R PACKAGE BHSBVAR

Paul Richardson
p.richardson.54391@gmail.com
November, 2018

ACKNOWLEDGEMENTS

The BHSBVAR package is based on the Matlab programs created by Baumeister and Hamilton (2015, 2017, 2018). I thank them for sharing their Matlab programs online.

INTRODUCTION

Identifying structural innovations from Vector Autoregression (VAR) models requires the researcher to make assumptions about the structural parameters in the model. Recursively identifying structural innovations with the Cholesky decomposition of the residual covariance matrix requires the researcher to assume exclusion or information lag restrictions for structural parameters. Identifying structural innovations with sign restrictions prevents the researcher from having to make zero restrictions but this method implicitly requires the researcher to assume a particular prior distribution they may not agree with. The method developed by Baumeister and Hamilton (2015, 2017, 2018) for estimating the parameters of a Structural Bayesian Vector Autoregression (SBVAR) model allows the researcher to explicitly include prior information about those parameters. Their method does not require the researcher to make extreme zero restrictions or force the researcher to assume a prior distribution about structural parameters the researcher does not agree with. For detailed information about this method see Baumeister and Hamilton (2015, 2017, 2018).

MODEL

Let Y be an $(n \times T)$ matrix of endogenous variables. X is an $(k \times T)$ matrix containing L lags of Y and a constant. A is an $(n \times n)$ matrix containing the short-run elasticities or the structural relationships between the endogenous variables in Y from a Structural Vector Autoregression (SVAR) model. B is an $(n \times k)$ matrix containing lagged structural coefficients. U is an $(n \times T)$ vector of structural innovations. D is an $(n \times n)$ diagonal covariance matrix of the innovations from the structural model. n is the number of endogenous variables or equations. T is the number of observations and $k = nL + 1$.

Structural Vector Autoregression Model:

$$AY = BX + U \qquad U \sim N(0, D) \tag{1}$$

$$B = A\Phi \tag{2}$$

$$U = A\epsilon \tag{3}$$

$$D = \frac{UU^\top}{T} = A\Omega A^\top \tag{4}$$

Φ is an $(n \times k)$ matrix containing the lagged coefficients from the reduced form VAR model. ϵ is an $(n \times T)$ matrix of the VAR model residuals. Ω is an $(n \times n)$ symmetric covariance matrix of the residuals from the VAR model.

Reduced Form Vector Autoregression Model:

$$Y = \Phi X + \epsilon \qquad \epsilon \sim N(0, \Omega) \tag{5}$$

$$\Phi = (YX^\top)(XX^\top)^{-1} \tag{6}$$

$$\epsilon = Y - \Phi X \tag{7}$$

$$\Omega = \frac{\epsilon\epsilon^\top}{T} \tag{8}$$

Let y_i be an $(1 \times T)$ matrix containing a single endogenous variable (i) from Y. Let x_i be an $((L + 1) \times T)$ matrix of L lags of y_i and a constant. ϕ_i is an $(1 \times (L + 1))$ matrix containing the lagged coefficients from the reduced form univariate Autogregression (AR) model i. e_i is an $(1 \times T)$ matrix of the residuals from the univariate AR model i. e is an $(n \times T)$ matrix of residuals from the univariate AR models. Σ is an $(n \times n)$ symmetric covariance matrix of the residuals from the univariate AR models. Σ_i is the (i,i) element of Σ.

Reduced Form Univariate Autoregression Model:

$$y_i = \phi_i x_i + e_i \qquad e_i \sim N(0, \Sigma_i) \tag{9}$$

$$\phi_i = (y_i x_i^\top)(x_i x_i^\top)^{-1} \tag{10}$$

$$e_i = y_i - \phi_i x_i \tag{11}$$

$$\Sigma = \frac{ee^\top}{T} \tag{12}$$

Let P be an $(n \times k)$ matrix containing the prior position values for the reduced form lagged coefficient matrix, Φ. M^{-1} is an $(k \times k)$ symmetric matrix indicating confidence in P. R is an $(n \times k)$ matrix containing the prior position values for the reduced form lagged coefficient matrix (Φ) representing a belief that specific shocks have no long-run affect. V_i^{-1} is an $(k \times k)$ symmetric matrix indicating

confidence in R for equation i. β_i is an $(n \times k)$ lagged structural coefficient matrix. $\beta_i(i,)$ is an $(1 \times k)$ matrix of lagged structural coefficients for equation i from β_i. $B(i,)$ is an $(1 \times k)$ matrix of the lagged structural coefficients for equation i from B. ζ_i is an $(n \times n)$ matrix for equation i. Z is an $(n \times n)$ diagonal matrix. $\zeta_i(i,i)$ is the (i,i) element of the symmetric matrix ζ_i. $Z(i,i)$ is the (i,i) element of the the diagonal matrix Z. κ is an $(n \times n)$ diagonal matrix whose elements along the main diagonal represent confidence in the priors for the structural variances. κ_i refers to element (i,i) of κ. $diag(A\Sigma A^\top)$ is an $(n \times n)$ diagonal matrix whose main diagonal elements are the main diagonal elements from the matrix $A\Sigma A^\top$. τ_i refers to element (i,i) of τ. $D(i,i)$ refers to element (i,i) of D,

Structural Bayesian Vector Autoregression Model:

$$AY = BX + U \qquad U \sim N(0, D) \tag{13}$$

$$\beta_i = A(YX^\top + PM^{-1} + RV_i^{-1})(XX^\top + M^{-1} + V_i^{-1})^{-1} \tag{14}$$

$$B(i,) = \beta_i(i,) \tag{15}$$

$$\zeta_i = A[(YY^\top + PM^{-1}P^\top + RV_i^{-1}R^\top) - (YX^\top + PM^{-1} + RV_i^{-1})(XX^\top + M^{-1} + V_i^{-1})^{-1}(YX^\top + PM^{-1} + RV_i^{-1})^\top]A^\top \tag{16}$$

$$Z(i,i) = \zeta_i(i,i) \tag{17}$$

$$\tau = \kappa diag(A\Sigma A^\top) \tag{18}$$

$$\tau^* = \tau + \tfrac{1}{2}Z \tag{19}$$

$$D(i,i) = (\kappa_i + \tfrac{T}{2})^{-1}\tau_i^* \tag{20}$$

Baumeister and Hamilton (2015, 2017, 2018) developed an algorithm that estimates the parameters of an SVAR model using Bayesian methods (SBVAR). Their algorithm applies a random-walk Metropolis-Hastings algorithm to seek elasticity values for A, considering prior information, that diagonalizes the covariance matrix of the reduced form errors. Their algorithm provides more flexibility to the researcher in restricting the model parameters. The researcher is not required to assume information time lags as in recursively identified VAR models nor a prior distribution they have no control over as in sign restricted VAR models. For information about their algorithm see Baumeister and Hamilton (2015, 2017, 2018).

Posterior Distribution:

$$p(A|Y) \propto p(A)[det(A\Omega A^\top)]^{\frac{T}{2}} \prod_{i=1}^{n} \tau_i^{\kappa_i} (\tfrac{2}{T}\tau_i^*)^{-(\kappa_i + \frac{T}{2})} \tag{21}$$

Where $p(A)$ is the product of prior densities. $det(A\Omega A^\top)$ is the determinant of the matrix $A\Omega A^\top$.

Algorithm:

Step 1)
Generate proposals for the elements in A, $\tilde{A}^{(c+1)}$. $\tilde{A}^{(c)}$ are the starting values for A when $c = 1$. Compute $p(\tilde{A}^{(c)}|Y)$ and $p(\tilde{A}^{(c+1)}|Y)$. If $p(\tilde{A}^{(c+1)}|Y) < p(\tilde{A}^{(c)}|Y)$ set $p(\tilde{A}^{(c+1)}|Y) = p(\tilde{A}^{(c)}|Y)$ with probability $1 - \frac{p(\tilde{A}^{(c+1)}|Y)}{p(\tilde{A}^{(c)}|Y)}$.

Step 2)
Generate draws for D, $\tilde{D}^{(c+1)}$. $\tilde{D}_i{}^{(c+1)} \sim \Gamma(\kappa_i + \frac{T}{2}, \tilde{\tau}_i{}^{*(c+1)})$. $\tilde{D}_i{}^{(c+1)}$, κ_i, and $\tilde{\tau}_i{}^{*(c+1)}$ refers to element (i,i) of $\tilde{D}^{(c+1)}$, κ, and $\tilde{\tau}^{*(c+1)}$, respectively. $\tilde{\tau}^{*(c+1)}$ are estimates of τ, replacing A with $\tilde{A}^{(c+1)}$.

Step 3)
Generate draws for B, $\tilde{B}^{(c+1)}$. $\tilde{B}^{(c+1)}(i,) = \hat{\tilde{\beta}}_i^{(c+1)}(i,)$. $\hat{\tilde{\beta}}_i^{(c+1)} \sim N(\tilde{\beta}_i^{(c+1)}, \tilde{\Psi}_i^{(c+1)})$. $\tilde{\beta}_i^{(c+1)}$ are estimates of β_i, replacing A with $\tilde{A}^{(c+1)}$. $\tilde{\Psi}_i^{(c+1)} = \tilde{D}^{(c+1)^{-1}}(XX^\top + M^{-1} + V_i^{-1})^{-1}$.

Step 4)
Increase c by 1 and repeat Steps 1-4 for $c = 2, 3, ..., C$.

EXAMPLE

The `BHSBVAR` package provides a function for running Structural Bayesian Vector Autoregression (SBVAR) models and several functions for plotting results. The `BH_SBVAR()` function runs a SBVAR model with the methods developed by Baumeister and Hamilton (2015, 2017, 2018). The `IRF_Plots()` function creates plots of impulse responses. The `HD_Plots()` function creates plots of historical decompositions. The `Dist_Plots()` function creates posterior density plots of the model parameters in A, H, and $det(A)$ overlaid with prior densities to illustrate the difference between posterior and prior distributions. The following example illustrates how these functions can be applied to reproduce the results from Baumeister and Hamilton (2015)

```
> rm(list = ls())
> library(BHSBVAR)
> set.seed(123)
> data(USLMData)
> y <- matrix(data = c(USLMData$Wage, USLMData$Employment), ncol = 2)
> colnames(y) <- c("Wage", "Employment")
```

The first line clears memory. The second line loads the `BHSBVAR` package library. The third line sets the seed for random number generation. The fourth line imports the data used in this example.

The fifth line creates a matrix (y) containing quarter over quarter percent change of U.S. real wage and employment data use by Baumeister and Hamilton (2015).

```
> nlags <- 8
> itr <- 200000
> burn <- 0
> thin <- 20
> acc_irf <- TRUE
> h1_irf <- 20
> ci <- 0.975
```

nlags sets the lag length used in the SBVAR model. itr sets the number of iterations for the algorithm. burn is the number of draws to throw out at the beginning of the algorithm. thin sets the thinning parameter which will thin the Markov-Chains. acc_irf indicates whether accumulated impulse responses are to be computed and returned. h1_irf indicates the time horizon for computing impulse responses. ci indicates the credibility intervals to be returned.

```
> pA <- array(data = NA, dim = c(2, 2, 8))
> pA[, , 1] <- c(0, NA, 0, NA)
> pA[, , 2] <- c(1, NA, -1, NA)
> pA[, , 3] <- c(0.6, 1, -0.6, 1)
> pA[, , 4] <- c(0.6, NA, 0.6, NA)
> pA[, , 5] <- c(3, NA, 3, NA)
> pA[, , 6] <- c(NA, NA, NA, NA)
> pA[, , 7] <- c(NA, NA, 1, NA)
> pA[, , 8] <- c(2.4, NA, 2.4, NA)
```

These lines create an array containing all the information needed to set priors for each element in A. Each column contains the prior information for the parameters in each equation. The third dimension of pA should always have a length of 8. The first slice of the third dimension of pA indicates the prior distribution (NA - no prior, 0 - symmetric t-distribution, 1 - non-central t-distribution). The second slice indicates sign restrictions for symmetric t-distributions (NA - no restriction, 1 - positive restriction, -1 - negative restriction). The third slice indicates the position of the prior. The fourth slice indicates the scale or confidence in the prior. The fifth slice indicates the degrees of freedom for the t-distribution prior. The sixth slice indicates skew for non-central t-distribution prior. The seventh slice indicates long-run restriction scale or confidence in the long-run restriction. The eighth slice indicates the random-walk proposal scale parameters which adjust the algorithm's acceptance rate and the ability of the algorithm to adequately cover the model's parameter space. For information about priors for A see Baumeister and Hamilton (2015, 2017, 2018). The formulas used to compute the density of the prior distributions are listed in the Appendix.

```
> pP <- matrix(data = 0, nrow = ((nlags * ncol(pA)) + 1), ncol = ncol(pA))
> pP[1:nrow(pA), 1:ncol(pA)] <-
+   diag(x = 1, nrow = nrow(pA), ncol = ncol(pA))
> x1 <-
```

```
+   matrix(data = NA, nrow = (nrow(y) - nlags),
+          ncol = (ncol(y) * nlags))
> for (k in 1:nlags) {
+   x1[, (ncol(y) * (k - 1) + 1):(ncol(y) * k)] <-
+     y[(nlags - k + 1):(nrow(y) - k),]
+ }
> x1 <- cbind(x1, 1)
> colnames(x1) <-
+   c(
+     paste(
+       rep(colnames(y), nlags), ".L",
+       sort(rep(seq(from = 1, to = nlags, by = 1), times = ncol(y)),
+            decreasing = FALSE),
+       sep = ""
+       ),
+     "cons"
+     )
> y1 <- y[(nlags + 1):nrow(y),]
> ee <- matrix(data = NA, nrow = nrow(y1), ncol = ncol(y1))
> for (i in 1:ncol(y1)) {
+   xx <- cbind(x1[, seq(from = i, to = (ncol(x1) - 1), by = ncol(y1))], 1)
+   yy <- matrix(data = y1[, i], ncol = 1)
+   phi <- solve(t(xx) %*% xx, t(xx) %*% yy)
+   ee[, i] <- yy - (xx %*% phi)
+ }
> somega <- (t(ee) %*% ee) / nrow(ee)
> lambda0 <- 0.2
> lambda1 <- 1
> lambda3 <- 100
> v1 <- matrix(data = (1:nlags), nrow = nlags, ncol = 1)
> v1 <- v1^((-2) * lambda1)
> v2 <- matrix(data = diag(solve(diag(diag(somega)))), ncol = 1)
> v3 <- kronecker(v1, v2)
> v3 <- (lambda0^2) * rbind(v3, (lambda3^2))
> v3 <- 1 / v3
> pP_sig <- diag(x = 1, nrow = nrow(v3), ncol = nrow(v3))
> diag(pP_sig) <- v3
```

These lines create matrices containing prior position (pP) and scale or confidence (pP_sig) information for the reduced form lagged coefficient matrix Φ. pP and pP_sig correspond to the P and M^{-1} matrices from Equation 14, respectively. Variance estimates from univariate Autoregression models, lambda0, lambda1, and lambda3 are used to construct pP_sig. lambda0 controls the overall confidence in the priors, lambda1 controls the confidence in higher order lags, and lambda3 controls the confidence in the constant term. For information about priors for Φ and B see Baumeister and Hamilton (2015, 2017, 2018), Doan, Sims, and Zha (1984), and Sims and Zha

(1984).

```
> pR_sig <-
+   array(data = 0,
+         dim = c(((nlags * ncol(y)) + 1),
+                 ((nlags * ncol(y)) + 1),
+                 ncol(y)))
> Ri <-
+   cbind(
+     kronecker(matrix(data = 1, nrow = 1, ncol = nlags),
+               matrix(data = c(1, 0), nrow = 1)),
+     0)
> pR_sig[,,2] <- (t(Ri) %*% Ri) / 0.1
> kappa1 <- matrix(data = 2, nrow = 1, ncol = ncol(y))
```

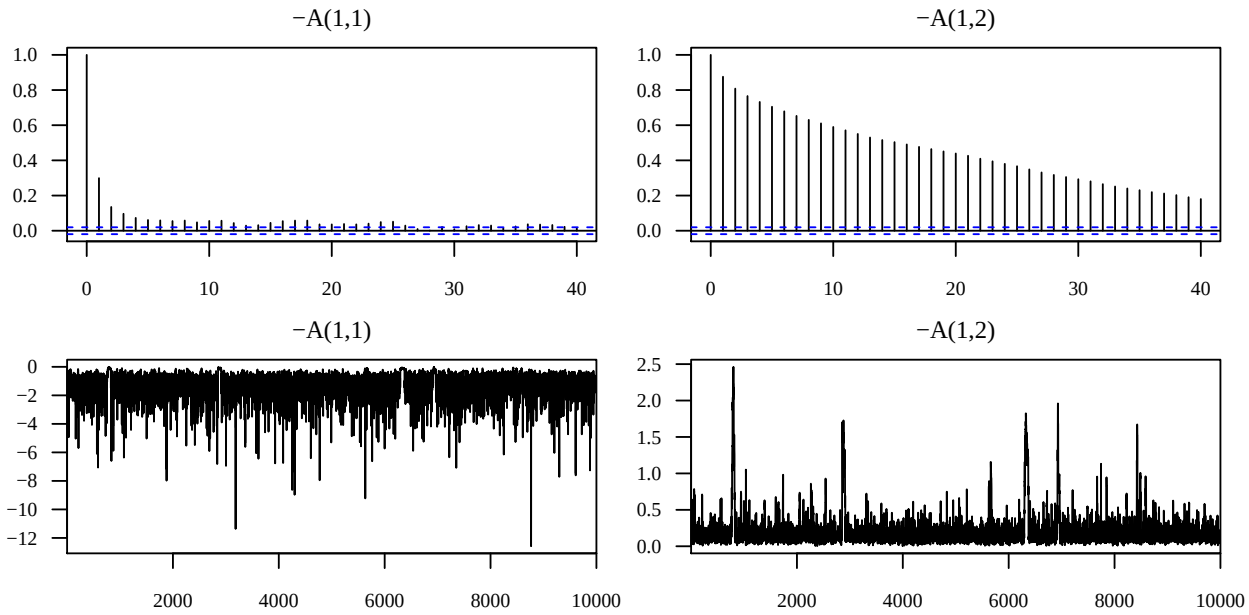
These lines create an array (`pR_sig`) containing values indicating confidence in long-run restrictions. `pR_sig` corresponds to the V_i^{-1} matrix from Equation 14. The matrix R from Equation 14 will be created automatically by the `BH_SBVAR()` function. The length of the third dimension of `pR_sig` is equal to the number of endogenous variables or the number of equations in the model. The first slice of the third dimension contains all zeros since there are no long-run restrictions for the first equation in the model. The second slice contains values indicating the confidence in the long-run restriction assigned to the lagged parameters of the second equation. For information about long-run restrictions see Baumeister and Hamilton (2015, 2018) and Blanchard and Quah (1989). `kappa1` is an $(1 \times n)$ matrix whose values correspond to the elements of the main diagonal of κ from Equation 18 and indicates the confidence in prior information about the structural variances.

```
> par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
+     mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)
> results1 <-
+   BH_SBVAR(y = y, nlags = nlags, pA = pA, pP = pP, pP_sig = pP_sig,
+            pR_sig = pR_sig, kappa1 = kappa1, itr = itr, burn = burn,
+            thin = thin, acc_irf = acc_irf,
+            h1_irf = h1_irf, ci = ci)
```

The `BH_SBVAR()` function allows the user to include prior information for A, $det(A)$, H, P, R, and κ directly when running a SBVAR model. The `pdetA` and `pH` arguments are arrays containing prior information for $det(A)$ and the elements of H and are not included in this example. The `BH_SBVAR()` function returns a list that includes the acceptance rate (`accept_rate`) of the algorithm, a matrix containing the endogenous variables (`y`), a matrix containing the lags of the endogenous variables (`x`), and the prior information provided to the function (`pA`, `pdetA`, `pH`, `pP`, `pP_sig`, `pR`, `pR_sig`). Arrays containing estimates of the model parameters are returned (`A`, `detA`, `H`, `B`, `Phi`, `HD`, `IRF`). The first, second, and third slices of the third dimension of these arrays are lower, median, and upper bounds of the estimates, respectively. Lists containing the horizontal and vertical axis coordinates of posterior densities for the estimates of A, $det(A)$, and H are returned (`A_den`, `detA_den`, and `H_den`). In addition, line and autocorrelation plots of the Markov-Chains of A, $det(A)$, and H

are returned for diagnostic purposes. The line and autocorrelation plots provide an indication of how well the algorithm covers the model's parameter space. The line plots in Figure 1 display the Markov-chains of the estimates from the algorithm with the estimate values shown on the vertical axis and the iteration number shown on the horizontal axis. The ACF plots in Figure 1 displays the autocorrelation of the Markov-chains of the estimates from the algorithm with the correlation estimates on the vertical axis and the lag length shown on the horizontal axis.

FIGURE 1
Posterior Line and ACF Diagnostic Plots



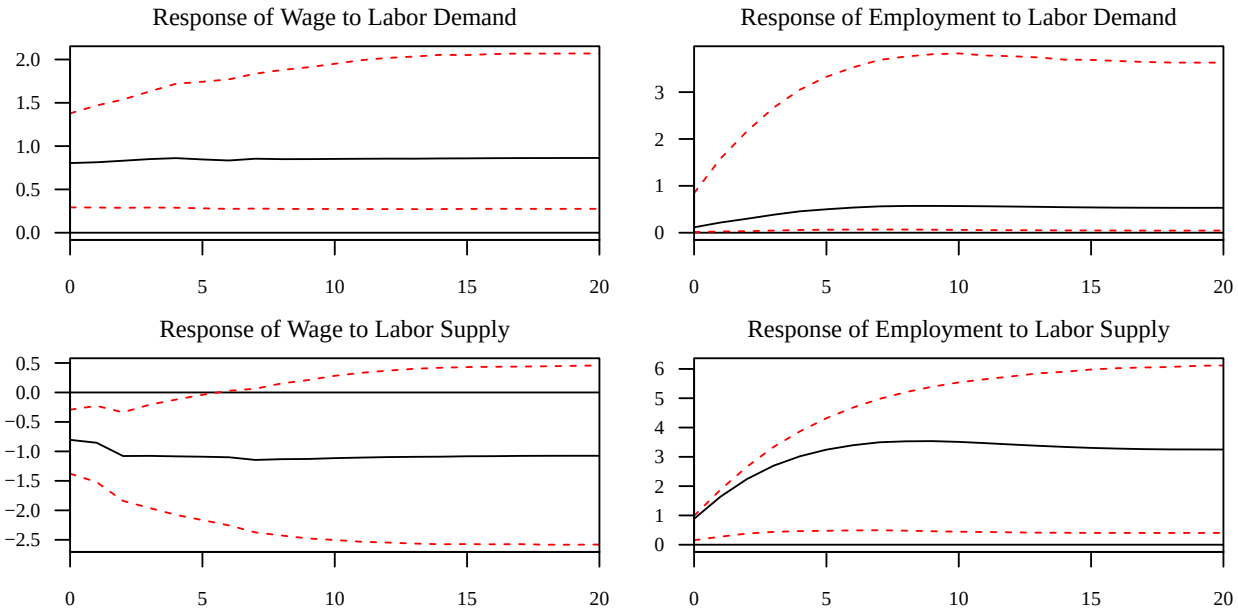
The titles of the plots in Figure 1 indicate the element of the coefficient matrix that is plotted. The plots for the elements of *A* are multiplied by -1 to illustrate elasticity values and/or isolate the dependent variable for each equation. These elements correspond to those found in the results from running the `BH_SBVAR()` function and correspond to the transpose of those from the mathematical representation from Equation 13. In other words, each column of the coefficient matrix arrays in the resulting list object from running the `BH_SBVAR()` function are coefficient estimates for each equation. However, each row of the coefficient matrices from the mathematical representation described in Equation 13 represent the parameters of each equation.

```
> VarNames <- colnames(USLMData)[2:3]
> ShockNames <- c("Labor Demand","Labor Supply")
> par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
+     mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)
> irf_results <-
+   IRF_Plots(results = results1, varnames = VarNames,
+             shocknames = ShockNames)
```

The first two lines store the names of endogenous variables and structural shocks. The third line sets the parameters used to display the plots. The `IRF_Plots()` function creates plots of impulse

responses. This function can be used to display the response of the endogenous variables following a particular structural shock. The `results` argument is a list object containing the unaltered results from the `BH_SBVAR()` function. The `varnames` and `shocknames` argument are character vectors containing the variable names and shock names, respectively. The `xlab` and `ylab` arguments are not included in this example but they allow the user to include labels for the horizontal and vertical axes, respectively. The units along the horizontal axis in the plots from Figure 2 represent time periods following an initial shock. The units along the vertical axis in the plots from Figure 2 represent percent change following an initial shock since the endogenous variables included in the model are quarter over quarter percent change of U.S. real wage and employment. Figure 2 displays the cumulative response of U.S. real wage growth and employment growth to U.S. labor demand and supply shocks. In addition, this function returns a list containing the data used to produce the plots in Figure 2.

FIGURE 2
Impulse Responses

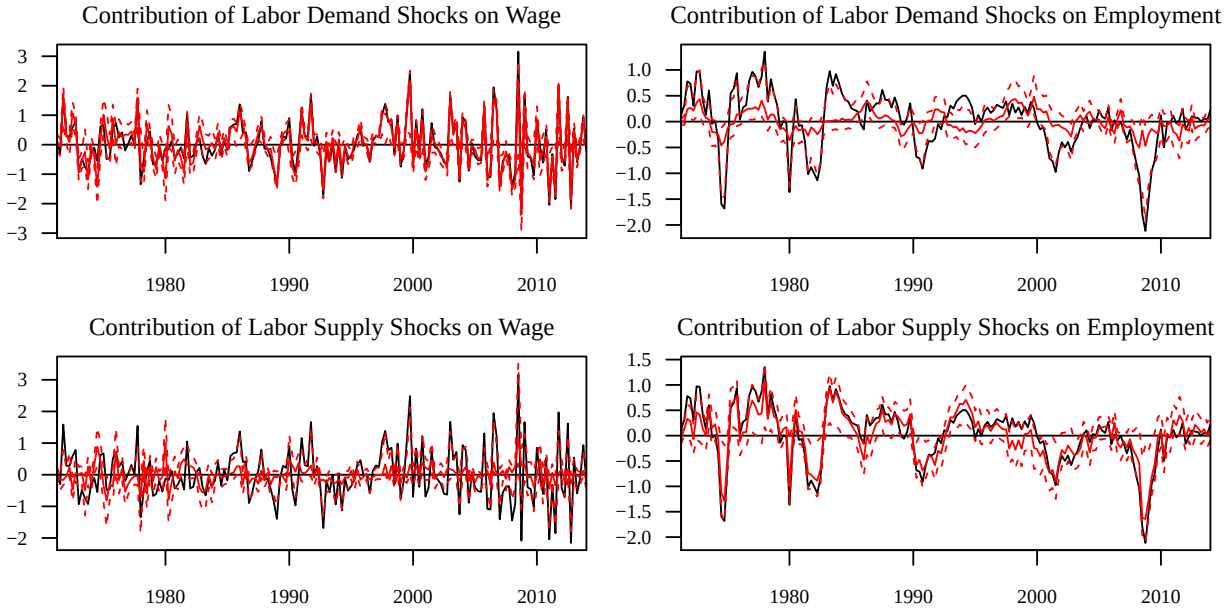



```
> par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
+     mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)
> hd_results <-
+   HD_Plots(results  = results1, varnames = VarNames,
+            shocknames = ShockNames,
+            freq = 4, start_date = c(1971, 2))
```

The first line sets the parameters used to display the plots. The `HD_Plots()` function creates plots of historical decompositions. This function can be used to display the cumulative effect of specific shocks on a particular endogenous variable at any given time period. The `results`, `varnames`, `shocknames`, `xlab`, `ylab` arguments for the `HD_Plots()` function are the same as those from the

IRF_Plots() function. However, the units along the horizontal axis in the plots from Figure 3 represent actual time periods. The units along the horizontal axis of each plot are created with the freq and start_date arguments. The freq argument is set to 4 since the endogenous variables are measured at a quarterly frequency in this example. The start_date argument represents the date of the first observation which is the second quarter of 1971 in this example. Figure 3 displays the historical decompositions. This function also returns a list of the data used to produce the plots in Figure 3.

FIGURE 3
Historical Decompositions

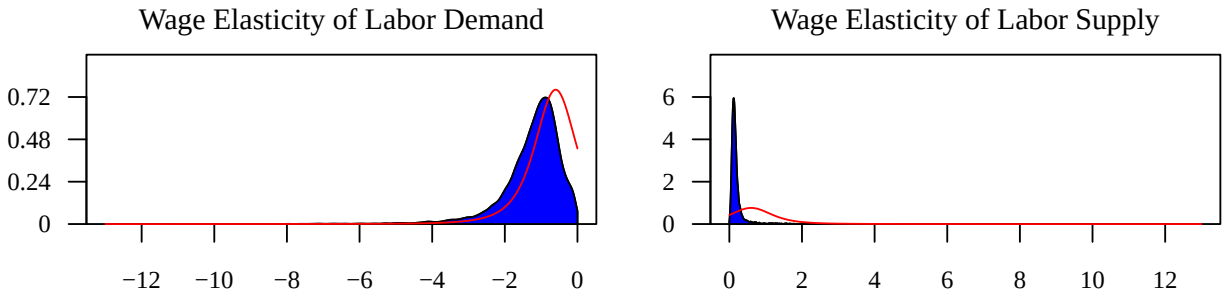



```
> A_titles <-
+   matrix(data = NA_character_, nrow = dim(pA)[1], ncol = dim(pA)[2])
> A_titles[1, 1] <- "Wage Elasticity of Labor Demand"
> A_titles[1, 2] <- "Wage Elasticity of Labor Supply"
> par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
+     mfcol = c(1, 2), mar = c(2, 2.2, 2, 1), las = 1)
> Dist_Plots(results = results1, A_titles = A_titles)
```

The first line sets the parameters used to display the plots. The Dist_Plots() function creates posterior density plots of the estimates for the elements in A, H, and $det(A)$. Prior densities are also plotted to illustrate the differences between posterior and prior distributions. The results, xlab, and ylab arguments for the Dist_Plots() function are the same as those from the IRF_Plots() and HD_Plots() functions. A_titles and H_titles arguments are matrices that contain the titles of the plots. The elements of the A_titles and H_titles matrices correspond to the elements of A and H arrays from the results of the BH_SBVAR() function. The posterior and prior density

plots for the elements of *A* are multiplied by -1 to illustrate elasticity values and/or the value of the coefficient if the dependent variable for each equation were isolated.

FIGURE 4
Posterior and Prior Distributions



APPENDIX

List of formulas used to compute the density of the prior distributions at some proposal value:
x1: is the proposal value.
c1: is the prior position parameter.
sigma: is the prior confidence in the position parameter, c1.
nu: is the degrees of freedom.
lam: is the non-centrality or skew parameter.
t-distribution:

```
> density <-
+   dt(x = ((x1 - c1) / sigma), df = nu, ncp = 0, log = FALSE) / sigma
```

Non-central t-distribution:

```
> density <-
+   dt(x = ((x1 - c1) / sigma), df = nu, ncp = lam, log = FALSE) / sigma
```

t-distribution truncated to be positive:

```
> density <-
+   dt(x = ((x1 - c1) / sigma), df = nu, ncp = 0, log = FALSE) /
+   (sigma *
+      (1 - pt(q = ((-c1) / sigma), df = nu, ncp = 0,
+              lower.tail = TRUE, log.p = FALSE)
+       )
+    )
```

t-distribution truncated to be negative:

```
> density <-
+   dt(x = ((x1 - c1) / sigma), df = nu, ncp = 0, log = FALSE) /
+   (sigma *
+      (pt(q = ((-c1) / sigma), df = nu, ncp = 0,
+          lower.tail = TRUE, log.p = FALSE)
+       )
+    )
```

REFERENCES

Baumeister, C., and Hamilton, J. D. (2015). Sign restrictions, structural vector autoregressions, and useful prior information. *Econometrica*, 83(5), 1963-1999.

Baumeister, C., and Hamilton, J. D. (2017). Structural interpretation of vector autoregressions with incomplete identification: Revisiting the role of oil supply and demand shocks (No. w24167). National Bureau of Economic Research.

Baumeister, C., and Hamilton, J. D. (2018). Inference in structural vector autoregressions when the identifying assumptions are not fully believed: Re-evaluating the role of monetary policy in economic fluctuations. *Journal of Monetary Economics*

Blanchard, O. J., and Quah, D. (1989). The Dynamic effects of aggregate demand and supply disturbances. *The American Economic Review*, 79(4), 655-673.

Sims, C. A., and Zha, T. (1998). Bayesian methods for dynamic multivariate models. *International Economic Review*, 39(4), 949-968.

Doan, T., Litterman, R. B., and Sims, C. (1984). Forecasting and conditional projection using realistic prior distributions. *Econometric Reviews*, 3(1), 1-100.