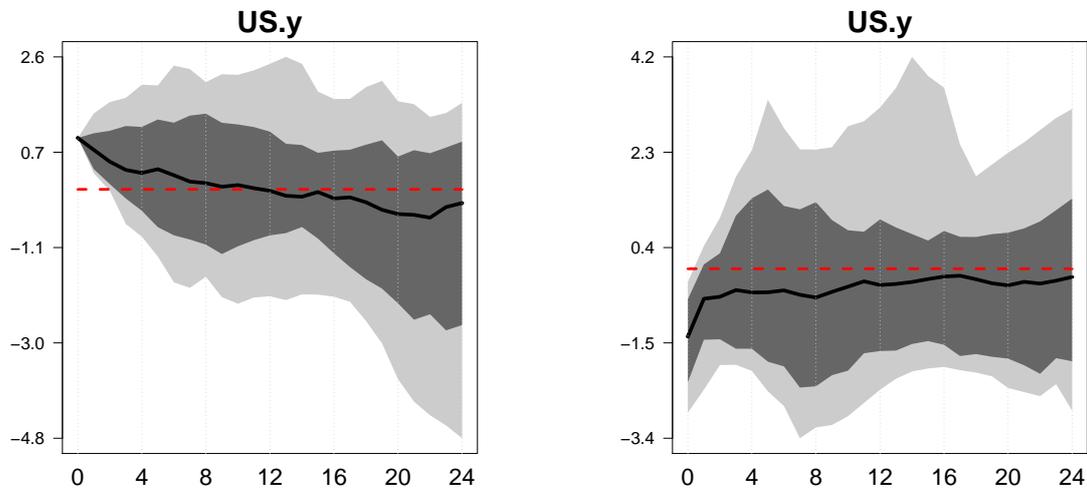
We can infer the number of successful rotation matrices by looking at `irf.sign$rot.nr`. To verify the sign restrictions visually, we can plot the respective responses and display them in [Figure 3](#).

```
R> plot(irf.sign,resp="US.y",shock.nr=1)
```

```
R> plot(irf.sign,resp="US.y",shock.nr=2)
```

## 4.2. Generalized forecast error variance decomposition (GFFEVD)

A forecast error variance decomposition indicates the amount of information each variable contributes to the other variables in the autoregression. It is calculated by examining how much of the forecast error variance of each of the variables can be explained by shocks to the other variables. In a system with fully orthogonalized errors, the shares of FEVD sum up to 1. In the GVAR context, however, and since we identify a shock only locally in a particular unit-specific submodel we typically have a certain degree of cross-unit residual correlation left. Consequently, the shares of the FEVD typically exceed unity. By contrast, a fully



(a) Response to aggregate demand shock (shock.nr=1).

(b) Response to an aggregate supply shock (shock.nr=2).

Figure 3: Responses of US output to positive aggregate supply and demand shocks.

orthogonalized system would yield shares that sum up to unity but inherits assumptions that are likely hard to defend for all units contained in the system.

One way of fixing this is to use a generalized forecast error variance decomposition. Like GIRFs, these are independent of the ordering but, since the shocks are not orthogonalized, shares of explained forecast error variance exceed unity. Recently, [Lanne and Nyberg \(2016\)](#) proposed a way of scaling the GFEVDs, which has the convenient property of the shares summing up to 1. Also, the results are independent of the ordering of the variables in the system. The Lanne-Nyberg corrected GFEVDs can be calculated using the `GFEVD.LN` command. We can either use a running mean (`running=TRUE`) or the full set of posterior draws. The latter is computationally very expensive.

```
R> gfevd.us.mp=gfevd.decomp(obj=model.1,nhor=24,
+                           running=TRUE,multithread=TRUE)$GFEVD
```

Start computing generalized forecast error variance decomposition of Bayesian Global Vector

Start computation on 8 cores (46 stable draws in total).

Size of IRF object: 9.9 Mb

Needed time for computation: 0 mins 9 seconds.

```
R> idx<-which(grepl("EA.",dimnames(gfevd.us.mp)[[2]]))
R> own<-colSums(gfevd.us.mp["EA.y",idx,])
R> foreign<-colSums(gfevd.us.mp["EA.y",-idx,])
R> par(mar=c(2.5,2.5,5,1))
R> barplot(t(cbind(own,foreign)),legend.text=c("own","foreign"),
```

```
+ args.legend=list(cex=2,x = "top", bty="n",ncol=2,
+ inset=c(-0.50,-0.25), xpd = TRUE))
```

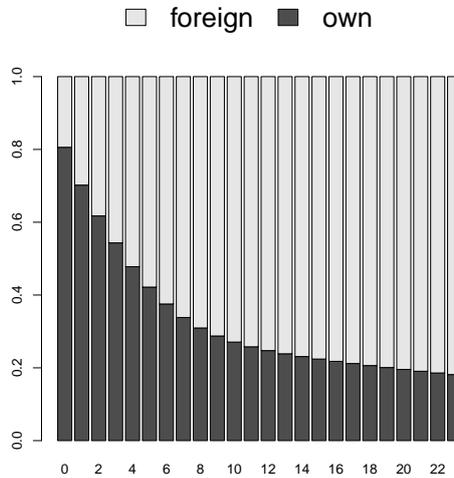


Figure 4: Shares of GFEVD explained by own and foreign variables.

In case we want to focus on a single unit in which we have fully identified all shocks, we can compute a simple forecast error variance decomposition (FEVD). This can be done by using the command `fevd.decomp`. Since the computation is very time consuming, the FEVDs are based on the posterior median only (as opposed to calculating FEVDs for each MCMC draw or using a running mean). In case the underlying shock has been identified via sign restrictions, the corresponding rotation matrix is the one that fulfills the sign restrictions at the point estimate of the posterior median of the reduced form coefficients (stored in `irf.obj$struc.obj$Rmed`). Alternatively one can submit a rotation matrix using the option `R` in `fevd.decomp`.

### 4.3. Historical decomposition

A historical decomposition allows to examine the relative importance of structural shocks in explaining deviations of a time series from its unconditional mean. This can be used to assess the hypothetical question of how data would have looked like if it was driven only by a particular structural shock (e.g., monetary policy shock) or a combination of structural shocks. It can be calculated using the function `hd.decomp`. The function also allows to compute the structural error of the model. To save computational time as well as due to storage limits, we use the point estimate of the posterior median (as opposed to calculating HDs and the structural error for each draw of the MCMC chain). In case the shock has been identified via sign restrictions, a rotation matrix has to be selected. If not specified otherwise (via `R`), the rotation matrix based on the posterior median of the reduced form coefficients (`irf.obj$struc.obj$Rmed`) will be used.

```
R> HD<-hd.decomp(irf.chol.us.mp)
```

Start computing historical decomposition of Bayesian Global Vector Autoregression.

```
Start computing HDs...
Size of object: 27.8 Mb
Needed time for computation: 0 mins 4 seconds.
```

#### 4.4. Unconditional forecasts

Since the GVAR framework was developed to capture cross-sectional dependencies, it can handle a rich set of dynamics. This can also be useful for forecasting global components (e.g., global output) or unit-specific variables, controlling for global factors. [Pesaran, Schuermann, and Smith \(2009\)](#) show that the GVAR yields competitive forecasts for a range of macroeconomic and financial variables. [Crespo Cuaresma \*et al.\* \(2016\)](#) demonstrate that Bayesian shrinkage priors can help improving GVAR forecasts and [Dovern \*et al.\* \(2016\)](#) and [Huber \(2016\)](#) yield evidence for further gains in forecast performance by using GVARs with stochastic volatility.

To compute forecasts with the **BGVAR** package, we use the command `predict`. To be able to evaluate the forecast, we have to specify the size of the hold-out sample when estimating the model. Here, `model.1` was estimated choosing a hold-out-sample of 8 observations (`h=8`). The 8-steps ahead forecasts (`fhorz=8`) can then be calculated by:

```
R> fcast <- predict(model.1, fhorz=8,save.store=TRUE)
```

The values for the forecasts are stored in `fcast$fcast` which contains also the credible intervals of the predictive posterior distribution. We can evaluate the forecasts with the retained observations looking at the root mean squared errors (RMSEs) or log-predictive scores (LPS).

```
R> lps.h8 <- lps(fcast)
R> rmse.h8 <- rmse(fcast)
```

The objects `lps.h8` and `rmse.h8` then each contain a  $8 \times k$  matrix with the LPS scores / RMSEs for each variable in the system over the forecast horizon.

Last, we can visualize the forecasts by typing

```
R> plot(fcast, resp="US.Dp", Cut=8)
```

with `Cut` denoting the number of realized data points that should be shown in the plot prior the forecasts start.

#### 4.5. Conditional forecasts

Similar to structural analysis, it is possible to use conditional forecasts, identified in a unit-specific submodel. For that purpose, we use the methodology outlined in [Waggoner and Zha \(1999\)](#) and applied in [Feldkircher, Huber, and Moder \(2015\)](#) in the GVAR context. The following lines set up a conditional forecast fixing US inflation for 5 periods to its last observed value in the sample.

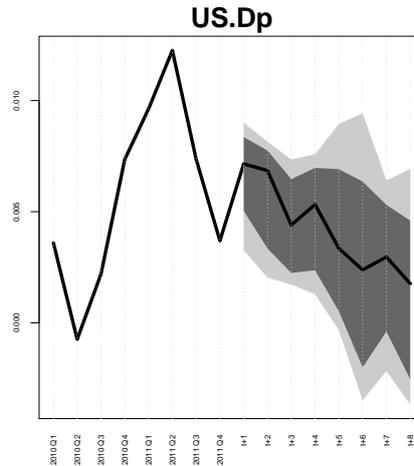


Figure 5: Unconditional forecast of US inflation, 8 periods ahead.

```
R> # matrix with constraints
R> constr <- matrix(NA,nrow=fcast$fhorz,ncol=ncol(model.1$xglobal))
R> colnames(constr) <- colnames(model.1$xglobal)
R> # set "US.Dp" for five periods on its last value
R> constr[1:5,"US.Dp"] <- model.1$xglobal[nrow(model.1$xglobal),"US.Dp"]
R> # compute conditional forecast (hard restriction)
R> cond_fcast <- cond.predict(constr=constr, bgvar.obj=model.1,
+                             pred.obj=fcast, constr_sd=NULL)
```

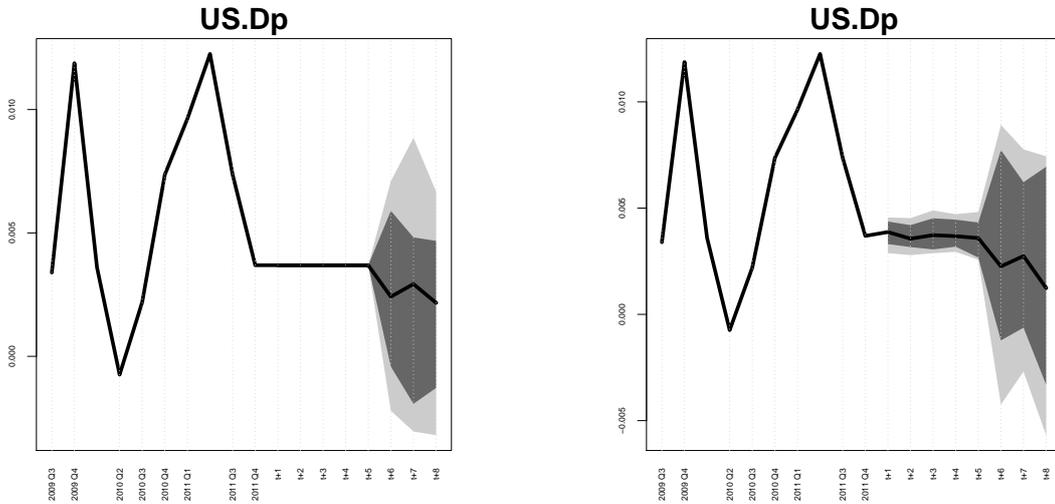
We could impose the same restrictions as "soft conditions" accounting for uncertainty by drawing from a Gaussian distribution with the conditional forecast in `constr` as mean and standard deviations in the matrix `constr_sd` of same size as `constr`.

```
R> constr_sd <- matrix(NA,nrow=fcast$fhorz,ncol=ncol(model.1$xglobal))
R> colnames(constr_sd) <- colnames(model.1$xglobal)
R> constr_sd[1:5,"US.Dp"] <- 0.001
R> cond_fcast2 <- cond.predict(constr=constr, bgvar.obj=model.1, pred.obj=fcast,
+                             constr_sd=constr_sd)
```

After having computed the conditional forecasts we plot them.

## 5. Summary and discussion

This article introduces the R package **BGVAR**. It implements Bayesian global vector autoregressions with shrinkage priors and stochastic volatility. The package comes along with convenience functions such as, print and plot functions that allow to summarize the output, carry out residual and model diagnostics and analyze the in-sample fit of the model. In addition, the library offers a fully fledged toolkit to compute impulse response functions, forecast error variance and historical error variance decompositions. Structural shocks can



(a) Without uncertainty bounds over the conditioning path.

(b) With uncertainty over the conditioning path.

Figure 6: Conditional forecast of US inflation, holding inflation constant for 5 periods.

be identified either using a combination of zero and sign restrictions or by applying a simple Cholesky decomposition. Sign restrictions and shocks can be applied on single units or on the cross-section. Generalized impulse response functions are implemented as well. The package also provides functions to conduct and evaluate out-of-sample forecasts as well as conditional forecasts that allow to set a future path for a particular variable of interest.

Further applications of the **BGVAR** package are contained in the vignette that accompanies the package. These feature estimation of the models with different weights, including additional units to model e.g., the joint monetary policy in the euro area (Feldkircher, Gruber, and Huber 2020), or the oil price (Cashin, Mohaddes, Raissi, and Raissi 2014), applications of zero- and sign restrictions coupled with rationality conditions of D’Amico and King (2015), and conditional and unconditional forecasts. Naturally, all the necessary commands are explained in the package documentation that also include detailed examples.

The Bayesian treatment of the unit-specific models is a unique feature of **BGVAR**. This allows to include large information sets and in turn, typically results in considerable gains in forecast performance over standard GVAR models. Another stand-alone feature of **BGVAR** is its efficient implementation: large parts of the estimation function are written in C++ utilizing the **Rcpp** and **RcppArmadillo** packages. In addition, both estimation and structural analysis can be carried out using parallel computing drawing on the **doParallel** and **foreach** packages. This implies that most applications can be carried out on standard desktop computer and in a reasonable amount of time, which greatly enhances the usability of the **BGVAR** package.

## Computational details

The results in this paper were obtained using R 3.6.3 with the **BGVAR** 2.0.1 package. R itself

and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

## Acknowledgements

The authors would like to thank Gregor Zens, Lukas Vashold and Nikolaus Kuschnig and Thomas O. Zörner for helpful comments and suggestions.

## References

- Ankargren S, Yang Y (2020). *mfavar: Mixed-Frequency Bayesian VAR Models*. R package version 0.5.3, URL <https://CRAN.R-project.org/package=mfavar>.
- 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.
- Bates D, Maechler M (2019). *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-18, URL <https://CRAN.R-project.org/package=Matrix>.
- 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.
- Carriero A, Clark TE, Marcellino M (2019). “Large Bayesian vector autoregressions with stochastic volatility and non-conjugate priors.” *Journal of Econometrics*, **212**(1), 137–154.
- Cashin P, Mohaddes K, Raissi M, Raissi M (2014). “The differential effects of oil demand and supply shocks on the global economy.” *Energy Economics*, **44**(C), 113–134. URL <https://ideas.repec.org/a/eee/eneeco/v44y2014icp113-134.html>.
- Chudik A, Pesaran MH (2016). “Theory and practice of GVAR modelling.” *Journal of Economic Surveys*, **30**(1), 165–197.
- Clark TE (2011). “Real-time density forecasts from Bayesian vector autoregressions with stochastic volatility.” *Journal of Business & Economic Statistics*, **29**(3), 327–341.
- Crespo Cuaresma J, Feldkircher M, Huber F (2016). “Forecasting with global vector autoregressive models: A Bayesian approach.” *Journal of Applied Econometrics*, **31**(7), 1371–1391.
- D’Amico S, King TB (2015). “What Does Anticipated Monetary Policy Do?” *Working Paper Series WP-2015-10*, Federal Reserve Bank of Chicago. URL <https://ideas.repec.org/p/fip/fedhwp/wp-2015-10.html>.
- Dees S, di Mauro F, Pesaran HM, Smith LV (2007). “Exploring the international linkages of the euro area: a global VAR analysis.” *Journal of Applied Econometrics*, **22**(1).
- Dovern J, Feldkircher M, Huber F (2016). “Does joint modelling of the world economy pay off? Evaluating global forecasts from a Bayesian GVAR.” *Journal of Economic Dynamics and Control*, **70**, 86 – 100. ISSN 0165-1889. doi:<http://dx.doi.org/10.1016/j.jedc.2016.05.001>.

- 1016/j.jedc.2016.06.006. URL <http://www.sciencedirect.com/science/article/pii/S0165188916301051>.
- Eddelbuettel D, François R (2011). “Rcpp: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08. URL <http://www.jstatsoft.org/v40/i08/>.
- Eddelbuettel D, Sanderson C (2014). “RcppArmadillo: Accelerating R with high-performance C++ linear algebra.” *Computational Statistics and Data Analysis*, **71**, 1054–1063. URL <http://dx.doi.org/10.1016/j.csda.2013.02.005>.
- Eickmeier S, Ng T (2015). “How do US credit supply shocks propagate internationally? A GVAR approach.” *European Economic Review*, **74**, 128–145.
- Feldkircher M, Gruber T, Huber F (2020). “International effects of a compression of euro area yield curves.” *Journal of Banking & Finance*, **113**, 11–14.
- Feldkircher M, Huber F (2016). “The international transmission of US shocks—Evidence from Bayesian global vector autoregressions.” *European Economic Review*, **81**, 167–188.
- Feldkircher M, Huber F, Moder I (2015). “Towards a New Normal: How Different Paths of US Monetary Policy Affect the World Economy.” *Economic Notes*, **44**(3), 409–418. ISSN 1468-0300. doi:10.1111/ecno.12041. URL <http://dx.doi.org/10.1111/ecno.12041>.
- Feldkircher M, Huber F, Pfarrhofer M (2019). “Factor Augmented Vector Autoregressions, Panel VARs, and Global VARs.” In P Fuleky (ed.), *Macroeconomic Forecasting in the Era of Big Data*, volume 52. Springer.
- Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2020). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-12, URL <https://CRAN.R-project.org/package=mvtnorm>.
- George EI, McCulloch RE (1993). “Variable selection via Gibbs sampling.” *Journal of the American Statistical Association*, **88**(423), 881–889.
- George EI, Sun D, Ni S (2008). “Bayesian stochastic search for VAR model restrictions.” *Journal of Econometrics*, **142**(1), 553–580.
- Geweke J (1992). “Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments.” In J Bernardo, J Berger, A Dawid, A Smith (eds.), *Bayesian statistics*, chapter 4, pp. 641–649. Clarendon Press.
- Giannone D, Lenza M, Primiceri GE (2015). “Prior selection for vector autoregressions.” *Review of Economics and Statistics*, **97**(2), 436–451.
- Griffin JE, Brown PJ, et al. (2010). “Inference with normal-gamma prior distributions in regression problems.” *Bayesian Analysis*, **5**(1), 171–188.
- Huber F (2016). “Density forecasting using Bayesian global vector autoregressions with stochastic volatility.” *International Journal of Forecasting*, **32**(3), 818–837.
- Huber F, Feldkircher M (2019). “Adaptive shrinkage in Bayesian vector autoregressive models.” *Journal of Business & Economic Statistics*, **37**(1), 27–39.

- Kastner G (2016). “Dealing with stochastic volatility in time series using the R package `stochvol`.” *Journal of Statistical Software*, **69**.
- 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.
- Kilian L, Lütkepohl H (2017). *Structural vector autoregressive analysis*. Cambridge University Press.
- Koop G, Korobilis D (2010). *Bayesian multivariate time series methods for empirical macroeconomics*. Now Publishers Inc.
- Koop GM (2013). “Forecasting with medium and large Bayesian VARs.” *Journal of Applied Econometrics*, **28**(2), 177–203.
- 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 (2019). *BVAR: Hierarchical Bayesian Vector Autoregression*. R package version 0.2.0, URL <https://CRAN.R-project.org/package=BVAR>.
- Lanne M, Nyberg H (2016). “Generalized forecast error variance decomposition for linear and nonlinear multivariate models.” *Oxford Bulletin of Economics and Statistics*, **78**(4), 595–603.
- Leydold J, Hormann W (2017). *GIGrvg: Random Variate Generator for the GIG Distribution*. R package version 0.5, URL <https://CRAN.R-project.org/package=GIGrvg>.
- Microsoft Corporation, Weston S (2019a). *doParallel: Foreach Parallel Adaptor for the ‘parallel’ Package*. R package version 1.0.15, URL <https://CRAN.R-project.org/package=doParallel>.
- Microsoft Corporation, Weston S (2019b). *foreach: Provides Foreach Looping Construct*. R package version 1.4.7, URL <https://CRAN.R-project.org/package=foreach>.
- Mohr FX (2019). *bvartools: Functions for Bayesian Inference of Vector Autoregressive Models*. R package version 0.0.2, URL <https://CRAN.R-project.org/package=bvartools>.
- Pesaran HH, Shin Y (1998). “Generalized impulse response analysis in linear multivariate models.” *Economics letters*, **58**(1), 17–29.
- Pesaran MH, Schuermann T, Smith LV (2009). “Forecasting economic and financial variables with global VARs.” *International Journal of Forecasting*, **25**(4), 642–675.
- Pesaran MH, Schuermann T, Weiner SM (2004). “Modeling regional interdependencies using a global error-correcting macroeconomic model.” *Journal of Business & Economic Statistics*, **22**(2), 129–162.
- Pfaff B (2008). “VAR, SVAR and SVEC Models: Implementation Within R Package `vars`.” *Journal of Statistical Software*, **27**(4), 1–32. URL <http://www.jstatsoft.org/v27/i04/>.

- Plate T, Heiberger R (2016). *abind: Combine Multidimensional Arrays*. R package version 1.4-5, URL <https://CRAN.R-project.org/package=abind>.
- Primiceri GE (2005). “Time varying structural vector autoregressions and monetary policy.” *The Review of Economic Studies*, **72**(3), 821–852.
- R Core Team (2019a). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- R Core Team (2019b). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Richardson P (2020). *BHSBVAR: Structural Bayesian Vector Autoregression Models*. R package version 2.0.2, URL <https://CRAN.R-project.org/package=BHSBVAR>.
- Rossi P (2019). *bayesm: Bayesian Inference for Marketing/Micro-Econometrics*. R package version 3.1-4, URL <https://CRAN.R-project.org/package=bayesm>.
- Rubio-Ramirez JF, Waggoner DF, Zha T (2010). “Structural vector autoregressions: Theory of identification and algorithms for inference.” *The Review of Economic Studies*, **77**(2), 665–696.
- Sims CA, Zha T (2006). “Were there regime switches in US monetary policy?” *The American Economic Review*, **96**(1), 54–81.
- Smith, LV and Galesi, A (2014). “GVAR Toolbox.” URL <https://sites.google.com/site/gvarmodelling/gvar-toolbox>.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL <http://www.stats.ox.ac.uk/pub/MASS4>.
- Waggoner DF, Zha T (1999). “Conditional forecasts in dynamic multivariate models.” *Review of Economics and Statistics*, **81**(4), 639–651.